Data Appraisal Without Data Sharing

Mimee Xu
New York University

Laurens van der Maaten
Facebook AI Research

Awni Hannun
Zoom AI

Abstract

One of the most effective approaches to improving the performance of a machine learning model is to procure additional training data. A model owner seeking relevant training data from a data owner needs to appraise the data before acquiring it. However, without a formal agreement, the data owner does not want to share data. The resulting Catch-22 prevents efficient data markets from forming. This paper proposes adding a data appraisal stage that requires no data sharing between data owners and model owners. Specifically, we use multi-party computation to implement an appraisal function computed on private data. The appraised value serves as a guide to facilitate data selection and transaction. We propose an efficient data appraisal method based on forward influence functions that approximates data value through its first-order loss reduction on the current model. The method requires no additional hyper-parameters or re-training. We show that in private, forward influence functions provide an appealing trade-off between high quality appraisal and required computation, in spite of label noise, class imbalance, and missing data. Our work seeks to inspire an open market that incentivizes efficient, equitable exchange of domain-specific training data.

1 INTRODUCTION

In the real world, machine learning researchers often find their training data insufficient. Indeed, advances from cancer detection (Majkowska et al., 2020; Shen et al., 2019) to speech recognition (Amodei et al., 2016; Ardila et al., 2020) all rely heavily on the amount of quality data that is available to train the models. In the research process, it is often necessary to acquire more training data than initially anticipated, potentially from external data owners. Yet, domain-specific data is often valuable, thus kept private by default.

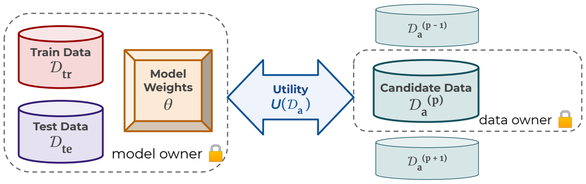

Figure 1: **Data Exchange.** Model owner and data owner guard their data from each other. Neither party can learn the true data utility, leading to a deadlock.

Consider a researcher using machine learning to detect phishing emails on her corporate network. As she seeks more data, privacy becomes a roadblock: not many peer companies will share their proprietary data, particularly without some form of compensation. At any rate, she must train her model on the data in order to evaluate its utility. Such a predicament compels her to either expend resources on data of unknown utility, or see her progress stall. The resulting Catch-22 prevents data sharing, as illustrated in Figure 1.

We thus seek an appraisal method to precede transactions. Such an appraisal is non-trivial because the value of data to a model owner depends on many factors, including the data the model owner already has, the complexity of their model, and the data distribution on which to perform predictions. Ideally, the model owner would: (1) re-train their model with and without the data to be appraised, and (2) measure the accuracy gain on the test set that results from including the additional training data.

The setup invokes secure multi-party computation, which is often used on machine learning tasks that have data privacy constraints, such as jointly training a model on disparate data (Mohassel and Zhang, 2017). However, private computation is unlike its plaintext counterparts: the training curves are not meant to be

Proceedings of the 25th International Conference on Artificial Intelligence and Statistics (AISTATS) 2022, Valencia, Spain. PMLR: Volume 151. Copyright 2022 by the author(s).

transparent to the model trainer, and hyper-parameter tuning, when done in private, is exceedingly costly.

Nevertheless, works in federated learning tackle profit sharing among data contributors with expansive private training, often examining all combinations of data selection (Song et al., 2019; Wang et al., 2019, 2020). However, the amount of computation and data required for these pioneering methods to be successful is far too great; after all, data and compute resources are oftentimes significant roadblocks for the typical researcher.

The most frugal researcher wishes to procure a dataset at a time, when the data is available and if the data is helpful. They won't risk including training data that is low-quality, and they don't want to splurge on private re-training just to find out. To appraise and select helpful data to train on, could the model owner avoid the cost of private training altogether?

To that end, we propose using private forward influence functions to perform appraisal before data transaction. A dataset's value is estimated with respect to a specific model, drawing on a first-order approximation to the test loss of the model updated with the new data. In secure computation, influence-based appraisal presents pronounced efficiency gain over fine-tuning, while evading private hyper-parameter tuning entirely.

We leverage our method in noisy, imbalanced, and incomplete data and show its efficiency and accuracy on logistic models, simulated with corruptions on MNIST, CIFAR-10's plane-to-car, and breast cancer classifciations. The results of our experiments show that computing influence functions via secure multi-party computation allows for high-quality data appraisal while requiring limited amounts of additional computation.

2 PROBLEM SETTING

We assume two parties in the transaction: a *model owner*, who is developing a machine-learning model with parameters θ, and a *data owner*, who possesses the dataset $\mathcal{D}_\mathrm{a}$ to be appraised. The model owner begins with training set $\mathcal{D}_\mathrm{tr}$ and test set $\mathcal{D}_\mathrm{te}$ to evaluate their model. To consider acquiring the data $\mathcal{D}_\mathrm{a}$, the model owner wishes to determine the utility gain from updating θ to fit $\mathcal{D}_\mathrm{tr} \cup \mathcal{D}_\mathrm{a}$. The model owner computes the model parameters $\hat{\theta}$ by minimizing the regularized empirical risk on the seed training dataset, $\mathcal{D}_\mathrm{tr}$:

$$\hat{\theta} = \arg\min_{\theta} \sum_{(\boldsymbol{x},y)\in\mathcal{D}_\mathrm{tr}} L(\boldsymbol{x}, y; \theta) + \lambda \|\theta\|_2^2. \quad (1)$$

After adding dataset $\mathcal{D}_\mathrm{a}$, they will compute the new optimal parameters, θ^*, by minimizing the regularized empirical risk on dataset $\mathcal{D}_\mathrm{tr} \cup \mathcal{D}_\mathrm{a}$ instead. We thus define dataset utility, $U(\mathcal{D}_\mathrm{a})$, as the difference between test losses on $\mathcal{D}_\mathrm{te}$:

$$U(\mathcal{D}_\mathrm{a}) = \frac{1}{|\mathcal{D}_\mathrm{te}|} \sum_{(\boldsymbol{x},y)\in\mathcal{D}_\mathrm{te}} L(\boldsymbol{x}, y; \hat{\theta}) - L(\boldsymbol{x}, y; \theta^*). \quad (2)$$

The challenge is to approximate this utility without requiring the model and data owners to share the model parameters, $\hat{\theta}$, or any of the datasets $\mathcal{D}_\mathrm{tr}$, $\mathcal{D}_\mathrm{te}$, and $\mathcal{D}_\mathrm{a}$. An appraisal function $f(\mathcal{D}_\mathrm{a})$ is designed as a proxy to $U(\mathcal{D}_\mathrm{a})$. To enable the use of influence functions for the proxy, we assume the loss $L(\cdot)$ is twice differentiable.

We further assumes $L(\theta)$ to be convex, excluding non-convex optimizations. We note that relaxing convexity would not change the application of our method, and would not affect the computational runtime of influence functions; the accuracy influence functions under non-convexity is an active area research (Basu et al., 2020a).

The proxy, $f(\cdot)$, only needs to recover the same relative utility over multiple datasets, $\{\mathcal{D}_\mathrm{a}^{(p)}\}$, as the ground truth, $U(\cdot)$. The user may calibrate $f(\cdot)$ to achieve a desired absolute utility depending on the use case. We thus assume the appraisal function to be scale-agnostic.

Threat Model. We assume a passively secure threat model. Both the model and data owners are *honest-but-curious.* The parties follow the MPC protocol but should not be able to learn anything from the data observed. We assume the appraisal of the dataset is revealed to both parties, and that the parties accept the associated information leakage. If such information leakage is unacceptable, the appraisal value can be kept secret while a single bit representing the acquisition decision can be revealed. A single bit result requires the model owner to pre-define a threshold value for $f(\mathcal{D}_\mathrm{a})$. Namely, to exclude negatively impacting datasets, a model owner may set the threshold to zero.

Metadata. The setup assumes both parties to have access to metadata about the dataset to be appraised, including the number of data samples, their dimensionality, and the number of classes. Relevant metadata may also include details on the data type, data encoding, label encoding, *etc.* We further assume that each $\mathcal{D}_\mathrm{a}$ to be appraised for a model has a fixed cardinality, which the model owner sets up prior to the appraisal.

3 DATA APPRAISAL WITHOUT DATA SHARING

To maintain the secrecy of the input data and model, appraisal function $f(\cdot)$ is evaluated using secure multi-party computation (MPC). However, MPC methods are compute-intensive, thus requiring careful crafting. Thus we closely examine the utility tradeoffs for forward

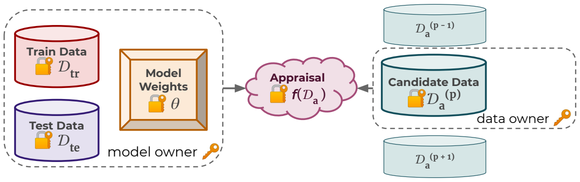
Train Data
$\mathcal{D}_{\text{tr}}$
Test Data
$\mathcal{D}_{\text{te}}$
Model Weights
θ
model owner
Appraisal
$f(\mathcal{D}_{\text{a}})$
$\mathcal{D}_{\text{a}}^{(\text{p}-1)}$
Candidate Data
$\mathcal{D}_{\text{a}}^{(\text{p})}$
data owner
$\mathcal{D}_{\text{a}}^{(\text{p}+1)}$

Figure 2: **Secure MPC.** Model owner and data owner encrypt their data. The appraisal is then performed privately, and its result is revealed to both parties.

influence functions against two efficient methods to appraise datasets: gradient norm and finetuning of the model using stochastic gradient descent (SGD).

