Learning to Limit Data Collection via Scaling Laws: A Computational Interpretation for the Legal Principle of Data Minimization

Learning to Limit Data Collection via Scaling Laws: A Computational Interpretation for the Legal Principle of Data Minimization

Divya Shanmugam
divyas@mit.edu
Massachusetts Institute of Technology
Cambridge, MA, USA

Fernando Diaz*
diazf@acm.org
Microsoft Research
Montreal, Canada

Samira Shabanian
samira.shabanian@microsoft.com
Microsoft Research
Montreal, Canada

Michèle Finck
m.finck@uni-tuebingen.de
University of Tübingen
Germany

Asia J. Biega
asia.biega@mpi-sp.org
Max Planck Institute for Security and Privacy
Bochum, Germany

ABSTRACT

Modern machine learning systems are increasingly characterized by extensive personal data collection, despite the diminishing returns and increasing societal costs of such practices. Yet, data minimisation is one of the core data protection principles enshrined in the European Union's General Data Protection Regulation ('GDPR') and requires that only personal data that is adequate, relevant and limited to what is necessary is processed. However, the principle has seen limited adoption due to the lack of technical interpretation.

In this work, we build on literature in machine learning and law to propose **FIDO**, a **F**ramework for **I**nhibiting **D**ata **O**vercollection. FIDO learns to limit data collection based on an interpretation of data minimization tied to system performance. Concretely, FIDO provides a data collection stopping criterion by iteratively updating an estimate of the *performance curve*, or the relationship between dataset size and performance, as data is acquired. FIDO estimates the performance curve via a piecewise power law technique that models distinct phases of an algorithm's performance throughout data collection *separately*. Empirical experiments show that the framework produces accurate performance curves and data collection stopping criteria across datasets and feature acquisition algorithms. We further demonstrate that many other families of curves systematically *overestimate* the return on additional data. Results and analysis from our investigation offer deeper insights into the relevant considerations when designing a data minimization framework, including the impacts of active feature acquisition on individual users and the feasability of user-specific data minimization. We conclude with practical recommendations for the implementation of data minimization.

*Now at Google.

FAccT '22, June 21–24, 2022, Seoul, Republic of Korea

ACM ISBN 978-1-4503-9352-2/22/06.
https://doi.org/10.1145/3531146.3533148

CCS CONCEPTS

• **Social and professional topics** → **Governmental regulations**; • **Applied computing** → **Law**; • **Computing methodologies** → *Feature selection*; Online learning settings.

ACM Reference Format:
Divya Shanmugam, Fernando Diaz, Samira Shabanian, Michèle Finck, and Asia J. Biega. 2022. Learning to Limit Data Collection via Scaling Laws: A Computational Interpretation for the Legal Principle of Data Minimization. In *2022 ACM Conference on Fairness, Accountability, and Transparency (FAccT '22), June 21–24, 2022, Seoul, Republic of Korea.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3531146.3533148

1 INTRODUCTION

Data minimisation is a core principle of the European Union's Data Protection Regulation [16], as well as data protection laws in other jurisdictions:

> "Personal data shall be: [...] adequate, relevant and limited to what is necessary in relation to the purposes for which they are processed (data minimisation)"

The requirement serves as a guideline for respectfully processing data. Recent empirical research has shown that it is possible to replicate the performance of data-driven systems with significantly less data [4, 9, 37, 39]. These findings are a consequence of the diminishing returns that data collection exhibits across applications and domains [22, 29, 36]. Recognizing that limiting data is possible, legal guidelines point to algorithmic techniques that could be incorporated into minimization pipelines, including feature selection [5] or examination of learning curves [10].

Yet, despite the existence of numerous algorithmic techniques that could be adapted to comply with the minimization requirement, the data minimisation principle has received little attention from the computer science community to date. As noted by scholars reviving discussion about the principle, a dearth of concrete mathematical definitions and guidelines is one of the main factors inhibiting adoption [3]. Indeed, qualitative research has shown a lack of consistent data minimization standards or an understanding of the principle among software developers [35].

Recent interpretations of data minimization propose to tie the data collection purpose in data-driven systems to performance metrics, an interpretation termed *performance-based data minimization* [3, 4]. Our work follows this interpretation, addressing the question of *how to proactively satisfy the performance-based data minimization principle in machine learning with personal data.*

Contributions. In this work, we propose FIDO, a Framework for Inhibiting Data Overcollection, and demonstrate how ongoing personal data collection could be approached in the context of data minimization.

FIDO's key conceptual proposal is to *adaptively learn an algorithm's performance curve* so that an *appropriate data collection stopping point* can be determined accurately. By modeling the performance curve directly, the framework remains flexible to different underlying feature acquisition algorithms and definitions of performance. Experiments involving multiple datasets in the recommender systems domain validate this flexibility and demonstrate FIDO's ability to estimate algorithm performance given additional data accurately *without* collecting more data.

The core technical insight FIDO contributes lies in its performance curve estimation procedure, which builds on recent work in machine learning literature that identifies three distinct phases in the performance curves of learning algorithms: the small data phase, the power law phase, and the diminishing returns phase [22]. We demonstrate that these performance phases can be observed in the context of user data collection and provide a technique to model the data collection phases directly by adaptively learning a piecewise power law curve. Empirical experiments show that other approaches to estimate performance curves systematically *overestimate* the return on additional data. Modeling each phase directly allows FIDO to not only learn the performance curve more accurately as data is acquired but also satisfy practical constraints of data minimization.

Finally, we examine issues related to user-level data minimization. We demonstrate the impacts that algorithmic components such as active feature acquisition (a technique to intelligently select which feature values to acquire) might have on minimization outcomes. We find that active feature acquisition can lead to unequal, concentrated data collection from a small set of users, in addition to decreased minimization performance for evolving user communities or when sensitive features are excluded from initial data collection. We also find that user-specific performance curves are highly variable, where the collection of more data can often result in a *decrease* in user-specific performance.

This paper seeks to offer a computational perspective on the GDPR's principle of data minimization, contributing insights to the ongoing discussions about the technical implementation of this core data protection principle, and to provide recommendations to practitioners and scholars moving forward.

2 LEGAL BACKGROUND

Data minimization is one of the core principles of European data protection law. In recent years, many have questioned its suitability in face of technical advances based on the repurposing of large quantities of data[28]. Some indeed fear that adherence to the principle 'would sacrifice considerable social benefit' as it may limit the innovative potential of machine learning [30]. Even so, data minimization remains one of the principles that ought to be respected regardless of the specific context of personal data processing. Controversies around the principle will continue as the adoption of the draft Data Governance and Data Acts would force discussions as to how to reconcile related legislative incentives to process more (personal) data with data minimization. In this paper, we discuss whether it is at all possible to reconcile data minimization with machine learning.