3.1 Secure Multi-Party Computation (MPC)

Two-Party Private Appraisal. Secure MPC allows two or more parties to jointly evaluate a function on their combined data without revealing that data (which includes model parameters) or any intermediate values computed during the function evaluation (Evans et al., 2017). The appraisal function $f(\cdot)$ requires as input the data owner's data $\mathcal{D}_{\text{a}}$ and model owner's data $\mathcal{M} = \{\mathcal{D}_{\text{tr}}, \mathcal{D}_{\text{te}}, \hat{\theta}\}$. Let the E be the encryption function with decryption given by D. The private function $f_{\text{priv}}(\cdot)$ performs $f(\cdot)$ with MPC such that:

$$f(\mathcal{D}_{\text{a}}, \mathcal{M}) = D(f_{\text{priv}}(E(\mathcal{D}_{\text{a}}), E(\mathcal{M}))).$$

As Figure 2 shows, sensitive data does not leave any party's machine in the clear. As a result, the appraisal computation can be public and auditable, eliminating the need to trust secure hardware (Ohrimenko et al., 2016) or rely on an intermediate escrow service. Additionally, though every private appraisal is simply a two-party MPC between a model owner and a data owner, the appraisal methods may be generalized to including more data owners so that the shared compuatations need not be repeated. In following sections, we assume that each dataset $\mathcal{D}_{\text{a}} \in \{\mathcal{D}_{\text{a}}^{(p)}\}$ is benchmarked in a private two-party MPC against a fixed model $\mathcal{M}$. In notation, we abbreviate $f(\mathcal{D}_{\text{a}}, \mathcal{M})$ to $f(\mathcal{D}_{\text{a}})$.

Engineering Challenges. Despite MPC's suitability for private machine learning, performant MPC code requires specially-engineered software. Notably, floating point arithmetic, comparisons, and nonlinearities are approximated on a case-by-case basis to balance runtime, communication, memory, and numerical precision. Consequently, high-level frameworks greatly facilitate machine learning with secure MPC and other forms of secure and private function evaluation (SEAL, 2020; Ludwig et al., 2020; CrypTen, 2020). In particular, CrypTen (CrypTen, 2020) has a PyTorch-like interface for constructing machine-learning models including support for automatic differentiation. The MPC implementations of the appraisal methods described in Section 3 mirror their PyTorch equivalents. While the ground truth ranking comes from re-training in the clear, both finetuning and influence appraisals are studied using secure MPC implementations in CrypTen (Knott et al., 2020). We note that all CrypTen experiments in this work require no additional change except for a numerical precision setting of 24 bits.

Workflow Challenges. When data is private and never exchanged, MPC can be a challenging workflow for machine-learning model development. Without an appraisal and transaction phase, private training often presumes that the data is exclusively applied to a particular model and never revealed. Such a rigid MPC setup for model training is unappealing. In the clear, a researcher often owns both the data and the model, yet still requires external data. If that training data is never revealed, the researcher loses the ability to monitor key metrics, debug data, and potentially tune model architectures, which are typical to the workflow of model developers. Our work, in contrast, aids model owners with appraisal values computed in private, prior to the exchange of data. Thus the eventually transacted data is in the clear, maximizing flexibility.

3.2 Appraisal Methods and Their Private Implementations

Gradient Norm. While gradient information sits at the core of influence and finetuning, the norm of the gradient itself is a poor approximation for utility. To demonstrate, consider

$$f_{\text{gn}}(\mathcal{D}_{\text{a}}) = \left\| \sum_{(\boldsymbol{x},y)\in\mathcal{D}_{\text{a}}} \nabla_{\theta} L\left(\boldsymbol{x}, y; \hat{\theta}\right) \right\|_2, \tag{3}$$

which measures how surprising $\mathcal{D}_{\text{a}}$ is to a model trained on $\mathcal{D}_{\text{tr}}$. Indeed, the gradient norm can be large when the prior distribution of classes in $\mathcal{D}_{\text{a}}$ differs from that of $\mathcal{D}_{\text{tr}}$, as desired when $\mathcal{D}_{\text{tr}}$ is class-imbalanced. Yet, the gradient norm can also be large when $\mathcal{D}_{\text{a}}$ contains unfamiliar but useless or even harmful data. Under a simple formulation of label noise, f_{gn} inverts the desired ranking, as we will illustrate in Section 4.2. More information is needed to reveal relative utility.

Model Finetuning. To approximate data utility arbitrarily well, finetune a model on $\mathcal{D}_{\text{a}} \cup \mathcal{D}_{\text{tr}}$:

$$f_{\text{ft}}(\mathcal{D}_{\text{a}}) = \sum_{(\boldsymbol{x},y)\in\mathcal{D}_{\text{te}}} L(\boldsymbol{x}, y; \hat{\theta}) - L(\boldsymbol{x}, y; \hat{\theta}_{\text{ft}}), \tag{4}$$

where $\hat{\theta}_{\text{ft}}$ are the parameters after a fixed number of SGD updates on $\mathcal{D}_{\text{a}} \cup \mathcal{D}_{\text{tr}}$ seeded with $\hat{\theta}$. Despite its success in optimization in plain text, fine-tuning via SGD in private has novel challenges: it can be rather computationally intensive when implemented via MPC, because the number of sequential passes can be large. Moreover, since inspecting the training loss is not possible, successful SGD optimization in secure MPC requires careful pre-tuning of hyper-parameters.

Forward Influence Functions. The influence function $\mathcal{I}(\boldsymbol{x}, y)$ associates a training sample with the change in the model parameters under an infinitisemal up-weighting of that sample in the risk (Cook and Weisberg, 1982; Koh and Liang, 2017). We use influence functions to approximate the change on the resulting loss from including the dataset $\mathcal{D}_{\mathrm{a}}$. Denoting the empirical Hessian $\boldsymbol{H}_{\hat{\theta}} = \frac{1}{|\mathcal{D}_{\mathrm{tr}}|} \sum_{(\boldsymbol{x},y)\in\mathcal{D}_{\mathrm{tr}}} \nabla_\theta^2 L(\boldsymbol{x}, y, \hat{\theta})$, the forward influence of sample $(\boldsymbol{x}, y)$ is given by:

$$\mathcal{I}(\boldsymbol{x}, y) = -\boldsymbol{H}_{\hat{\theta}}^{-1} \nabla_\theta L(\boldsymbol{x}, y, \hat{\theta}). \tag{5}$$

This function is a first-order approximation of the change in $\hat{\theta}$ for each sample $(\boldsymbol{x}, y) \in \mathcal{D}_{\mathrm{a}}$. In turn, we can use $\Delta\theta \approx \mathcal{I}$ to assess the influence of $(\boldsymbol{x}, y)$ on the test loss of $(\boldsymbol{x}_{\mathrm{te}}, y_{\mathrm{te}})$ via the chain rule:

$$L(\boldsymbol{x}_{\mathrm{te}}, y_{\mathrm{te}}; \theta^*) - L(\boldsymbol{x}_{\mathrm{te}}, y_{\mathrm{te}}; \hat{\theta}) \approx \nabla_\theta L(\boldsymbol{x}_{\mathrm{te}}, y_{\mathrm{te}}; \hat{\theta})^\top \mathcal{I}(\boldsymbol{x}, y). \tag{6}$$

Using these observations, we define the influence-based appraisal function to be the sum of each training sample's influence:

$$\begin{aligned} f_{\mathrm{if}}(\mathcal{D}_{\mathrm{a}}) = -\frac{1}{|\mathcal{D}_{\mathrm{a}}| \cdot |\mathcal{D}_{\mathrm{te}}|} \sum_{(\boldsymbol{x}_{\mathrm{te}}, y_{\mathrm{te}})\in\mathcal{D}_{\mathrm{te}}} \sum_{(\boldsymbol{x},y)\in\mathcal{D}_{\mathrm{a}}} \\ \nabla_\theta L(\boldsymbol{x}_{\mathrm{te}}, y_{\mathrm{te}}; \hat{\theta})^\top \boldsymbol{H}_{\hat{\theta}}^{-1} \nabla_\theta L(\boldsymbol{x}, y; \hat{\theta}). \end{aligned} \tag{7}$$

We note that under our formulation, the influence is computed *forward* on unseen samples before training on them. It is assumed that for $x \in \mathcal{D}_{\mathrm{a}}$, $\mathcal{D}_{\mathrm{a}} \not\subset \mathcal{D}_{\mathrm{tr}}$, departing from the influence functions defined by Koh and Liang (2017). For interested readers, Appendix A incudes a set of key derivations; Section 5 and Appendix C discuss other influence functions.

Forward Influence in Multiparty Computation. Computing $f_{\mathrm{if}}(\mathcal{D}_{\mathrm{a}})$ requires computing and inverting empirical Hessian, usually a costly operation. For $\theta \in \mathbb{R}^d$ this requires $O(d^3)$ operations. Prior works suggest employing approximations for Hessian inverse vector product (Agarwal et al., 2017; Koh and Liang, 2017; Guo et al., 2020b). However, to evaluate mutliple candidate datasets for a given model, the inverse Hessian need only be computed once. In this way, the cost of computing and inverting $\boldsymbol{H}_{\hat{\theta}}$ can be amortized over many evaluations. Furthermore, this can be done in the clear by the model owner as it requires only $\hat{\theta}$ and $\mathcal{D}_{\mathrm{tr}}$. Computing the gradient of the loss on the test set can also be done in the clear, as no new data is required. Hence the term $\frac{1}{n_{\mathrm{te}}} \sum_{(\boldsymbol{x},y)\in\mathcal{D}_{\mathrm{te}}} \nabla_\theta L(\boldsymbol{x}, y; \hat{\theta})^\top \boldsymbol{H}_{\hat{\theta}}^{-1}$ may be precomputed by the model owner once in the clear and then encrypted. This leaves only a private computation of the loss gradient for each $\mathcal{D}_{\mathrm{a}}$ followed by an inner-product in $\mathbb{R}^d$. Because private computation tends to dominate the overall runtime, this yields considerable computational savings compared to private finetuning, as we will demonstrate in Section 4.1.