Article 5(1)(c) GDPR provides that data shall be 'adequate, relevant and limited to what is necessary in relation to the purposes for which they are processed'. Article 25(2) GDPR reiterates that controllers only process personal data 'necessary for each specific purpose of the processing'. First, the data processed must be 'relevant', meaning that only pertinent data ought to be processed. This is designed to safeguard against the accumulation of data for the sake of gathering more data for undisclosed ends and stands in tension with the contemporary operation of ML systems, which often re-purpose data. Second, data can only be processed where it is adequate. Although this requirement is closely intertwined with relevance, adequacy is different in that it may sometimes require that more, not less, data is processed, such as where existing data is inadequate to draw inferences about demographic groups underrepresented in a dataset. Third, only necessary data ought to be processed, meaning that data controllers need to identify the minimum amount of data necessary to fulfill the purpose [3]. Beyond, there remain unresolved questions regarding the interpretation of data minimization, such as whether data minimization also requires the pseudonymisation of personal data and whether it implies that preference should be given to ordinary personal data over sensitive data [3]. Either way, it is worth pointing out that compliance with data minimization can also enhance the quality of ML as there is less need to clean the data and less risk of inaccuracy [10].

Data minimization in data-driven systems has been hindered by the lack of concrete computational formalizations, as the principle has received less academic attention in the computing community compared to fairness or transparency. While legal requirements leave room for interpretation, regulatory bodies then issue more specific guidelines to help translate these requirements into practice (guidelines for the implementation of data minimization in Machine Learning and Artificial Intelligence have been issued by the data protection authorities in the UK [5, 24, 25] or Norway [10]). Often, these guidelines mention potential techniques or implementation directions but are still not concrete enough to offer specific mathematical definitions or algorithms (which may lead to vastly varying implementations in practice [35]). The paper addresses this gap, exploring the feasibility of an interpretation based on algorithmic performance curves.

3 RELATED WORK

Ideas related to the legal concept of data minimization exist across fields in machine learning. We discuss past work on data minimization and performance curves below, and expand on intersections

with the literature on sample complexity and active feature acquisition in the supplementary material.

3.1 Perspectives on Data Minimization.

Our work follows a number of recent efforts to formalize the principle of data minimization. Biega et al. [4] propose an interpretation of data minimization that ties data collection purpose to system performance and discuss the feasability of data minimization in recommendation systems. Our work deepens this interpretation, noting that past work on privacy by design highlights data collection as a key area to implement data minimization. More specifically, we propose a learning-based framework to enforce a data collection stopping criterion, in addition to novel definitions of stopping criteria related to the returns on additional data, rather than absolute model performance.

Existing guidelines distinguish between *breadth-based* data minimization and *depth-based data minimization* [1]. In the former, one aims to minimize the number of features, while the latter concerns minimizing the overall amount of data collected for one data modality. Rastegarpanah et al. [34] study breadth-based data minimization and propose an audit method that uses feature imputation to identify whether the features used for a given model are necessary to preserve predictive performance to a pre-specified degree. Goldsteen et al. [17] also address breadth-based data minimization by identifying how to best generalize features during inference, using ideas from knowledge distillation and data anonymization. In contrast, FIDO is a depth-based data minimization framework meant to guide data collection during model training.

While it has been shown empirically that data can be minimized through various domain- and algorithm-specific heuristics [4], a question remains of how to automatically learn when to stop data collection for various personalization systems and feature acquisition strategies; the remainder of this paper focuses on this problem. For a thorough treatment of the harms data minimization protects against, we refer the interested reader to Biega and Finck [3].

3.2 Performance Curves.

Our approach is closely related to empirical research on performance curve estimation, which examines the relationship between dataset size and model performance. The literature considers many metrics, including sensitivity [19], error rates [18], accuracy [8, 27], and confidence [11, 26]. While these works typically assume a power law relationship, alternatives have been considered and shown to be comparable in accuracy [12, 27].

Tae and Whang [36] propose a data collection framework most closely related to ours. They use performance curves to identify classes which require more data to achieve equitable error rates. Tae and Whang [36] assume a power law relationship throughout the data collection process. In contrast, we model regions of the performance curve separately, and most importantly, use the performance curve to *identify a data collection stopping point.*

Literature on learning curves—the relationship between training epochs and model performance—is related to our own, but aims to learn curves that are generalizable across model configurations or hyperparameter sets. Thus, the methods assume access to hundreds of architectures [2] or multiple datasets [40] and depart from our setting. We include a nonparametric baseline inspired by this work in the supplementary material and find that the resulting performance curves are insufficient for achieving data minimization criteria.

4 PROBLEM FORMALIZATION

The key proposal in this paper is to maximally limit data collection given a target performance using *performance curves*. We plot the true performance curves for the GoogleLocal-L dataset [33] and MovieLens-20M dataset [20] and contrast these to the three phases of data collection identified by Hestness et al. [22] (Fig. 1 left, courtesy of Hestness et al. [22]). The phases are:

(1) *The small data region*, where the collected data is insufficiently representative and model performance is poor.
(2) *The power-law region*, where there is a direct trade-off between the amount of data collected and performance.
(3) *The irreducible error/diminishing returns region*, where the collection of more data does not lead to model improvement.

We can make two observations from Figure 1. The first is that the phases of data collection identified by Hestness et al. [22] in machine translation exist for recommendation datasets. This is true for both GoogleLocal and MovieLens. The second observation is that a mere 10% of the dataset lands data collection outside of the small data region. A majority of data collection occurs between the power law region and the diminishing returns region. This has important implications for modeling the performance curve: modeling the entire region using one power law curve produces underestimates of the predicted generalization error given additional data. In later sections, we show that modeling each phase separately mitigates this effect.

From a practical perspective, the data collection phases could be used to decide when collecting more data is necessary for reliable model performance (the small data region), when a user should opt to trade more data for better performance (the power-law region), and when data collection should stop (the irreducible error, or diminishing returns region).

The distinction of these phases is also pertinent from a legal perspective. In particular, the application of data minimization's necessity criterion would indicate that continued collection of personal data in the third phase would be hard to justify as it is not "necessary" to improve the model and meet its underlying purpose. We formalize these implications into a formal stopping criterion based on an empirical derivative of the the learned performance curve.

4.1 Formal Interpretation.

We follow a recent interpretation that ties data collection to *model performance metrics* [4]. However, operationalizing this interpretation remains unclear. We propose a formalization based on the returns in performance from additional data.

4.1.1 Scenario and Notation. We assume a scenario where a data processor operates a service (a model M, such as a recommender system) and collects data from a pool of queryable data $\mathcal{P}$ (consisting of user-feature-value triples) generated by a population of

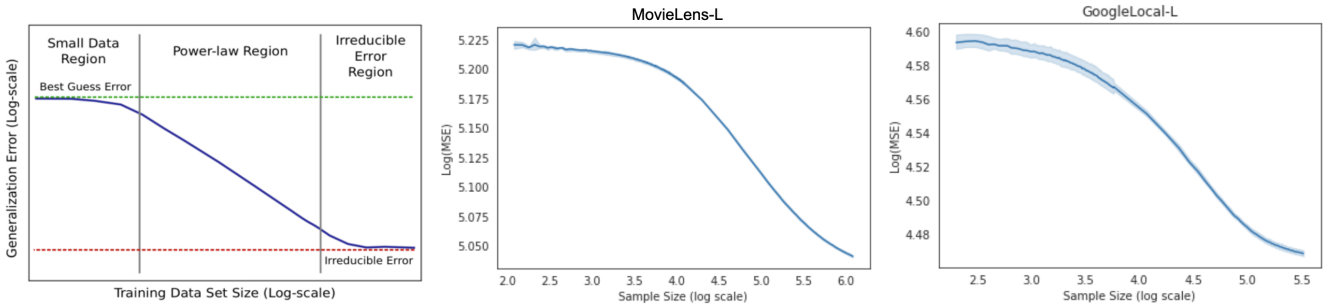

Figure 1: Model performance over the course of data collection. On the left, a figure from Hestness et al. [22] plots the phases of data collection. On the right, we plot model performance over data collection from GoogleLocal and MovieLens-20M. Note the change in slope over the course of data collection; we see for both GoogleLocal and MovieLens-20M, the return on additional data decreases, in line with current understanding of the diminishing returns region. Preprocessing details for each dataset can be found in the supplementary material.

users $\mathcal{U}$. We define M as the collection of parameters learned using personal data, including hyperparameters.