4 EXPERIMENTAL RESULTS

We aim to answer the following research questions:

1. In terms of runtime and usability in secure MPC, how do forward influence functions compare with finetuning and alternative data appraisal methods?

2. How robust is influence function-based appraisal under data corruption and class imbalance?

3. How effective is a greedy dataset selection strategy in which a model owner sequentially chooses to acquire the dataset with the highest influence function value?

We train and evaluate the model on classification problems using the MNIST (LeCun and Cortes, 1998) and CIFAR-10 (Krizhevsky, 2009) datasets: on MNIST, we classify ten digits, and on CIFAR-10, we distinguish planes from cars. Additionally, we verify our findings using Wisconsin diagnostic dataset for breast cancer (WDBC) (Dua and Graff, 2017). The examples consist of features computed from images of breast mass biopsies along with the target benign or malignant cancer diagnosis. The classification problem is solvable when 70% of the data is used for training (Agarap, 2018).

In each of the experiments, we fix the initial training model, including $\mathcal{D}_{\mathrm{tr}}$, $\mathcal{D}_{\mathrm{te}}$, and $\hat{\theta}$, and only intervene on the quality of the datasets to construct $\{\mathcal{D}_{\mathrm{a}}^{(p)}\}$, such that their ranking is salient. Prior to evaluating the appraisal functions on $\mathcal{D}_{\mathrm{a}}$, we train the model on the seed training set $\mathcal{D}_{\mathrm{tr}}$ until convergence to obtain $\hat{\theta}$.

We study three types of alterations on the datasets to simulate variations that are likely to arise in an open data market: (1) *label noise* in which the correct label of an example is changed with some non-zero probability; (2) *class imbalance* in which the marginal frequency of the labels varies between candidate datasets; and (3) *missing features* in which the candidate datasets vary in terms of which features they provide.

To simulate needing additional data, the initial model is trained on 1-10% of the available dataset, further seeded with a 9:1 imbalance in binary classifications.

The models are L2-regularized logistic regressors. To best approximate the optimal classifier, the baseline weights are obtained via L-BFGS (Liu and Nocedal, 1989). For ranking statistics, Spearman's Correlation Coefficient is used, denoted as ρ (Dodge, 2008).

Table 1: Correlation ρ of appraised values and data utility with varying amounts of label noise. Finetuning runtimes are limited to 1×, 4× and 16× of influence runtime, each benchmarked on the *best* performances under three learning rates: 0.001, 0.1, and 10. Hyperparameter tuning runtime for finetuning is excluded.

	Finetuning			**Influence**
learning rate	1×	4×	16×	1 epoch
0.001	0.61	0.58	0.72	
0.01	0.95	1.0	1.0	0.96
10	0.96	0.59	0.88	

4.1 In MPC, Forward Influence Functions Are More Usable Than Finetuning

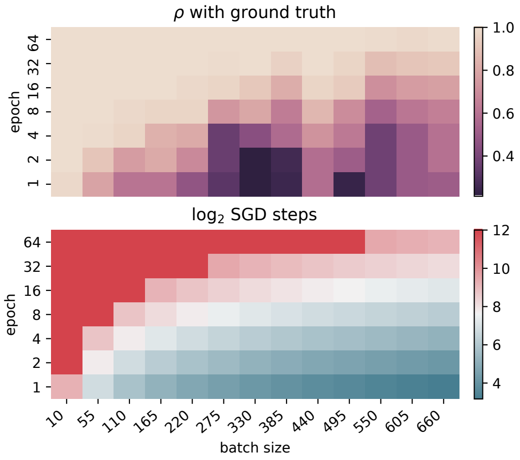

Figure 3: Correlation of appraisal with utility (top; purple is lower) and runtime (bottom; blue is faster) for finetuning hyperparameters batch size configuration (x-axis) and epochs (y-axis; logarithmic).

Influence Requires No Additional Hyperparameters. Although finetuning can approximate the test loss arbitrarily well, discovering the hyperparameters that achieve low error requires careful pre-tuning in the clear. In MNIST, small batch sizes and large epochs, as recommended for finetuning, often have high computational runtime (Table 1). Figure 3 summarizes the effect of finetuning hyperparameters on the correlation of appraisal with utility (top) and runtime (bottom). The hyperparameter selections in green result in few passes, but picking them will lead to sensitive rank correlation, thus requiring extensive tuning or scheduling. Meanwhile, safe hyperparameter settings tend to result in relatively large number of SGD passes. Both strategies incur significant computational cost. Lastly, even using the best batch size configurations, finetuning on noisy MNIST can fail to be competitive (Table 1).

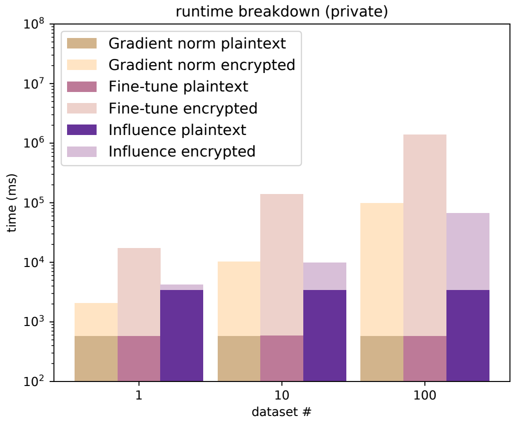

Figure 4: Log Scale Runtimes Spent On Plaintext And Encrypted Computations For All Three Appraisal Methods.

Influence Has Minimal Private Runtime. For any dataset, private influence performs a full-batch gradient step and a vector multiplication of dimension d for $\theta \in \mathbb{R}^d$. Thus, computing influence in private is comparable to that of finetuning with one SGD pass – the minimal without subsampling. In secure MPC, private runtimes tend to dominate as the number of evaluation grows. For a reasonable hyperparameter setting of 16 steps of full-batch gradient descent for fine-tuning, Figure 4 presents the total runtime of each appraisal function, separating the encrypted from the plaintext runtimes under plane-to-car setup. Due to influence functions' efficient setup with no additional hyperparameter, it trades a high one-time overhead for a convenient implementation that scales well in private.

4.2 Forward Influence Recovers Dataset Ranking Under Noise and Imbalance

We evaluate the efficacy of our data appraisals in two scenarios: (1) a scenario in which the utility of the data varies because of label noise in that data and (2) a scenario in which the utility varies because the data distribution does not match the distribution that the model owner is interested in.

Gradient Norm Is Insufficient. Despite their conceptual similarity, label noise and class imbalance are distinct corruptions that challenge naive, gradient-based methods. When gradient norm is used for appraisal, both datasets of poor balance (undesirable) and

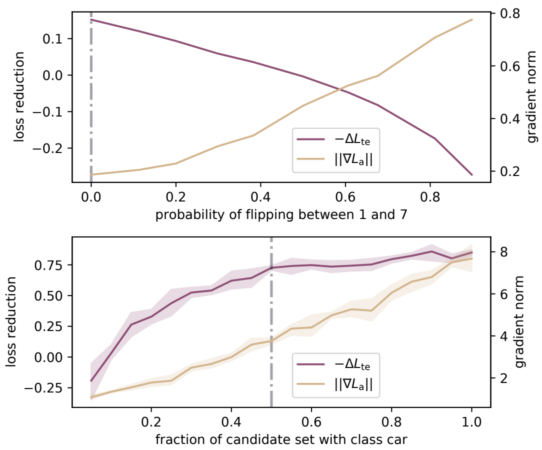

Figure 5: Gradient norm appraisal and test loss reduction as a function of MNIST label noise (top, $\rho = -1$) and CIFAR-10 plane-to-car class balance (bottom, $\rho = 1$).

datasets of low noise (desirable) would obtain similarly low numerical values. As shown in Figure 5, the gradient norm appraisal value (y-axis; note that the units vary per method) is monotonic over datasets under our two sets of experiments: label noise (x-axis) on MNIST (top) and data imbalance on CIFAR-10. The gradient norm curve (purple) aligns with risk reduction (yellow) under data imbalance, but crosses it under labels noise. Using only the norm of the gradient, though fast to compute, is an unreliable predictor for data value.

Label Noise. In our first scenario, we vary the utility of the dataset $\mathcal{D}_{\mathrm{a}}$ by introducing label noise. In particular, we use 1% of the MNIST training data as $\mathcal{D}_{\mathrm{tr}}$. The remaining training data is split into 10 candidate datasets $\mathcal{D}_{\mathrm{a}}^{(p)}$ with $p = 1, \ldots, 10$. For each of the candidate sets $\mathcal{D}_{\mathrm{a}}^{(p)}$, we randomly flip labels 1 and 7 with probability $p/10$. We evaluate models on $\mathcal{D}_{\mathrm{te}}$. Table 1 presents the correlation ρ of the label-noise probabilities with the appraisal value, including under three finetuning learning rates: 0.001, 0.1, and 10. The correlations are high for the model finetuning and influence function methods, suggesting that influence-based appraisal captures data utility.

Distribution Mismatch. In our second scenario, we focus on influence-based appraisal and study its efficacy under distribution mismatch. We simulate the mismatch between: (1) $\mathcal{D}_{\mathrm{tr}}$ and $\mathcal{D}_{\mathrm{te}}$ and (2) the candidate datasets $\mathcal{D}_{\mathrm{a}}^{(p)}$ by varying the prior over classes. To do so, we construct a training set from CIFAR-10 with a 10:1 ratio of plane-to-car and a balanced test set with a 1:1 ratio of plane-to-car. We then construct 20 candidate datasets $\mathcal{D}_{\mathrm{a}}^{(p)}$ of which exactly $(5 \cdot p)\%$ are planes and the remainder are cars, with $p = 1, \ldots, 20$.

Table 2: Influence Appraisal Correlation $\rho\pm\ \sigma$ With Data Utility on WDBC Over 10 Runs.

Corruption	Rank Correlation
None	0.880 ±0.081
Noise (Up to 1/5)	0.863 ±0.064
Noise (Up to 1/2)	0.844 ±0.106
Missing Features	0.810 ±0.213

The candidate datasets are of size $|\mathcal{D}_{\mathrm{a}}^{(p)}| = 440$. We repeat this process five times, sampling the datasets randomly each time.

Figure 6 shows scatter plots of: (a) the rank of the influence-based appraisal value, $f_{\mathrm{if}}(\mathcal{D}_{\mathrm{a}})$, of each of the 5×20 candidate datasets and (b) the rank of the utility or test accuracy of those datasets (see caption for details). The experimental results show that the influence-based appraisal value correlates well with gains in utility. Specifically, $f_{\mathrm{if}}(\mathcal{D}_{\mathrm{a}})$ allows the model owner to select a candidate dataset that closely resembles their desired distributions in most situations. However, zooming in on different ranges of class ratios (c-d), influence-based appraisal value $f_{\mathrm{if}}(\mathcal{D}_{\mathrm{a}})$ is becomes less informative when the class ratio deviates far from that of both the training and testing datasets.

4.3 Applying Influence Appraisal On Corrupted Cancer Patient Data

Real world applications often use passively gathered data of varying quality. Though the samples are not created for machine learning, they may be included for training. We simulate such a scenario with breast cancer detection from hospital screenings. We corrupt datasets by adding noise or removing features, and then apply influence-based appraisal to rank the datasets.

The first set of experiments concerns the rank correlation of datasets between forward influence functions and the ground truth losses, which trains $\mathcal{D}_{\mathrm{tr}} \cup \mathcal{D}_{\mathrm{a}}^{(p)}$ for all p to convergence. The same data is then corrupted. To simulate missing features, 10 columns are dropped (out of 30). Furthermore, we simulate label noise in candidate set, $\mathcal{D}_{\mathrm{a}}^{(p)}$, benign (positive) and malignant (negative) diagnoses are flipped under a binomial distribution of parameter $p/500$ and $p/200$ for $p = 1, \ldots, 100$.

Influence-based appraisal is able to inform the model owner the relative value in very noisy datasets. Figure 7 shows scatterplots of 100 datasets' evaluation (a) when all columns are retained. (b) when 10 feature columns are dropped, and (c-d) when labels are flipped with probability $p/500$ and $p/200$ for $\mathcal{D}_{\mathrm{a}}^{(p)}$. Table 2 shows rank correlation consistently above 80%. When all columns

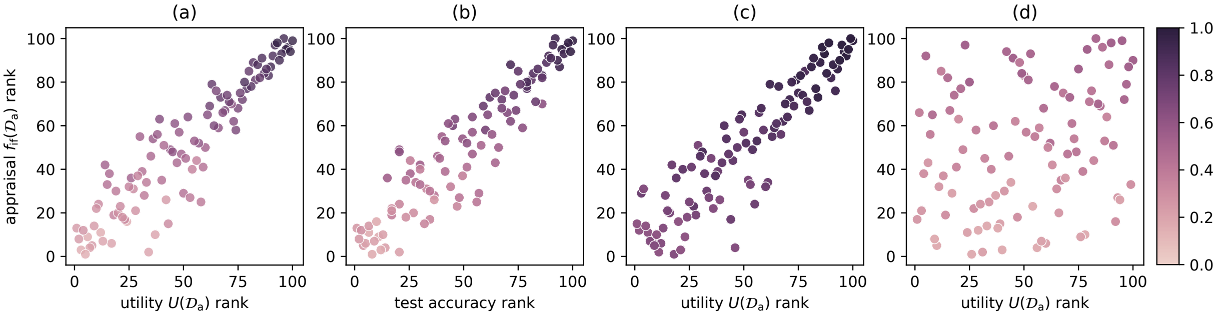

Figure 6: **Left a-b:** Rank of influence-based appraisal $f_{\text{if}}(\mathcal{D}_{\text{a}})$ (y-axis) as a function of the rank of the utility (a; $\rho{=}0.923$) and the test accuracy (b; $\rho{=}{-}0.927$) on CIFAR-10's plane-to-car dataset. **Right c-d:** Rank of $f_{\text{if}}(\mathcal{D}_{\text{a}})$ as a function of the rank of the utility on CIFAR-10 dataset for which the rate of cars is in the range $[0, 0.45]$ (c; $\rho{=}0.908$) and $[0.55, 1.0]$ (d; $\rho{=}0.247$). Each dot is a sampled dataset, colored according to the ratio of the undersampled class in $\mathcal{D}_{\text{a}}$.

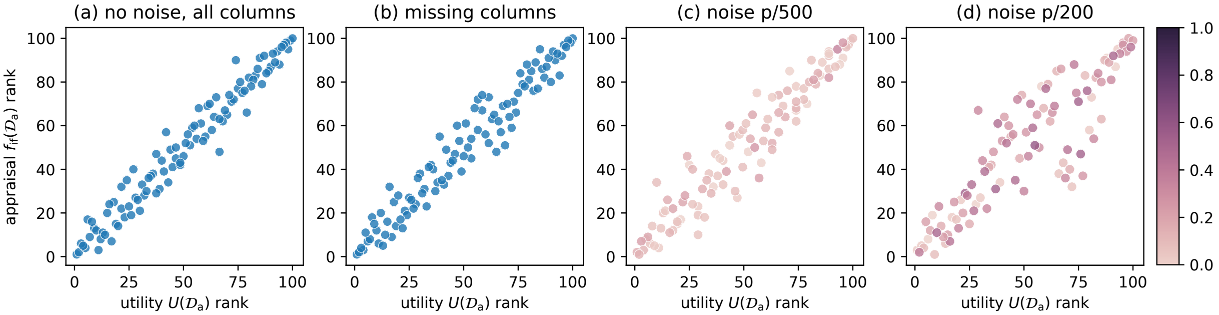

Figure 7: The rank of appraised values (y-axis) as a function of the rank data utility (x-axis) with varying data corruptions. The noiseless datasets (a-b) are benchmarked under 30 features and 20 features. The noisy datasets (c-d) are colored with noise level as a fraction of each dataset's label flips between "Benign" and "Malignant", and retain all features.

are preserved, the trained model can be used to identify helpful datasets. When 10 columns are missing, performance varies greatly; as the training set has less information about the problem, its second order landscape at convergence is less informative. Nevertheless, influence functions show robust ranking in the presence of missing features and noise.

In the second set of experiments we examine the loss dynamic from repeatedly using influence functions for data selection. Raj et al. (2020) proposes a strategy of data inclusion by selecting samples of the highest influence among a set of available candidates. In contrast to their setup where the candidates are existing training sets, samples in an open data markets that we simulate are often farther from the data distribution. Given a base model and 100 candidate datasets, two strategies are used in 15 iterations to select a dataset at a time, without replacement. Figure 8 shows the loss in varying noise, with 10 columns randomly dropped at each run. Despite the diverse seed models, the loss curves for greedy strategy based on influence (purple) often drops sooner than that of a random approach to selecting data. As more noise is injected to the candidate labels (c-d), influence consistently outperforms random selection, which is a strong baseline.

5 RELATED WORKS

We present two most similar lines of work. A more thorough treatment is included in the Appendix C.

Data Pricing in Federated Markets. Efficient private appraisals can especially aid federated learning settings where 1. privacy requirements are salient, and 2. the compute resources available *pre-transaction* are limited. In differential privacy and federated learning literature, Li et al. (2014); Song et al. (2019) and Wang et al. (2019, 2020) privately assess sets of data *after* the model is trained on them, while our solution does not require private training. Nevertheless, our approach to craft appraisal functions to suit privacy constraints complements recent works on acquisition strategies and Nash equilibria in emerging data markets (Azcoitia and Laoutaris, 2020; Pejó et al., 2018). Also under game-theoretic lens is computing Shapley values Shapley (1952) to assess training data for machine learning (Ghorbani and Zou, 2019; Jia et al., 2019; Azcoitia and Laoutaris, 2020; Azcoitia et al., 2020). A

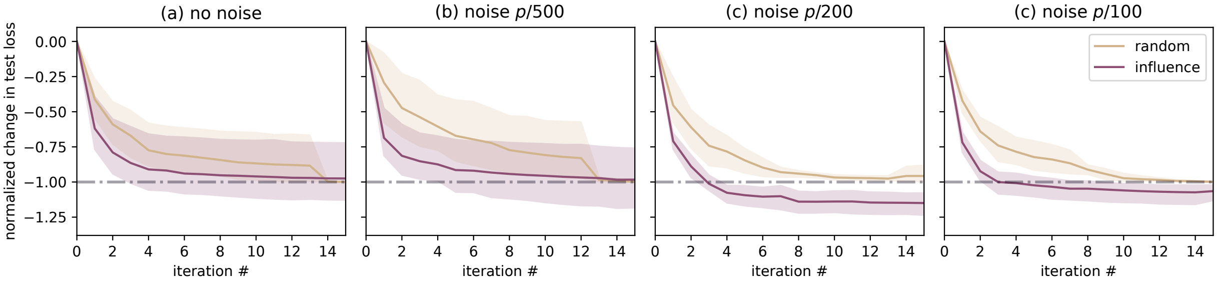

Figure 8: The change in test loss (y-axis) as a function of repeated rounds of data inclusion under varying noise levels. **Random**: choose a random dataset at each round. **Influence**: choose the dataset with the highest influence-based appraisal. For each graph, test loss change is normalized by the maxmimum test reduction in the control group. Averages and variances are taken over 5 runs.

primary motivation for using Shapley values is to enable equitable concurrent data assessment, while we focus on a limited scale where datasets are acquired one at a time. Indeed in sequential acquisition, a dataset acquired at a later stage of research may see its appraisal value lowered, if other datasets had reduced test loss. As a result, our appraisal incentivizes small-scale data owners to join the appraisal as early as possible.