The acquired data is used to train M and make predictions for each user $u \in \mathcal{U}$. We further assume that, when data collection begins, the data processor has access to some initial data: $\mathcal{I}$ for training the model and $\mathcal{V}$ set aside to validate model performance predictions. Such initial data would include any data that is historical, purchased, or collected in different markets.

During data collection, the processor applies a feature acquisition policy $H(\mathcal{P}, n)$ which queries n feature values from $\mathcal{P}$. Queries equate to the collection of a specific user-feature-value for inclusion in the training set for model M. We refer to the union of initial and acquired data as $\mathcal{A}$ and let $|\cdot|$ denote the cardinality of a set. Let σ_M represent the true performance curve for M, which maps a domain of training dataset sizes to a range of model performances as measured by a performance metric σ. User-specific performance curves for M are termed σ_M^u for a given user u. The objective used to train M translates to the processing purpose.

The processor can use the resulting predicted performance curve to adhere to a concrete data minimization objective. In the main text, we propose and validate one such objective, detailed below: *returns-based* data minimization. Experiments in the supplementary material furthermore engage an alternate formalization of data minimization, where the stopping criterion is determined by the *relative model performance* achieved rather than the performance returns, providing further evidence of FIDO's flexibility.

4.1.2 Minimizing by Returns in Performance. We minimize in reference to a threshold on the *return* in model performance of additional data. One could select an appropriate threshold by assessing user preferences or selecting a sufficiently small threshold such that user experience is not affected. Formally, a data collector would cease data collection once the slope of the performance curve drops below threshold t:

$$\frac{d\sigma_M}{dn}(|\mathcal{A}|) \leq t \tag{1}$$

Producing an accurate approximation of σ_M is thus central to any performance-based data minimization objective. Our experiments show that existing approaches produce performance curves that are insufficient for the stated objectives. We compensate for these shortcomings by providing an accurate parametric model to approximate σ_M.

5 FIDO: FRAMEWORK FOR INHIBITING DATA OVERCOLLECTION

The framework accepts three parameters: feature acquisition algorithm H, model M, and performance metric σ. There are three steps: (1) H acquires a portion of the available data (*Data Collection*), (2) the performance curve is fit to the new data (*Curve Fitting*), and (3) Steps 1 and 2 repeat until the conditions of Step 3 (*Stopping Criterion Evaluation*) are met.

5.1 Step 1: Collect Data.

In this step, a feature acquisition algorithm H collects q observations from the pool of available observations $\mathcal{P}$. Smaller q translate to more conservative data collection processes and to more accurate estimates of the stopping criterion at the expense of decreased efficiency (smaller q mean that Steps 2 and 3 are executed more frequently). One might choose to set a larger q early on during data processing, and decrease it as the data processing continues.

5.2 Step 2: Fit the Performance Curve.

The key idea underpinning this step is to model the phases of data collection *separately* via a piecewise power law curve. The piecewise power law curve is the piece-wise combination of three power law curves, which we will refer to as f_0, f_1, and f_2:

$$f(x) = \begin{cases} f_0(x) = a_0 x^{-b_0} & 0 \leq x \leq t_0 \\ f_1(x) = a_1 x^{-b_1} & t_0 < x \leq t_1 \\ f_2(x) = a_2 x^{-b_2} & t_1 < x \end{cases} \tag{2}$$

where $f(x)$ accepts as input a training set size x. We fit the piecewise power law curve to subsamples of $\mathcal{A}$ of different sizes. More specifically, given the query size parameter q, we generate $|\mathcal{A}|/q$ samples such that the size of each consecutive sample increases

by q. We train the model on each sample and evaluate model performance on $\mathcal{V}$. The resulting pairs of values (sample size and performance on the validation set) are then used to fit $f(x)$. We fit the parameters for f_0, f_1, and f_2 using weighted non-linear least squares; details can be found in the supplementary material.

Finally, we optimize thresholds t_0 and t_1 using coordinate descent, such that the *slopes* of each consecutive pair of power laws maximally differ. This translates to an iterative approach that first estimates t_0 while t_1 is left fixed, and then estimates t_1 while t_0 is fixed, until convergence. More formally, we alternate optimization between the following objectives:

$$\max_{t_0} \quad |b_0 - b_1| \qquad s.t. \qquad 0 < t_0 < t_1 \tag{3}$$

$$\max_{t_1} \quad |b_1 - b_2| \qquad s.t. \qquad t_0 < t_1 < |\mathcal{A}| \tag{4}$$

While t_0 and t_1 do not appear directly in the optimization objective, they impact the estimates of b_0, b_1, and b_2 by delimiting which sample size and model performance pairs are used to estimate each decay parameter (as described in Equation (2)). Assuming the underlying function is a piecewise power law curve, Equations (3) and (4) are convex optimizations and converge to the correct thresholds.

This follows our intuition regarding the each region: namely, that the phases are distinguished by the differing *return* in additional data. Note that in this paper we assume that model performance increases as we collect more data. This is a common assumption in the performance curve literature [22, 27], but there are cases in which additional data may hurt model performance. We discuss one such case in Section 6.3.2. In these settings, a different family of parametric curves should be used and FIDO remains flexible to alternate performance curve estimation procedures.

In theory, one could compute a stopping criterion directly from the subsamples of $\mathcal{A}$, without fitting the performance curve. This has two undesirable consequences. The first is the instability of the resulting stopping criteria, due to the noise inherent to performance measurements from individual subsamples, as we will see in later experiments. Second, learning a performance curve allows the data minimizer to reason about performance given additional data, *without collecting that data.* This affords the framework flexibility to a range of data minimization objectives, including those that cease data collection based on absolute model performance rather than performance increase rate (relevant experiments are in the supplement).

5.3 Step 3: Evaluate Stopping Criterion.

In this step, the resulting performance curve is used to accomplish a specific data minimization objective. Note that these are not the only reasonable data minimization objectives and the framework can adapt to different formulations.

Minimizing by returns requires the data collector to specify a threshold for return after which data collection should stop, $t \in \mathbb{R}$. We can use the performance curves to estimate this quantity by taking the derivative at a given sample size:

$$\hat{s} = \begin{cases} -b_0 a_0 x^{-b_0-1} & 0 \le x \le t_0 \\ -b_1 a_1 x^{-b_1-1} & t_0 < x \le t_1 \\ -b_2 a_1 x^{-b_2-1} & t_1 < x \end{cases} \tag{5}$$

Dataset	# Users	# Items	Item type	Sparsity
MovieLens-L	5000	17400	movie	1.7%
MovieLens-S	1000	11529	movie	2.6%
GoogleLocal-L	1500	265807	business	0.1%
GoogleLocal-S	500	104766	business	0.3%

Table 1: Dataset statistics.