Influence Functions. Measuring the effect of the data under leave-one-out training is known as Cook's distance in linear regression or the influence curve in regression residuals (Cook, 1977; Cook and Weisberg, 1982). Many contemporary works employ influence functions to explain existing training examples aposteri, applied to interpretability (Koh and Liang, 2017; Guo et al., 2020b), cross-validation (Giordano et al., 2019), poisoning attacks (Jagielski et al., 2021), and training data removal (Guo et al., 2020a; Koh et al., 2019). As a result, influence functions are usually 1. defined with respect to the trained model, 2. used to approximate parameter change under data removal. In contrast, we 1. use forward influence functions where the model has not seen the new data, concurrent to Raj et al. (2020)'s subsampling experiment for model selection and 2. applied to privately recover relative ranking. Incidentally, with the addition of MPC, we demonstrate a use case predicted by Giordano et al. (2019), where influence is chosen for our application where the Hessian inverse computation is a worthwhile tradeoff.

6 LIMITATIONS

Our procedure shows an appealing tradeoff between computation and privacy, but has limitations.

Recontruction Over Many Queries. While threshold-based appraisal limits the information leak to 1 bit, in theory, a strong adversary may reconstruct the data (or model) by observing appraisal values.

Descrimination of Arbitrary Data. Though f_{if} can discriminate quality differences despite corruptions, the choice of the model and $\mathcal{D}_{\text{a}}$ dictates a fundamental limit e.g. Figure 6d, when the class imbalance of $\mathcal{D}_{\text{tr}}$ and $\mathcal{D}_{\text{a}}$ cancels out. Moreover f_{if} is defined on a limited class of models: twice differentiable and convex in θ. Whether convexity can be relaxed in influence functions is its own active area of research (Basu et al., 2020a,b).

7 CONCLUSION

Our work presents fast and equitable data appraisal without data sharing, where a model owner can appraise another party's data without requiring any data (or model) sharing between the two parties. We craft efficient evaluations by leveraging secure MPC techniques to avoid private training. Three fast data appraisal implementations can operate in this setting: gradient norms, model finetuning, and forward influence functions. However, gradient norm contains too little information when faced with noisy and imbalanced data; finetuning becomes sensitive to hyper-parameters under privacy constraints. Our empirical results suggest that appraising data using influence function leads to accurate valuations in many scenarios, while requiring limited computation and no hyper-parameter optimization. Lastly, we demonstrate the practical effectiveness of influence-based appraisal in a breast cancer detection task with greedy, sequential data acquisition, which outperforms random selection under data corruptions. Future work focuses on broadening the applications of private data appraisal, including extending private data appraisal to more complex non-linear models with efficient inverse Hessian product approximations.

Acknowledgments

Research performed at Facebook AI Research, New York. The authors thank Brian Knott and Alex Melville for helpful discussions.

References

Agarap, A. F. M. (2018). On breast cancer detection: An application of machine learning algorithms on the wisconsin diagnostic dataset. In *Proceedings of the 2nd International Conference on Machine Learning and Soft Computing*, ICMLSC '18, pages 5–9, New York, NY, USA. ACM.

Agarwal, N., Bullins, B., and Hazan, E. (2017). Second-order stochastic optimization for machine learning in linear time. *The Journal of Machine Learning Research*, 18(1):4148–4187.

Amodei, D., Ananthanarayanan, S., Anubhai, R., Bai, J., Battenberg, E., Case, C., Casper, J., Catanzaro, B., Cheng, Q., Chen, G., et al. (2016). Deep speech 2: End-to-end speech recognition in english and mandarin. In *International conference on machine learning*, pages 173–182.

Ardila, R., Branson, M., Davis, K., Kohler, M., Meyer, J., Henretty, M., Morais, R., Saunders, L., Tyers, F., and Weber, G. (2020). Common voice: A massively-multilingual speech corpus. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 4218–4222.

Azcoitia, S. A. and Laoutaris, N. (2020). Try before you buy: A practical data purchasing algorithm for real-world data marketplaces. *arXiv preprint arXiv:2012.08874*.

Azcoitia, S. A., Paraschiv, M., and Laoutaris, N. (2020). Computing the relative value of spatio-temporal data in wholesale and retail data marketplaces. *arXiv preprint arXiv:2002.11193*.

Basu, S., Pope, P., and Feizi, S. (2020a). Influence functions in deep learning are fragile. *arXiv preprint arXiv:2006.14651*.

Basu, S., You, X., and Feizi, S. (2020b). On second-order group influence functions for black-box predictions. In *International Conference on Machine Learning*, pages 715–724. PMLR.

Cook, R. D. (1977). Detection of influential observation in linear regression. *Technometrics*, 19(1):15–18.

Cook, R. D. and Weisberg, S. (1982). *Residuals and influence in regression*. New York: Chapman and Hall.

CrypTen (2020). CrypTen (v 0.1). `https://github.com/facebookresearch/CrypTen`. Facebook AI Research.

Dodge, Y. (2008). *Spearman Rank Correlation Coefficient*, pages 502–505. Springer New York, New York, NY.

Dua, D. and Graff, C. (2017). UCI machine learning repository.

Evans, D., Kolesnikov, V., and Rosulek, M. (2017). A pragmatic introduction to secure multi-party computation. *Foundations and Trends® in Privacy and Security*, 2(2-3).

Ghorbani, A. and Zou, J. (2019). Data shapley: Equitable valuation of data for machine learning. In *International Conference on Machine Learning*.

Giordano, R., Stephenson, W., Liu, R., Jordan, M., and Broderick, T. (2019). A swiss army infinitesimal jackknife. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1139–1147.

Guo, C., Goldstein, T., Hannun, A., and van der Maaten, L. (2020a). Certified data removal from machine learning models. In *International Conference on Machine Learning*.

Guo, H., Rajani, N. F., Hase, P., Bansal, M., and Xiong, C. (2020b). Fastif: Scalable influence functions for efficient model interpretation and debugging. *arXiv preprint arXiv:2012.15781*.

Hampel, F. R. (1974). The influence curve and its role in robust estimation. *Journal of the american statistical association*, 69(346):383–393.

Jagielski, M., Severi, G., Pousette Harger, N., and Oprea, A. (2021). Subpopulation data poisoning attacks. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 3104–3122.

Jia, R., Dao, D., Wang, B., Hubis, F. A., Hynes, N., Gürel, N. M., Li, B., Zhang, C., Song, D., and Spanos, C. J. (2019). Towards efficient data valuation based on the shapley value. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1167–1176.

Knott, B., Venkataraman, S., Hannun, A., Sengupta, S., Ibrahim, M., and van der Maaten, L. (2020). Crypten: Secure multi-party computation meets machine learning.

Koh, P. W. and Liang, P. (2017). Understanding black-box predictions via influence functions. In *International Conference on Machine Learning*, pages 1885–1894.

Koh, P. W. W., Ang, K.-S., Teo, H., and Liang, P. S. (2019). On the accuracy of influence functions for measuring group effects. In *Advances in Neural Information Processing Systems*, pages 5254–5264.

Krause, A. and Golovin, D. (2014). Submodular function maximization. *Tractability*, 3:71–104.

Krause, A. and Guestrin, C. (2008). Beyond convexity: Submodularity in machine learning. *ICML Tutorials*.

Krause, A. and Horvitz, E. (2008). A utility-theoretic approach to privacy and personalization. In *AAAI*, volume 8, pages 1181–1188.

Krizhevsky, A. (2009). Learning multiple layers of features from tiny images.

LeCun, Y. and Cortes, C. (1998). The MNIST database of handwritten digits.

Li, C., Li, D. Y., Miklau, G., and Suciu, D. (2014). A theory of pricing private data. *ACM Transactions on Database Systems (TODS)*, 39(4):1–28.

Liu, D. C. and Nocedal, J. (1989). On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1):503–528.

Ludwig, H., Baracaldo, N., Thomas, G., Zhou, Y., Anwar, A., Rajamoni, S., Ong, Y., Radhakrishnan, J., Verma, A., Sinn, M., et al. (2020). Ibm federated learning: an enterprise framework white paper v0. 1. *arXiv preprint arXiv:2007.10987*.

Majkowska, A., Mittal, S., Steiner, D. F., Reicher, J. J., McKinney, S. M., Duggan, G. E., Eswaran, K., Cameron Chen, P.-H., Liu, Y., Kalidindi, S. R., et al. (2020). Chest radiograph interpretation with deep learning models: assessment with radiologist-adjudicated reference standards and population-adjusted evaluation. *Radiology*, 294(2):421–431.

Mohassel, P. and Zhang, Y. (2017). Secureml: A system for scalable privacy-preserving machine learning. In *2017 IEEE symposium on security and privacy (SP)*, pages 19–38. IEEE.

Munteanu, A., Schwiegelshohn, C., Sohler, C., and Woodruff, D. P. (2018). On coresets for logistic regression. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 6562–6571.

Ohrimenko, O., Schuster, F., Fournet, C., Mehta, A., Nowozin, S., Vaswani, K., and Costa, M. (2016). Oblivious multi-party machine learning on trusted processors. In *25th {USENIX} Security Symposium ({USENIX} Security 16)*, pages 619–636.

Pejó, B., Tang, Q., and Biczók, G. (2018). The price of privacy in collaborative learning. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 2261–2263.

Raj, A., Musco, C., Mackey, L., and Fusi, N. (2020). Model-specific data subsampling with influence functions. *arXiv preprint arXiv:2010.10218*.