Once $\hat{s}$ falls below t, data collection stops. Implementation details are in the supplementary material.

6 EXPERIMENTS

6.0.1 Datasets. We perform experiments on two datasets in the recommender system domain: MovieLens-20M [20] and GoogleLocal [21, 33]. The datasets contain user ratings for movies and businesses, respectively. For each, the task is to predict user ratings for unseen items. We sample each dataset at two sizes to examine how results generalize across user numbers and sparsity levels. Dataset statistics can be found in Table 1 and preprocessing pipelines can be found in the supplementary material.

Each dataset is subject to the same initial, validation, and test splits, where each split is 10% of the total ratings and stratified across users. The remaining 70% of the data is the queryable rating set $\mathcal{P}$. We produce 5 random splits of each dataset according to these divisions. All results are reported over the 5 splits. We assume random feature acquisition unless otherwise stated.

6.0.2 Alternate Curve Models. Methods relating dataset size to performance commonly assume a power law model [22, 36]. We benchmark FIDO's piecewise power law technique against alternate approaches in the literature. We include *2P-PL-Initial* to determine the benefit of updating the curve as data is acquired, and a two-parameter power law method *2P-PL* to represent the most common approach to fitting performance curves [13, 36]. The remaining baselines represent variations of the power law curve that capture the notion of diminishing returns. The first (*3P-PL*) models the irreducible error directly and the second (*3P-PL-Exp*) models an additional exponential decay. The parameter fitting approach is the same for our method and described in the supplementary material.

- *2P-PL-Initial*: Fits two-parameter power law ($f(x) = ax^b$) to subsamples of $\mathcal{I}$.
- *2P-PL*: Fits two-parameter power law ($f(x) = ax^b$) to subsamples of $\mathcal{A}$.
- *3P-PL*: Fits three-parameter power law ($f(x) = ax^b + c$) to subsamples of $\mathcal{A}$.
- *3P-PL-Exp*: Fits three-parameter power law with an exponential cutoff ($f(x) = x^a e^{bx} + c$) to subsamples of $\mathcal{A}$.
- *Naive*: Estimates slope of the performance curve empirically via last two subsamples of $\mathcal{A}$.
- *Oracle*: Estimates slope via a discrete approximation using *all* sample size and performance pairs. Exact implementation is in the supplementary material.

We include comparisons to two additional baselines—a variant of the proposed curve model with two pieces (for the power law curve stage and diminishing returns stage) rather than three, and a nonparametric regression model—in the supplementary material.

6.0.3 Hyperparameters. We assume that M is a FunkSVD [15] recommendation system. We use the same hyperparameters for the number of latent features r and query size q across all experiments:

r is set to be 30, and q is set to be 2% of the number of queryable entries.

6.0.4 Metrics. We measure performance via mean-squared error (MSE), a standard recommendation evaluation metric. Methods are compared based on (1) return given additional data (change in MSE per additional feature-value observation) and (2) the amount of data collected for a given threshold t (reported over 5 dataset splits). We compute statistical significance via a paired t-test with a Bonferroni correction.

6.1 Evaluation of Data Minimization Objective.

We compare the amount of data collected by FIDO using a suite of performance curve models, including the piecewise power law technique described in Section 5.2, to the amount of data collected by an oracle with access to *all* sample size and performance pairs. For a variety of thresholds, FIDO paired with the piecewise power law technique halts data collection significantly closer to a criterion with access to all sample size and performance estimates (Table 2, p < 1e-5). Due to the noise of the performance measurements, *Naive* produces stopping criteria that are both noisier and further from the true stopping point. Later experiments show that *Naive* fails to generalize to different feature acquisition algorithms and cannot accommodate alternate performance-based data minimization objectives.

The slope estimates drawn from the performance curve in FIDO are more accurate than those of the curve-fitting baselines across the stages of data collection (Figure 2 (A)). *3P-PL-Exp* consistently underestimates the return on additional data over the course of data collection, and subsequently halts data collection too early. In contrast, *2P-PL-Initial*, *2P-PL*, and *3P-PL* overestimate the return on additional data and frequently halt data collection later than the empirically derived stopping point. These results suggest that the use of *3P-PL-Exp* would produce a conservative data minimization approach in that such a data minimization method would be unlikely to over-collect data. The reverse is true for the remaining baselines – such a data minimization approach would likely collect more data than is required.

This is a direct result of each curve model's ability to model the last stage of data collection accurately. Examine the accuracy of each curve model's estimate of performance given the entire dataset (Figure 2 (B)). Each method converges to the true value for model performance (red) over the course of data collection. The power law baselines (*2P-PL-Initial*, *2P-PL*, *3P-PL*) underestimate error given $\mathcal{P}$. This confirms results from prior work in machine translation [27], and is a consequence of extrapolation from the power law region into the diminishing returns region. *3P-PL-Exp* instead overestimates test error because the e^{-bx} term produces a curve too flat to describe the true relationship; illustrative plots for the performance curve fits are included in the supplementary material.

The halting points for GoogleLocal-S and MovieLens-S exhibit more noise than their larger counterparts. This suggests that producing reliable estimates of the return on additional data is more challenging for smaller datasets.

6.2 Robustness to Query Size.

In the previous experiments we used a query size of 2%. Here, we investigate the robustness of FIDO's predictions to different query size values $q \in [0.005, 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07]$. Large q simulate a setting in which large batches of data are acquired at one time, while smaller q simulate the continuous arrival of new user data.

Table 3 reports that FIDO provides the closest estimation of the true model performance across different query sizes. While FIDO produces the most faithful estimates of the true stopping point across query sizes *it is more sensitive to small query sizes* compared to baselines using a single power law curve. Lower query sizes require models to be retrained more frequently to produce data to fit the performance curve. Thus, for models with computationally intensive training procedures, it is optimal to choose the largest query size that maintains accuracy. On the other hand, smaller queries will lead to more accurate stopping decisions and less data overcollection. It is up to a practitioner to select the query size based on their domain knowledge, and our results suggest that the set of query sizes producing accurate estimates is quite large.

6.3 Robustness to Feature Acquisition Algorithms.

Thus far, we have considered data collection where observations are queried randomly from $\boldsymbol{Q}$. AFA methods improve upon this approach by instead querying feature values based on their uncertainty [6, 14, 23] or contribution to a downstream task [31, 38]. Successful AFA methods collect less data than random feature acquisition and deliver equivalent performance. Recent work has shown that this success is often dependent on initialization conditions [32].

We consider two popular AFA methods: *Stability* and *Query-by-Committee* (*QBC*). *QBC* [6] employs three matrix imputation approaches (k-NN, EM, and SVD) to predict missing feature values. *Stability* takes a similar approach and predicts missing feature-values using SVD given different ranks. Each feature-value's uncertainty corresponds to the variance in predicted values. Both algorithms request the highest variance feature-values. For *Stability*, we follow the approach of [23] and set the ranks to be $[1, 2, 3, 4, 5]$.