SEAL (2020). Microsoft SEAL (release 3.5). `https://github.com/Microsoft/SEAL`. Microsoft Research, Redmond, WA.

Shapley, L. S. (1952). A value for n-person games. Technical report, Rand Corp Santa Monica CA.

Shen, L., Margolies, L. R., Rothstein, J. H., Fluder, E., McBride, R., and Sieh, W. (2019). Deep learning to improve breast cancer detection on screening mammography. *Scientific reports*, 9(1):1–12.

Song, T., Tong, Y., and Wei, S. (2019). Profit allocation for federated learning. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 2577–2586. IEEE.

Ting, D. and Brochu, E. (2018). Optimal subsampling with influence functions. *Advances in neural information processing systems*, 31.

Wang, G., Dang, C. X., and Zhou, Z. (2019). Measure contribution of participants in federated learning. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 2597–2604. IEEE.

Wang, T., Rausch, J., Zhang, C., Jia, R., and Song, D. (2020). A principled approach to data valuation for federated learning.

Supplementary Material: Data Appraisal Without Data Sharing

A Forward Influence Functions

We expand on the setup as well as re-hash the key steps for deriving *forward influence functions* in the context of empirical risk minimization. We start with the setup and derivation, and we finish with important comments on the impact of various assumptions in our setting.

Setup. Recall that the data is owned by two disparate parties: a *model owner*, who is developing the model, and a *data owner*, who possesses the dataset $\mathcal{D}_{\mathrm{a}}$ to be appraised. The model owner begins with a test set $\mathcal{D}_{\mathrm{te}}$ and their initial training set $\mathcal{D}_{\mathrm{tr}}$. Before acquiring the data $\mathcal{D}_{\mathrm{a}}$, the model owner wants a peek at the utility gain from updating θ to fit $\mathcal{D}_{\mathrm{tr}} \cup \mathcal{D}_{\mathrm{a}}$. The initial model parameters $\hat{\theta}$ are obstained by minimizing the regularized empirical risk on $\mathcal{D}_{\mathrm{tr}}$:

$$\hat{\theta} = \arg\min_{\theta} \sum_{(\boldsymbol{x},y)\in\mathcal{D}_{\mathrm{tr}}} L(\boldsymbol{x}, y; \theta) + \lambda \|\theta\|_2^2. \tag{8}$$

If the dataset $\mathcal{D}_{\mathrm{a}}$ were included, new parameters θ^* would be obtained by minimizing risk on dataset $\mathcal{D}_{\mathrm{tr}} \cup \mathcal{D}_{\mathrm{a}}$ instead. The value of concern is the utility of $\mathcal{D}_{\mathrm{a}}$, as evaluated on test loss:

$$U(\mathcal{D}_{\mathrm{a}}) := \frac{1}{|\mathcal{D}_{\mathrm{te}}|} \sum_{(\boldsymbol{x},y)\in\mathcal{D}_{\mathrm{te}}} L(\boldsymbol{x}, y; \hat{\theta}) - L(\boldsymbol{x}, y; \theta^*). \tag{9}$$

Derivation. Given Equation 8, we make a linear extrapolation:

$$U(\mathcal{D}_{\mathrm{a}}) \approx \frac{1}{|\mathcal{D}_{\mathrm{te}}|} \sum_{(\boldsymbol{x},y)\in\mathcal{D}_{\mathrm{te}}} \nabla_\theta L(\boldsymbol{x}, y; \hat{\theta}) \cdot (\hat{\theta} - \theta^*). \tag{10}$$

The model owner can compute the gradient of the model on the test set in plaintext. Because $L(\cdot)$ is twice differentiable, we have the empirical Hessian matrix associated with the training samples

$$\boldsymbol{H}_{\hat{\theta}} := \frac{1}{|\mathcal{D}_{\mathrm{tr}}|} \sum_{(\boldsymbol{x},y)\in\mathcal{D}_{\mathrm{tr}}} \nabla_\theta^2 L(\boldsymbol{x}, y, \hat{\theta}). \tag{11}$$

This Hessian and its associative inverse can also be computed in plaintext.

Suppose we upweigh a sample, $(\boldsymbol{x}_0, y_0)$, by an infinitesimal amount ϵ, and study the effect of this perturbation on the resulting model parameters. The associated loss is thus formulated as $\epsilon L(\boldsymbol{x}_0, y, \theta) + \sum_{(\boldsymbol{x},y)\in\mathcal{D}_{\mathrm{tr}}} L((\boldsymbol{x}, y, \theta)$. Training the new model till convergence to get new parameter θ^*, we can assume that the gradient of its loss is 0, or

$$\epsilon \nabla_\theta L(\boldsymbol{x}_0, y, \theta^*) + \sum_{(\boldsymbol{x},y)\in\mathcal{D}_{\mathrm{tr}}} \nabla_\theta L(\boldsymbol{x}, y, \theta^*) = 0. \tag{12}$$

We write the left hand side as an function of the new parameters, where

$$f(\theta^*) := \epsilon \nabla_\theta L(\boldsymbol{x}_0, y, \theta^*) + \sum_{(\boldsymbol{x},y)\in\mathcal{D}_{\mathrm{tr}}} \nabla_\theta L(\boldsymbol{x}, y, \theta^*). \tag{13}$$

We wish to find a relation between the parameters before and after the perturbation. To that end, denote the parameter difference $\Delta_\theta := \theta^* - \hat{\theta}$. The goal is to find a closed expression for Δ_θ, given the identity $f(\theta^*) = 0$.

As $\epsilon \to 0$, the new training set is just the original training data, or $\mathcal{D} \to \mathcal{D}_{\mathrm{tr}}$. The resulting model (from the non-perturbation), as we know, is optimal at $\hat{\theta}$. Therefore, the first two terms in the Taylor expansion of $f(\theta^*)$ around $\Delta_\theta = 0$ is $f(\theta^*) \approx f(\hat{\theta}) + f'(\hat{\theta}) \cdot \Delta_\theta$. We write

$$0 = f(\theta^*) \approx f(\hat{\theta}) + f'(\hat{\theta}) \cdot \Delta_\theta$$

Additionally, Equation 13 gives us

$$f(\hat{\theta}) := \epsilon \nabla_\theta L(\boldsymbol{x}_0, y, \hat{\theta}) + \sum_{(\boldsymbol{x},y)\in\mathcal{D}_{\mathrm{tr}}} \nabla_\theta L(\boldsymbol{x}, y, \hat{\theta}).$$

We thus obtain the approximation

$$\sum_{(\boldsymbol{x},y)\in\mathcal{D}_{\mathrm{tr}}} \nabla_\theta L(\boldsymbol{x}, y, \hat{\theta}) + \sum_{(\boldsymbol{x},y)\in\mathcal{D}_{\mathrm{tr}}} \nabla^2_\theta L(\boldsymbol{x}, y, \hat{\theta}) \cdot \Delta_\theta + \epsilon \nabla_\theta L(\boldsymbol{x}_0, y, \hat{\theta}) + \epsilon \nabla^2_\theta L(\boldsymbol{x}_0, y, \hat{\theta}) \cdot \Delta_\theta \approx 0. \tag{14}$$

Recall that on the original seed dataset $\mathcal{D}_{\mathrm{tr}}$, parameter $\hat{\theta}$ is optimal, so $\sum_{(\boldsymbol{x},y)\in\mathcal{D}_{\mathrm{tr}}} \nabla_\theta L((\boldsymbol{x}, y, \hat{\theta}) = 0$. This allows for a simplification:

$$\sum_{(\boldsymbol{x},y)\in\mathcal{D}_{\mathrm{tr}}} \nabla^2_\theta L(\boldsymbol{x}, y, \hat{\theta}) \cdot \Delta_\theta + \epsilon \nabla_\theta L(\boldsymbol{x}_0, y, \hat{\theta}) + \epsilon \nabla^2_\theta L(\boldsymbol{x}_0, y, \hat{\theta}) \cdot \Delta_\theta \approx 0. \tag{15}$$

Solving for Δ_θ approximately requires taking the inverse of the empirical Hessian (see discussion notes 1 and 4 for detailed discussion).

$$\big(|\mathcal{D}_{\mathrm{tr}}| \boldsymbol{H}_{\hat{\theta}} + \epsilon \nabla^2_\theta L(\boldsymbol{x}_0, y, \hat{\theta})\big) \cdot \Delta_\theta = -\epsilon \nabla_\theta L(\boldsymbol{x}_0, y, \hat{\theta}). \tag{16}$$

Multiply both sides with the scaled Hessian inverse

$$\big(1 + \frac{\epsilon}{|\mathcal{D}_{\mathrm{tr}}|} \boldsymbol{H}^{-1}_{\hat{\theta}} \nabla^2_\theta L(\boldsymbol{x}_0, y, \hat{\theta})\big) \cdot \Delta_\theta = -\frac{\epsilon}{|\mathcal{D}_{\mathrm{tr}}|} \boldsymbol{H}^{-1}_{\hat{\theta}} \nabla_\theta L(\boldsymbol{x}_0, y, \hat{\theta}). \tag{17}$$

Drop the term $\epsilon \nabla^2_\theta L(\boldsymbol{x}_0, y, \hat{\theta})$ (see discussion note 4), and take the derivate of both sides with respect to ϵ and write

$$\frac{\delta \Delta_\theta}{\delta \epsilon} = -\frac{1}{|\mathcal{D}_{\mathrm{tr}}|} \boldsymbol{H}^{-1}_{\hat{\theta}} \nabla_\theta L(\boldsymbol{x}_0, y, \hat{\theta}). \tag{18}$$

We thus obtain our influence formulation or $\mathcal{I}(\boldsymbol{x}, y) = -\boldsymbol{H}^{-1}_{\hat{\theta}} \nabla_\theta L(\boldsymbol{x}, y, \hat{\theta})$. Forward influence refers to its application on unseen data (see discussion note 2). Applying it to evaluate the change of loss given a particular dataset $\mathcal{D}_{\mathrm{a}}$ gives us the key appraisal component:

$$\mathcal{I}(\mathcal{D}_{\mathrm{a}}) = -\boldsymbol{H}^{-1}_{\hat{\theta}} \sum_{(\boldsymbol{x},y)\in\mathcal{D}_{\mathrm{a}}} \nabla_\theta L(\boldsymbol{x}, y, \hat{\theta}), \tag{19}$$

before scaling (by set cardinality) and combining with Equation 10.