6.3.1 FIDO is robust to different AFA algorithms. Across both AFA algorithms and multiple thresholds, FIDO halts data collection closest to the true stopping point (p <1e-3). Table 4 reports these results for GoogleLocal-L. Expectedly, we see that for the same stopping criterion (e.g., a threshold of -5.0e-7), AFA algorithms collect less data than random feature acquisition algorithms. Existing curve-fitting approaches are more competitive in the AFA setting, which can be attributed to an expanded power law region (illustrative performance curves can be found in the supplementary material.

In general, trends for each baseline hold in this setting: the curve models which use a single power law (*2P-PL-Initial*, *2P-PL*, *3P-PL*) stop collecting data too late on average, while *3P-PL-Exp* stops collecting data too early. *Naive* is naturally affected by the noisier performance curves and performs significantly worse than the other methods.

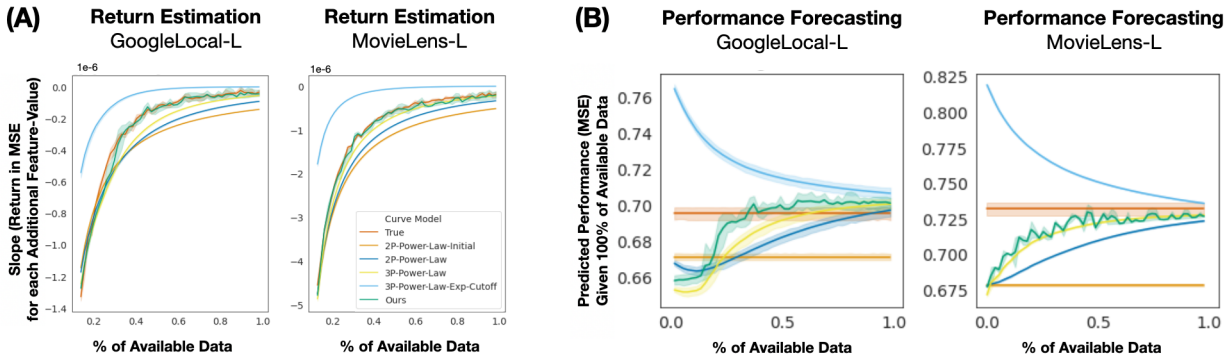

Figure 2: Evaluation of performance curves over the course of data collection. Our method outperforms baselines in estimating *return* on additional data, where return is defined as the reduction in model error given an additional feature-value observation (A). Supporting plots for GoogleLocal-S and MovieLens-S are in the supplementary material. For each dataset, we plot predicted performance, $\sigma_M(|\mathcal{I} \cup \mathcal{P}|)$, over the course of data collection using different curve-fitting methods (B). Our method (dark green) matches the true performance (red) most closely at all stages.

Dataset	Threshold	2P-PL-Initial	2P-PL	3P-PL	3P-PL-Exp	Naive	FIDO	Oracle
GoogleLocal-L	-5.0e-07	0.32 ± 0.00	0.32 ± 0.01	0.32 ± 0.01	0.16 ± 0.00	**0.27 ± 0.02**	**0.29 ± 0.02**	0.27 ± 0.01
GoogleLocal-L	-2.0e-07	0.73 ± 0.01	0.61 ± 0.01	0.53 ± 0.01	0.25 ± 0.01	0.36 ± 0.04	**0.42 ± 0.03**	0.41 ± 0.02
GoogleLocal-L	-5.0e-08	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	0.40 ± 0.02	0.52 ± 0.08	**0.68 ± 0.05**	0.68 ± 0.05
GoogleLocal-S	-5.0e-07	0.88 ± 0.01	0.71 ± 0.01	0.60 ± 0.01	0.28 ± 0.01	0.37 ± 0.07	**0.42 ± 0.06**	0.47 ± 0.03
GoogleLocal-S	-2.0e-07	1.00 ± 0.00	1.00 ± 0.00	0.91 ± 0.03	0.39 ± 0.01	0.44 ± 0.08	**0.51 ± 0.14**	0.58 ± 0.06
GoogleLocal-S	-5.0e-08	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	**0.55 ± 0.01**	0.46 ± 0.09	**0.59 ± 0.08**	0.64 ± 0.10
MovieLens-L	-5.0e-07	0.13 ± 0.00	0.13 ± 0.00	0.13 ± 0.00	0.13 ± 0.00	**0.14 ± 0.01**	**0.13 ± 0.00**	0.13 ± 0.00
MovieLens-L	-2.0e-07	0.29 ± 0.00	0.27 ± 0.00	0.26 ± 0.00	0.13 ± 0.00	**0.25 ± 0.01**	**0.25 ± 0.02**	0.23 ± 0.01
MovieLens-L	-5.0e-08	1.00 ± 0.00	0.76 ± 0.01	0.62 ± 0.01	0.24 ± 0.01	0.43 ± 0.08	**0.53 ± 0.05**	0.53 ± 0.02
MovieLens-S	-5.0e-07	0.50 ± 0.02	0.53 ± 0.01	0.51 ± 0.01	0.23 ± 0.01	**0.37 ± 0.04**	**0.46 ± 0.05**	0.45 ± 0.03
MovieLens-S	-2.0e-07	1.00 ± 0.00	1.00 ± 0.00	0.97 ± 0.02	0.36 ± 0.01	0.51 ± 0.08	**0.68 ± 0.14**	0.79 ± 0.08
MovieLens-S	-5.0e-08	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	0.64 ± 0.02	0.66 ± 0.22	**0.99 ± 0.03**	1.00 ± 0.00

Table 2: Performance over diminishing returns criterion. Each value is the fraction of data collected using a given method while adhering to the diminishing returns stopping criterion, averaged over 5 runs. Our method halts collection closest to the true stopping point across thresholds (p-value 3e-6). Bolded entries are closest to the true stopping point, within a standard deviation.

6.3.2 AFA algorithm performance depends on initial system data. In Figure 3(A) we examine the dependence of AFA algorithm performance on (1) the type of initialized data and (2) the feature acquisition algorithm employed. We consider two additional types of initialization; user-subset (initialized randomly across a subset of users) and item-subset (initialized randomly across subset of items). In each of these cases, the test set is formed from a random sample that includes ratings from all users. In agreement with Munjal et al. [32], we observe that the performance depends on data initialization conditions. When the initialization data is a random sample across users and items, AFA algorithms perform similarly. However, when the initialization data contains only a subset of users, or only a subset of items, AFA begins to decrease in performance. This is consequential in cases where (i) the population of data subjects is evolving (initialization data does not contain users who join at a later time), and (ii) the data processor is not initially allowed to collect certain feature values because of external constraints (e.g., feature sensitivity).

7 USER-SPECIFIC IMPACT ANALYSES

Previous sections discuss minimized data collection in terms of diminishing return across all users. In this section, we examine FIDO's effect on per-user metrics. We discuss how user performance-based data minimization departs from traditional assumptions for performance curves and recommend areas for further research.

7.1 Evaluation of User-Specific Data Minimization.

We analyze the performance of the framework on user-specific performance metrics in two cases. In the first, we replicate the

Query Size	Threshold	2P-PL-Initial	2P-PL	3P-PL	3P-PL-Exp	Naive	FIDO	Oracle
0.005	-2.0e-07	0.70 ± 0.01	0.62 ± 0.01	0.55 ± 0.01	0.21 ± 0.00	0.24 ± 0.04	0.30 ± 0.04	0.35 ± 0.01
0.010	-2.0e-07	0.71 ± 0.01	0.62 ± 0.01	0.55 ± 0.01	0.22 ± 0.00	0.32 ± 0.05	0.36 ± 0.02	0.39 ± 0.03
0.020	-2.0e-07	0.73 ± 0.01	0.62 ± 0.01	0.53 ± 0.01	0.25 ± 0.00	0.39 ± 0.03	0.41 ± 0.03	0.42 ± 0.02
0.030	-2.0e-07	0.75 ± 0.02	0.63 ± 0.02	0.53 ± 0.02	0.27 ± 0.01	0.43 ± 0.02	0.46 ± 0.02	0.43 ± 0.01
0.040	-2.0e-07	0.79 ± 0.03	0.64 ± 0.02	0.53 ± 0.02	0.29 ± 0.00	0.44 ± 0.02	0.46 ± 0.02	0.44 ± 0.02
0.050	-2.0e-07	0.81 ± 0.04	0.65 ± 0.02	0.54 ± 0.02	0.32 ± 0.00	0.45 ± 0.05	0.49 ± 0.02	0.46 ± 0.02
0.060	-2.0e-07	0.78 ± 0.02	0.63 ± 0.00	0.52 ± 0.02	0.32 ± 0.00	0.45 ± 0.03	0.47 ± 0.00	0.45 ± 0.03
0.070	-2.0e-07	0.84 ± 0.03	0.64 ± 0.03	0.50 ± 0.00	0.35 ± 0.03	0.49 ± 0.03	0.50 ± 0.00	0.47 ± 0.03

Table 3: Robustness to Query Size. Applied to GoogleLocal-L, each method's robustness to different query sizes sheds light on the tradeoffs involved in query size selection. The proposed method and *Naive* are competitive in their accuracy in estimating the true stopping point, while *2P-PL* and *3P-PL* produce the most consistent stopping criteria over threshold sizes.

AFA Alg	Threshold	2P-PL-Initial	2P-PL	3P-PL	3P-PL-Exp	Naive	FIDO	Oracle
Stability	-5.0e-07	0.30 ± 0.00	0.31 ± 0.03	0.31 ± 0.06	0.16 ± 0.00	0.24 ± 0.03	0.26 ± 0.06	0.22 ± 0.06
Stability	-2.0e-07	0.71 ± 0.02	0.64 ± 0.06	0.57 ± 0.11	0.24 ± 0.03	0.27 ± 0.09	0.47 ± 0.11	0.42 ± 0.09
Stability	-1.0e-07	1.00 ± 0.00	1.00 ± 0.00	0.96 ± 0.06	0.29 ± 0.06	0.29 ± 0.14	0.70 ± 0.21	0.75 ± 0.15
QBC	-5.0e-07	0.32 ± 0.02	0.33 ± 0.00	0.33 ± 0.00	0.19 ± 0.02	0.23 ± 0.00	0.24 ± 0.02	0.21 ± 0.02
QBC	-2.0e-07	0.73 ± 0.02	0.67 ± 0.02	0.58 ± 0.02	0.24 ± 0.02	0.23 ± 0.00	0.38 ± 0.03	0.36 ± 0.03
QBC	-1.0e-07	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	0.31 ± 0.02	0.23 ± 0.00	0.88 ± 0.27	0.96 ± 0.00

Table 4: Robustness to AFA Algorithm. The proposed method adheres most closely to the true stopping point across AFA algorithms applied to data collection from GoogleLocal-L.

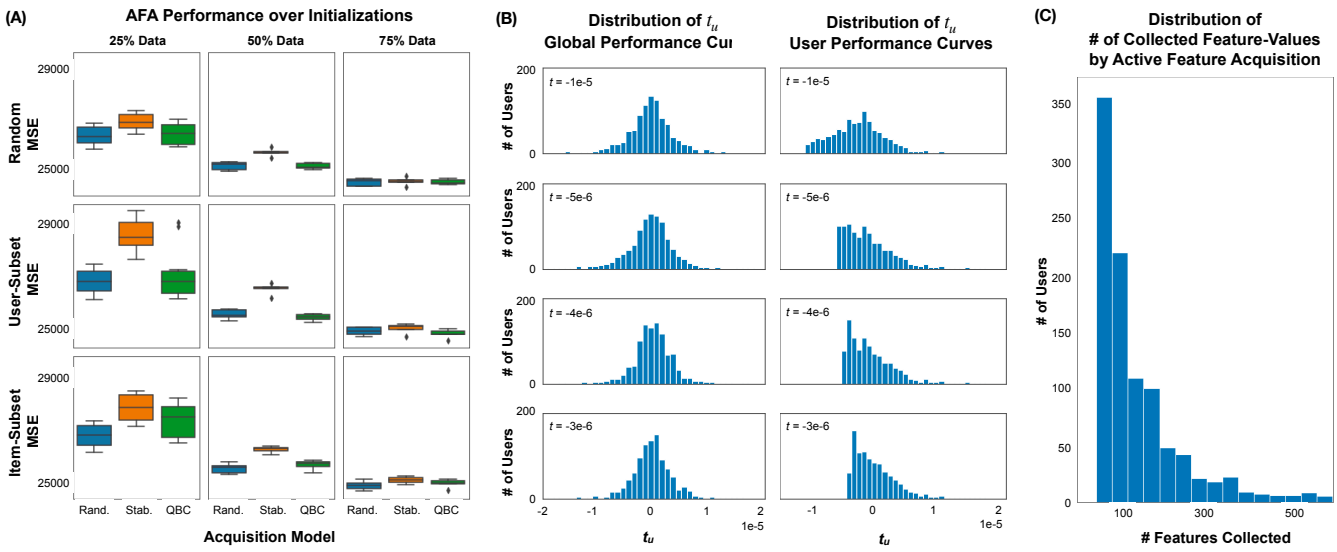

Figure 3: (A) We show that the performance achieved during data collection depends on both the AFA algorithm employed and the initialization conditions. Error bars are reported over 5 random initializations. (B) User-specific performance metrics when minimized data collection learns one curve for all users (left) and a curve for each user (right). Unsurprisingly, we see that the mode of user-specific performance increases as t increases. This is also the case when learning user-specific performance curves, but there is a large spread of true returns on additional data. This suggests future areas of work for learning accurate user-specific performance curves. (C) We also show that a small portion of users bear the majority of the data collection burden in a histogram of the quantity of features acquired per user by *Stability* from MovieLens-S halfway through data collection.

setting discussed in previous sections and learn a performance curve for the entire dataset. In the second, FIDO learns a performance curve *per user* and applies a stopping criterion based on goal t to each curve. We use the same procedure to estimate t_u as t earlier, except we use a user-specific performance curve rather than a global performance curve. The user-specific performance curve is learned by replacing a global performance metric (MSE over all users) with a user-specific performance metric (MSE over a set of items specific to one user). Estimating t_u from σ_M^u produces a t_u for each user.

In Figure 3(B), we plot the distribution of user returns in performance given a global threshold t (left). As t increases, the distribution mode shifts up and the variance in user fractions of performance decreases. Note that the x-axis for each histogram extends beyond 1. This is because more data does not necessarily translate to increased *per-user* performance. Two factors are responsible: 1) The small validation set size for each user produces noisy performance estimates and 2) the collection of additional data can still hurt user performance if this data is not representative. The assumption of monotonically increasing performance over data collection does not hold, and accordingly, the framework does not perform as well. A key takeaway from this experiment is that the return in performance may not be an appropriate metric for data minimization on a per user level.