Discussion. Note 1. strong convexity is usually assumed (Koh and Liang, 2017), so that the Hessian matrix is positive definite. This is a stronger assumption than necessary, only the empirical Hessian with respect to the combined dataset needs to be positive-definite. In practice, we assume convexity and use regularization when inverting the Hessian (alternatively, a pracitioner may implement the numerical function to avoid inverting the Hessian altogether), so the method can be potentially applied to problems when the Hessian is not positive definite.

Note 2. In machine learning literature, influence functions typically assume $(\boldsymbol{x}_0, y_0)$ to be part of the training data. Here we are using the numerical form of the result, but applying the extrapolation to new data $\mathcal{D}_{\mathrm{a}}$, hence it is referred to as a forward influence. A mismatched data construction is standard technique in the construction of influence functions (Hampel, 1974; Giordano et al., 2019). We especially study the impact of this mismatch in Figure 6c-d.

Note 3. The Taylor Expansions' validity likely matters little in application, but it is worth mentioning that the loss function is preferred to be second-order smooth. The truncation error, on the other hand, is only studied in Basu et al. (2020b), interacted with non-convexity.

Note 4. Additionally, dropping the term $\epsilon \nabla^2 L(\boldsymbol{x}_0, y, \hat{\theta})$ from the first order expansion is effectively approximating the gradient on the new data point with the gradient of the previous model, which may not be bounded. This approximation is also present in the usual influence definition.

B Experiment Hyper-parameters

We note the hyper-parameters relevant to our implementation and evaluation.

CrypTen The software is implemented using PyTorch defaults, with only the precision number changed to 24 bits.

CIFAR-10 PlaneCar Baseline model is trained with L-BFGS at 1000 iterations, and learning rate $1e-4$ to ensure convergence. Each candidate data is sized 440. For fine-tuning, L2 regularization set to $1e-3$, learning rate 0.1, which are assumed to be from hyper-parameter search conducted by the model owner on their existing training data $\mathcal{D}_{\mathrm{tr}}$. To inject class imbalance and label noise, there are 20 uniform values between 0 and 1. Each perturbation is repeated 5 times, generating 100 sample datasets. Over 5 runs, influence achieves an average correlation of 90.7% .

WDBC Baseline model is trained with L-BFGS at 1000 iterations, and learning rate $1e-4$ to ensure convergence. Each candidate data is sized 30, and 100 datasets are sampled to make comparison. L2 regularization is set to $1e-3$ though the experiments reproduced at 0.1 have similar performance. For injecting label noise, there are noise levels 0, 0.002, 0.005, and 0.10, representing the portion of flipped labels (benign to malign or malign to benign). Each experiment is repeated 10 times.

C Related Works (Extended)

While we uniquely propose adding a private appraisal stage pre-transaction, our approaches draw from a long line of research. We now discuss works that tackle similar incentive problems in model-training, particularly between model and data owners. In this section, we extendedly discuss the commonalities, differences, and potential future application in the context of related works.

C.1 Private Data Appraisal in Federated Learning Scenarios

Private data appraisal is studied with differential privacy (Li et al., 2014) and federated learning (Song et al., 2019; Wang et al., 2019, 2020). Similar to our setup, each data contributor deserves a payout based on the quality and quantity of data contributed. Because private data is valuable, distrusting parties would rather not reveal their data in plaintext before the payout. Private training and assessment therefore dominate the propsed approaches. In a typical federation setup, the goal is to assess multiple sets of data *after* the model is trained on them. The exchange of data is treated as foregone conclusion, with the private pricing serving as a mechanism to incentivize more data contributions.

In contrast, our setup dictates that the assessment necessarily predates the decisions to include the training data. This lets the model and data owners decide if it is worth engaging in the transaction based on the appraised value. Finally, our proposed influence-based appraisals are computed without incurring the computational intensity of training. While their methods often involve repeated training in private, the computational intensity of private influence computation is approximately that of only one pass of private gradient updates. Private appraisals can indeed be applied to federated learning. However our work assumes a simpler ownership model (where one party performs training), and decidedly procures one dataset at a time. Influence-based appraisal is fast, yet it is ultimately an approximation with no guarantee of absolute fairness. Yet, shifting federated learning procedures by adding an appraisal stage induces three advantages: 1. our method, along with its associated privacy and computation costs, is calibrated for acquiring unseen data, therefore saving a lot of training time on potentially low-quality data, and 2. our appraisal leads to an added incentive by effectively rewarding early adopters, which can be particularly useful for new markets. 3. our MPC methods afford multiple parties, and sequential acquisition readily scales linearly with the number of datasets and parties.

C.2 Data Exchange Through Game-Theorectic Lens

Building an efficient data market for machine learning has been theorized in many research communities. While we design a solution for low-resource model owners with MPC, other works focus on the economic theories exacting equity between large data contributors.

Data Market Games. A primary motivation for our work is to enable efficient data markets for low-resource projects, similar to utility and privacy tradeoff theorized by Krause and Horvitz (2008).

Many have specifically surmised the rise of a data market for the sole purpose of trading training data for machine learning models. We expand on that premise by realizing an efficient privacy-perserving appraisal by applying multi-party computation, solving primarily the incentives problems in a noisy market. However, our focus on crafting the appraisal to suit privacy constraints only fills in a small part of the whole puzzle; Pejó et al. (2018) uses privacy price to factor in contextual privacy desires from participating parties, applied to whether two parties are incentived to train together. Additionally, Azcoitia and Laoutaris (2020) proposes Try Before You Buy, by supposing heuritstic evaluations that can be of linear runtime with respect to the number of data owners. They further prove the efficiency of various acquisition strategies. Our work enables these strategies by improving privacy incentives.

Shapley Values. Over concurrent datasets, Shapley values from Shapley (1952) have been proposed as an equitable method for data appraisal (Ghorbani and Zou, 2019; Jia et al., 2019; Azcoitia and Laoutaris, 2020; Azcoitia et al., 2020). A primary motivation of Shapley values over influence-based approaches is the invariance to the order of data aquisition. Instead, we focus on the case where the order of acquisition is important, as earlier acquisition decisions may justifiably affect the perceived value of data that arrive later. We thus pursue valuation techniques based on leave-one-out training.

Additionally, evaluating data owners one-by-one creates favorable incentives. Suppose a data owner fears similar or duplicate data to be available, which will render their data less useful if included prior to evaluation, the data owner may be eager to participate early. This incentive may be especially useful for low resource settings.

C.3 Influence Functions

Assessing the impact of data to a statistical model is by itself a long studied subject. A natural method defines the impact through leave-one-out training. Measuring the effect of the data under leave-one-out training is known as Cook's distance in linear regression (Cook, 1977) or the influence curve in regression residuals (Cook and Weisberg, 1982). Many contemporary works employ influence functions to explain existing training examples aposteri, including for interpretability (Koh and Liang, 2017; Guo et al., 2020b), efficient cross-validation (Giordano et al., 2019), poisoning attacks (Jagielski et al., 2021), and efficient training data removal (Guo et al., 2020a; Koh et al., 2019). As a result, influence functions are usually 1. defined with respect to the trained model, 2. used to approximate parameter change under data removal. For non-convex models, Basu et al. (2020a) finds the approximation errors for those influence functions sensitive to depth, regularization, and data composition, part of which Basu et al. (2020b) mitigates by expanding influence to include second order terms in the approximation.

A few works effectively use influence to appraise datasets prior to training. Evaluating the expected utility of training examples is instrumental to efficient data exchanges, where existing model and additional data belong to separate individuals. Most recently, Raj et al. (2020) applies forward influence functions to quickly select unseen training samples for model selection. They find that for sufficient training data, the first order approximation of the model's test loss through (forward) influence functions is valid, under convexity, smoothness, good regularization, and bounded gradients. Like our setup, Raj et al. (2020) seeks to evaluate between candidate examples prior to training, approximating the data's relative importance rather than predicting the exact parameter change or losses. Our work expands the method to perform data evaluation and selection in private, achieving computational gains against retraining in secure MPC. Finally, we are primarily concerned with enabling data transaction rather than active learning. While experiments in Raj et al. (2020) select a pool of samples from uncorrupted training data, our setup adds substaintial noise to the candidate data sets to simulate an open data market.

Challenges to Influence-Based Approximations Even though our influence-based methods can be apllied to deep models trained on non-convex losses, applying it as is may impact accuracy. (Koh and Liang, 2017; Basu et al., 2020a,b) study post-training influence functions with deep models trained with non-convex loss in deep models, showing that they are both empirically useful, yet also fragile. Their influence functions are found to be sensitive to hyperparameters, such as architecture and regularization strength, and particularly reliant on convex loss and shallow networks (Basu et al., 2020a). Additionally, for groups of data, the makeup of the group and its size affect the approximation error (Koh et al., 2019; Basu et al., 2020b).

Nevertheless, Basu et al. (2020b,a) acknowledge that after summing up a set of influences, as we do, peculiarities in individual samples' influence approximations matter less. Furthermore, in our data market application, it is only desired that influence functions retain the value ranking among potential datasets under the realistic constraints, such as noise, class imbalance, and missing data. As the candidate dataset size and model architecture are assumed constant, as both belong to the model owner, group size and inter-architecture differences that make influence functions fickle become irrelavant. Finally, the datasets to evaluate under forward influence are often not part of the training set. This novel use case lets influence functions differentiate between data sets that may diverge greatly from the initial training and testing sets, for which they are empirically informative.