7.2 Lessons on User-Specific Data Minimization.

Methods designed to address user-specific data minimization should consider:

7.2.1 Tensions between AFA and fairness. While AFA is a natural choice for limited data collection, we find that it introduces disparate data collection burden across users by collecting a different number of features from different users. Figure 3(C), plots a histogram of the quantity of collected data over users for AFA algorithm *Stability*, for dataset MovieLens-S (similar trends exist for other datasets). AFA algorithms "exploit" a small number of users by collecting a large number of feature-values from them. Yet, our experiments also show that increased data collection significantly correlates with better performance for individual users. It is likely that certain users would choose to bear the burden of excess data collection in exchange for better performance. Thus, data minimization given an AFA approach raises questions of both user fairness and user agency.

7.2.2 More complex curve models. It is worth exploring a family of parametric curves that describe the phenomenon of a user whose model performance degrades with the collection of additional data. Approximately 20% of users in each of the four datasets exhibit this property, suggesting that monotonic curve models are not the right choice for modeling user-specific performance curves. Moreover, user-specific performance curves are noisier and as a result, may benefit from drawing multiple samples at each subsample size.

8 PUTTING IT ALL TOGETHER

Thus far, we have put forth a framework to guide data collection by learning the an algorithm's performance curve as data arrives. Subsequent experiments validated the resulting performance curve on a returns-based data minimization objective. These experiments deliver a number of takeaways for the data minimization practitioner. Here, we summarize the implications of our results on specific design choices:

(1) **Choosing a Feature Acquisition Algorithm.** Based on our experiments, we recommend random feature acquisition to guide data minimization. We base this recommendation on three reasons: 1) random feature acquisition produces a smoother performance curve, and thus, more accurate performance estimates, 2) our work confirms recent findings that demonstrate how the success of intelligent data collection depends upon initialization conditions [32], and 3) AFA algorithms can place excess data collection burden on specific users, as our user-specific impact analyses show.
(2) **Choosing a Performance Curve Model.** The piecewise power law curve model is best fit to describe the relationship between dataset size and performance in the datasets we consider. We see that this choice may not be consequential when the diminishing returns region is small, as we see with MovieLens-S (Sec. **??**).
(3) **Creating a Representative Validation Set.** This framework hinges on the creation of a representative validation set. As is the case with many systems that operate on continuously collected data, a representative validation set may need to be updated to account for data drift over time.
(4) **Identifying a Relevant Objective.** While the framework we propose is flexible to many objectives, it is worth considering when one objective may be more desirable than another. We provide a case study of an alternate objective, based on relative performance, in Section **??**. One can consider its *ease of definition*: how well does the selected objective reflect user preferences? Gauging user preferences in terms of relative performance may be easier to survey for than diminishing returns. One could also consider the objective's *specificity*. Our experiments show that minimizing by diminishing returns is more accurate earlier on during data collection compared to minimization by relative performance. Dataset size plays a role too, as our experiments show that both the prediction of returns and relative performance are noisier in smaller datasets.
(5) **Performing User-Specific Impact Analyses.** Data minimization may disproportionately affect marginalized populations. Recent work [7, 36] has shown that in some instances, it is necessary to collect *more* data to ensure the equitable performance. As a result, it is imperative to perform per-user analyses when studying and proposing methods for data minimization.

9 LIMITATIONS

FIDO is not without limitations. First, the framework relies on access to a validation set large enough to approximate performance on the test set. Experiments on MovieLens-S and GoogleLocal-S show that smaller validation set sizes translate to higher variance in data minimization performance. One could mitigate this variation in performance by intelligently constructing the validation set, rather than randomly. Such an approach could be useful in settings where

concept drift is common, and the validation set must be updated to remain representative of the test set.

Moreover, this work considers data acquisition where feature-values are roughly homogeneous in terms of sensitivity, since each is an item rating. Data minimization concerns not only whether data is collected, but also what *type* of data is collected. This includes the identifiability or pseudonymity of the collected data. Intersections between FIDO and recent work on breadth-based data minimization [17, 34] can address feature-specific data minimization concerns.

Finally, FIDO is only one way to implement data minimization and represents one facet of a comprehensive approach to data minimization. In practice, a data processor would deploy multiple techniques that work in concert to minimize data across stages of a system's life cycle, including data collection, storage, and inference.

10 DISCUSSION

This work aims to bridge the gap between the legal principle of data minimization and its practical realization. Data-driven systems often operate under the assumption that more data is unequivocally better, for all parties involved. This might not the case when the identifiability, sensitivity, liability, and storage costs of collected data are acknowledged and appropriately balanced. We intend our work to be a step towards respectful data collection.

Towards this end, we propose FIDO, a Framework for Inhibiting Data Overcollection. FIDO takes an idea established across domains in machine learning—the ubiquity of scaling laws in data driven systems—and uses it to provide a performance-based stopping criterion. FIDO accomplishes this by acquiring data in small batches, refitting an estimate of the performance curve, and applying a performance-based stopping criterion. FIDO uses a piecewise power law technique grounded in different phases of data collection to produce an accurate estimate of the performance curve.

Our empirical investigation of FIDO revealed findings with practical implications for the implementation of data minimization. Specifically, certain performance curve families (e.g., the three-parameter power law) systematically overcollect data by not modeling each data collection phase separately. We also demonstrate how active feature acquisition—a technique which might be thought of as a go-to tool for data minimization—can be undesirable in the context of personal data protection. We found that AFA can place excess data collection burden on a small set of users, and that the technique's performance depends on data initialization conditions, with degrading performance in simulations of evolving user communities or restricted feature sets

In light of these complexities, we believe that data minimization compliance in machine learning models will require similar efforts as the principle of fairness has garnered. Definitions, formal implementations and caveats will depend on the application domain, the underlying model (e.g., recommendation, classification). The piecewise power law may not be appropriate for all domains, including user-specific data minimization. Moreover, adversarial approaches are possible: a data processor acting in bad faith may choose a model class that requires a large amount of personal data. Further research is necessary to ensure data minimization occurs despite malicious data collection practices.

ACKNOWLEDGMENTS

We thank John Guttag, Hansa Srinivasan, and Hal Daumè III for helpful comments.