C.4 Submodular Optimization and Coreset Selection

Optimal dataset selection is a combinatorial search where the optimal solutions follow a diminishing return curve. We hereby describe a connection between our greedy evaluation and submodular optimization.

Utility maximization over candidate datasets is submodular: when no new data is selected $\mathcal{D}_\mathrm{a} = \emptyset$, the utility $U(\mathcal{D}_\mathrm{a}) = 0$; when similar data is included in the existing training set, the machine learning model often needs it less. However, the probem is in general NP-hard, thus intractable. Existing works in submodular optimization give a 3/4 optimality for greedy solutions under positivity where $U(\mathcal{D}_\mathrm{a}) > 0$ for $\mathcal{D}_\mathrm{a} \neq \emptyset$ (Krause and Golovin, 2014); unfortunately in an open market, the positivity assumption is not practical, as it requires that we exclude potential adversarial data poisoning altogether. Thus, our work does not follow submodular optimization; nevertheless, submodularity affords alternative direction to convexity to study the bounds of using threshold-based influence functions in more generic machine learning models. For that purpose, we direct interested readers to Krause and Horvitz (2008); Krause and Guestrin (2008).

For actively selecting unseen data, a closely related problem looks at using gradient information for subset selection by deriving a scaler, when evaluating the empirical risk minimization with every data set is impractical (Munteanu et al., 2018; Raj et al., 2020). Influence functions, especially the additive variety akin to Koh and Liang (2017); Giordano et al. (2019)'s first order formulation, can be used as an alternative to ranking datasets without the computation (Raj et al., 2020; Ting and Brochu, 2018). In particular, Raj et al. (2020) suggests greedy additions of top-ranking training samples using influence functions to fast iterate over model selection process. However, a coreset selection framework is more approproiate for choosing multiple sets of data. Instead, we focus on selecting just one dataset at a time, which ignores the interactions between different candidate datasets.

References

Agarap, A. F. M. (2018). On breast cancer detection: An application of machine learning algorithms on the wisconsin diagnostic dataset. In *Proceedings of the 2nd International Conference on Machine Learning and Soft Computing*, ICMLSC '18, pages 5–9, New York, NY, USA. ACM.

Agarwal, N., Bullins, B., and Hazan, E. (2017). Second-order stochastic optimization for machine learning in linear time. *The Journal of Machine Learning Research*, 18(1):4148–4187.

Amodei, D., Ananthanarayanan, S., Anubhai, R., Bai, J., Battenberg, E., Case, C., Casper, J., Catanzaro, B., Cheng, Q., Chen, G., et al. (2016). Deep speech 2: End-to-end speech recognition in english and mandarin. In *International conference on machine learning*, pages 173–182.

Ardila, R., Branson, M., Davis, K., Kohler, M., Meyer, J., Henretty, M., Morais, R., Saunders, L., Tyers, F., and Weber, G. (2020). Common voice: A massively-multilingual speech corpus. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 4218–4222.

Azcoitia, S. A. and Laoutaris, N. (2020). Try before you buy: A practical data purchasing algorithm for real-world data marketplaces. *arXiv preprint arXiv:2012.08874*.

Azcoitia, S. A., Paraschiv, M., and Laoutaris, N. (2020). Computing the relative value of spatio-temporal data in wholesale and retail data marketplaces. *arXiv preprint arXiv:2002.11193*.

Basu, S., Pope, P., and Feizi, S. (2020a). Influence functions in deep learning are fragile. *arXiv preprint arXiv:2006.14651*.

Basu, S., You, X., and Feizi, S. (2020b). On second-order group influence functions for black-box predictions. In *International Conference on Machine Learning*, pages 715–724. PMLR.

Cook, R. D. (1977). Detection of influential observation in linear regression. *Technometrics*, 19(1):15–18.

Cook, R. D. and Weisberg, S. (1982). *Residuals and influence in regression.* New York: Chapman and Hall.

CrypTen (2020). CrypTen (v 0.1). `https://github.com/facebookresearch/CrypTen`. Facebook AI Research.

Dodge, Y. (2008). *Spearman Rank Correlation Coefficient*, pages 502–505. Springer New York, New York, NY.

Dua, D. and Graff, C. (2017). UCI machine learning repository.

Evans, D., Kolesnikov, V., and Rosulek, M. (2017). A pragmatic introduction to secure multi-party computation. *Foundations and Trends® in Privacy and Security*, 2(2-3).

Ghorbani, A. and Zou, J. (2019). Data shapley: Equitable valuation of data for machine learning. In *International Conference on Machine Learning*.

Giordano, R., Stephenson, W., Liu, R., Jordan, M., and Broderick, T. (2019). A swiss army infinitesimal jackknife. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1139–1147.

Guo, C., Goldstein, T., Hannun, A., and van der Maaten, L. (2020a). Certified data removal from machine learning models. In *International Conference on Machine Learning*.

Guo, H., Rajani, N. F., Hase, P., Bansal, M., and Xiong, C. (2020b). Fastif: Scalable influence functions for efficient model interpretation and debugging. *arXiv preprint arXiv:2012.15781*.

Hampel, F. R. (1974). The influence curve and its role in robust estimation. *Journal of the american statistical association*, 69(346):383–393.

Jagielski, M., Severi, G., Pousette Harger, N., and Oprea, A. (2021). Subpopulation data poisoning attacks. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 3104–3122.

Jia, R., Dao, D., Wang, B., Hubis, F. A., Hynes, N., Gürel, N. M., Li, B., Zhang, C., Song, D., and Spanos, C. J. (2019). Towards efficient data valuation based on the shapley value. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1167–1176.

Knott, B., Venkataraman, S., Hannun, A., Sengupta, S., Ibrahim, M., and van der Maaten, L. (2020). Crypten: Secure multi-party computation meets machine learning.

Koh, P. W. and Liang, P. (2017). Understanding black-box predictions via influence functions. In *International Conference on Machine Learning*, pages 1885–1894.

Koh, P. W. W., Ang, K.-S., Teo, H., and Liang, P. S. (2019). On the accuracy of influence functions for measuring group effects. In *Advances in Neural Information Processing Systems*, pages 5254–5264.

Krause, A. and Golovin, D. (2014). Submodular function maximization. *Tractability*, 3:71–104.

Krause, A. and Guestrin, C. (2008). Beyond convexity: Submodularity in machine learning. *ICML Tutorials.*

Krause, A. and Horvitz, E. (2008). A utility-theoretic approach to privacy and personalization. In *AAAI*, volume 8, pages 1181–1188.

Krizhevsky, A. (2009). Learning multiple layers of features from tiny images.

LeCun, Y. and Cortes, C. (1998). The MNIST database of handwritten digits.

Li, C., Li, D. Y., Miklau, G., and Suciu, D. (2014). A theory of pricing private data. *ACM Transactions on Database Systems (TODS)*, 39(4):1–28.

Liu, D. C. and Nocedal, J. (1989). On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1):503–528.

Ludwig, H., Baracaldo, N., Thomas, G., Zhou, Y., Anwar, A., Rajamoni, S., Ong, Y., Radhakrishnan, J., Verma, A., Sinn, M., et al. (2020). Ibm federated learning: an enterprise framework white paper v0. 1. *arXiv preprint arXiv:2007.10987*.

Majkowska, A., Mittal, S., Steiner, D. F., Reicher, J. J., McKinney, S. M., Duggan, G. E., Eswaran, K., Cameron Chen, P.-H., Liu, Y., Kalidindi, S. R., et al. (2020). Chest radiograph interpretation with deep learning models: assessment with radiologist-adjudicated reference standards and population-adjusted evaluation. *Radiology*, 294(2):421–431.

Mohassel, P. and Zhang, Y. (2017). Secureml: A system for scalable privacy-preserving machine learning. In *2017 IEEE symposium on security and privacy (SP)*, pages 19–38. IEEE.

Munteanu, A., Schwiegelshohn, C., Sohler, C., and Woodruff, D. P. (2018). On coresets for logistic regression. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 6562–6571.

Ohrimenko, O., Schuster, F., Fournet, C., Mehta, A., Nowozin, S., Vaswani, K., and Costa, M. (2016). Oblivious multi-party machine learning on trusted processors. In *25th {USENIX} Security Symposium ({USENIX} Security 16)*, pages 619–636.

Pejó, B., Tang, Q., and Biczók, G. (2018). The price of privacy in collaborative learning. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 2261–2263.

Raj, A., Musco, C., Mackey, L., and Fusi, N. (2020). Model-specific data subsampling with influence functions. *arXiv preprint arXiv:2010.10218*.

SEAL (2020). Microsoft SEAL (release 3.5). `https://github.com/Microsoft/SEAL`. Microsoft Research, Redmond, WA.

Shapley, L. S. (1952). A value for n-person games. Technical report, Rand Corp Santa Monica CA.

Shen, L., Margolies, L. R., Rothstein, J. H., Fluder, E., McBride, R., and Sieh, W. (2019). Deep learning to improve breast cancer detection on screening mammography. *Scientific reports*, 9(1):1–12.

Song, T., Tong, Y., and Wei, S. (2019). Profit allocation for federated learning. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 2577–2586. IEEE.

Ting, D. and Brochu, E. (2018). Optimal subsampling with influence functions. *Advances in neural information processing systems*, 31.

Wang, G., Dang, C. X., and Zhou, Z. (2019). Measure contribution of participants in federated learning. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 2597–2604. IEEE.

Wang, T., Rausch, J., Zhang, C., Jia, R., and Song, D. (2020). A principled approach to data valuation for federated learning.