REFERENCES

[1] Datatilsynet: The Norwegian Data Protection Authority. [n.d.]. Artificial Intelligence and Privacy.
[2] Bowen Baker, Otkrist Gupta, Ramesh Raskar, and Nikhil Naik. 2017. Accelerating neural architecture search using performance prediction. *arXiv preprint arXiv:1705.10823* (2017).
[3] Asia J. Biega and Michèle Finck. 2021. Reviving Purpose Limitation and Data Minimisation in Data-Driven Systems. *Technology and Regulation* (2021).
[4] Asia J. Biega, Peter Potash, Hal Daumé, Fernando Diaz, and Michèle Finck. 2020. Operationalizing the Legal Principle of Data Minimization for Personalization. In *ACM(43) SIGIR '20*. 399–408.
[5] Reuben Binns and Valeria Gallo. 2019. *Data minimisation and privacy-preserving techniques in AI systems*. https://ico.org.uk/about-the-ico/news-and-events/ai-blog-data-minimisation-and-privacy-preserving-techniques-in-ai-systems/
[6] Shayok Chakraborty, Jiayu Zhou, Vineeth Balasubramanian, Sethuraman Panchanathan, Ian Davidson, and Jieping Ye. 2013. Active Matrix Completion. *ICDM*.
[7] Irene Chen, Fredrik D Johansson, and David Sontag. 2018. Why is my classifier discriminatory?. In *NeurIPS*. 3539–3550.
[8] Junghwan Cho, Kyewook Lee, Ellie Shin, Garry Choy, and Synho Do. 2015. Medical Image Deep Learning with Hospital PACS Dataset. *CoRR* (2015).
[9] Richard Chow, Hongxia Jin, Bart Knijnenburg, and Gokay Saldamli. 2013. Differential data analysis for recommender systems. In *ACM7-RecSys*. 323–326.
[10] Norwegian Data Protection Authority Datatilsynet. 2018. *Artificial Intelligence and Privacy*. https://www.datatilsynet.no/en/regulations-and-tools/reports-on-specific-subjects/ai-and-privacy/
[11] Kevin K. Dobbin, Yingdong Zhao, and Richard M. Simon. 2008. How Large a Training Set is Needed to Develop a Classifier for Microarray Data? *Clinical Cancer Research* 14, 1 (2008), 108–114.
[12] Tobias Domhan, Jost Tobias Springenberg, and Frank Hutter. 2015. Speeding up automatic hyperparameter optimization of deep neural networks by extrapolation of learning curves. In *IJCAI-24*.
[13] Rosa L Figueroa, Qing Zeng-Treitler, Sasikiran Kandula, and Long H Ngo. 2012. Predicting sample size required for classification performance. *BMC Medical Informatics and Decision Making* 12, 1 (2012).
[14] Yoav Freund, H Sebastian Seung, Eli Shamir, and Naftali Tishby. 1997. Selective sampling using the query by committee algorithm. *Machine learning* 28 (1997).
[15] Simon Funk. 2006. Netflix update: Try this at home. *https://sifter.org/ simon/journal/20061211.html* (2006).
[16] GDPR. 2016. Regulation (EU) 2016/679 of the European Parliament and of the Council. *Official Journal of the European Union* (2016).
[17] Abigail Goldsteen, Gilad Ezov, Ron Shmelkin, Micha Moffie, and Ariel Farkash. 2021. Data minimization for GDPR Compliance in machine learning models. *AI and Ethics* (2021), 1–15.
[18] Seda Gürses, Carmela Troncoso, and Claudia Diaz. 2015. Engineering privacy by design reloaded. In *Amsterdam Privacy Conference*. 1–21.
[19] Karimollah Hajian-Tilaki. 2014. Sample size estimation in diagnostic test studies of biomedical informatics. *Journal of Biomedical Informatics* 48 (2014), 193 – 204.
[20] F Maxwell Harper and Joseph A Konstan. 2015. The movielens datasets: History and context. *Acm-TiiS* 5, 4 (2015), 1–19.
[21] Ruining He, Wang-Cheng Kang, and Julian McAuley. 2017. Translation-based recommendation. In *ACM11-RecSys*. 161–169.
[22] J Hestness, S Narang, N Ardalani, G Diamos, H Jun, H Kianinejad, M Patwary, Y Yang, and Y Zhou. 2017. Deep Learning Scaling is Predictable, Empirically. *arXiv preprint arXiv:1712.00409* (2017).
[23] Sheng-Jun Huang, Miao Xu, Ming-Kun Xie, Masashi Sugiyama, Gang Niu, and Songcan Chen. 2018. Active feature acquisition with supervised matrix completion. In *ACM24-SIGKDD*. 1571–1579.
[24] UK Information Commisioner's Office: ICO. 2018. *Guide to Data Protection. Some basic concepts.* Retrieved Jan 22, 2020 from https://ico.org.uk/for-organisations/guide-to-data-protection/introduction-to-data-protection/some-basic-concepts/
[25] UK Information Commisioner's Office: ICO. 2018. *Guide to the General Data Protection Regulation (GDPR). Principle (c): Data minimisation.* Retrieved Jan 22, 2020 from https://ico.org.uk/for-organisations/guide-to-data-protection/guide-to-the-general-data-protection-regulation-gdpr/principles/data-minimisation/
[26] HM Kalayeh and David A Landgrebe. 1983. Predicting the required number of training samples. *IEEE-TPAMI* 6 (1983), 664–667.
[27] Prasanth Kolachina, Nicola Cancedda, Marc Dymetman, and Sriram Venkatapathy. 2012. Prediction of learning curves in machine translation. In *ACL '12*.
[28] Bert-Jaap Koops. [n.d.]. The trouble with European data protection law'(2014). *International Data Privacy Law* 4 ([n. d.]), 250.

[29] Andreas Krause and Eric Horvitz. 2010. A utility-theoretic approach to privacy in online services. *Journal of Artificial Intelligence Research* 39 (2010), 633–662.
[30] Mark MacCarthy. 2018. In Defense of Big Data Analytics. *The Cambridge Handbook of Consumer Privacy* (2018), 47–78.
[31] Prem Melville, Maytal Saar-Tsechansky, Foster Provost, and Raymond Mooney. 2005. An expected utility approach to active feature-value acquisition. In *ICDM*.
[32] Prateek Munjal, Nasir Hayat, Munawar Hayat, Jamshid Sourati, and Shadab Khan. 2020. Towards Robust and Reproducible Active Learning Using Neural Networks. *arXiv* (2020), arXiv–2002.
[33] Rajiv Pasricha and Julian McAuley. 2018. Translation-based factorization machines for sequential recommendation. In *ACM-12-RecSys*. 63–71.
[34] Bashir Rastegarpanah, Krishna Gummadi, and Mark Crovella. 2021. Auditing Black-Box Prediction Models for Data Minimization Compliance. *Advances in Neural Information Processing Systems* 34 (2021).
[35] Awanthika Senarath and Nalin Asanka Gamagedara Arachchilage. 2018. Understanding Software Developers' Approach towards Implementing Data Minimization. *arXiv preprint arXiv:1808.01479* (2018).
[36] Ki Hyun Tae and Steven Euijong Whang. 2020. Slice Tuner: A Selective Data Collection Framework for Accurate and Fair Machine Learning Models. (2020).
[37] Nicholas Vincent, Brent Hecht, and Shilad Sen. 2019. "Data Strikes": Evaluating the Effectiveness of a New Form of Collective Action Against Technology Companies. In *The World Wide Web Conference*. ACM.
[38] Duy Vu, Mikhail Bilenko, Maytal Saar-tsechansky, and Prem Melville. 2007. Intelligent Information Acquisition for Improved Clustering.
[39] Hongyi Wen, Longqi Yang, Michael Sobolev, and Deborah Estrin. 2018. Exploring recommendations under user-controlled data filtering. In *ACM12 RecSys*. 72–76.
[40] Martin Wistuba and Tejaswini Pedapati. 2020. Learning to Rank Learning Curves. In *International Conference on Machine Learning*. PMLR, 10303–10312.