Learning to Limit Data for Data Minimization

First Author Name,[1] **Second Author Name,** [2] **Third Author Name** [1]
[1] Affiliation 1
[2] Affiliation 2
firstAuthor@affiliation1.com, secondAuthor@affilation2.com, thirdAuthor@affiliation1.com

Abstract

Data minimization is a legal obligation defined in the European Union's General Data Protection Regulation (GDPR) as the responsibility to process an adequate, relevant, and limited amount of personal data in relation to a processing purpose. However, the lack of technical interpretations of the principle in the context of machine learning systems has inhibited adoption. In this paper, we follow a technical interpretation of data minimization that ties processing purpose to system performance. We propose a data collection algorithm that operationalizes this interpretation to predict a performance-based stopping criterion. The algorithm builds on prior work that relates dataset size to performance by modeling distinct stages of data collection separately. We show that this approach produces stopping criterion that are significantly more accurate than past approaches on two datasets. We conclude with practical recommendations for the implementation of data minimization.

Introduction

Article 5(1)(c) of the European Union's Data Protection Regulation (GDPR 2016) as well as data protection laws in other jurisdictions mandate a principle of *data minimization*:

> "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. Among the components of minimization, limitation requires that residual data, which is unnecessary for a declared processing purpose, is minimized out. Depending on the specific stage of the machine learning pipeline (data collection, pre-processing, training, or storage), this may mean discarding or not collecting such data in the first place. Research on privacy by design (Gürses, Troncoso, and Diaz 2015) highlights data collection as a key area to implement data minimization. Moreover, reduction of unnecesary data yields direct benefits to data processors in terms of improved computational efficiency and decreased storage costs.

Perspectives and prior work on data minimization Despite legal requirements, the principle has witnessed little adoption in the context of algorithmic profiling and decision-making systems, due to a dearth of specific computational interpretations and guidelines. Indeed, qualitative research has shown a lack of data minimization standards among practitioners (Senarath and Arachchilage 2018). Moreover, legal scholars have highlighted how the adoption of this principle risks limiting the success of data-intense systems (Zarsky 2017). A recent audit of a recommendation app highlighted the tension between the limitation and adequacy requirements—minimization of sensitive personal attributes may inhibit the detection of discriminatory effects (Galdon Clavell et al. 2020).

In contrast, recent empirical research has shown that it is possible to produce similar results from data-intense systems with significantly less data (Biega et al. 2020; Chow et al. 2013; Vincent, Hecht, and Sen 2019; Wen et al. 2018). These findings are a consequence of the diminishing returns property that data collection exhibits across applications and domains (Hestness et al. 2017; Krause and Horvitz 2010; Tae and Whang 2020). Recognizing that limiting data is possible, legal guidelines point to algorithmic techniques that could be incorporated into minimization pipelines, including feature selection (Binns and Gallo 2019) or examination of learning curves (Datatilsynet 2018).

Computational interpretations of limitation in data minimization and concrete proposals for using the aforementioned algorithmic techniques remain open questions. Several challenges contribute to this gap. Reports by the Norwegian and British data protection authorities note that a key issue is not the reduction of data quantity, but identifying which data is necessary and which is excessive in relation to a processing purpose (Datatilsynet 2018; ICO 2017). Addressing this challenge, Biega et al. (2020) have recently proposed to interpret the processing purpose in data-intense systems through model performance metrics, an interpretation termed *performance-based data minimization*. Our work follows this interpretation. We address the question of how data processors may proactively satisfy performance-based minimization requirements.

Contributions In this paper, we present a data collection algorithm that *adaptively learns a performance-based stop-*

ping criterion to satisfy minimization requirements, either globally or for each data subject. Our approach builds on a recent insight from the machine learning literature that formalizes three distinct phases in the performance curves of learning algorithms—the small data phase, the power law phase, and the diminishing returns phase (Hestness et al. 2017). We show that these phases can also be observed in the context of user data collection and that modeling these phases separately allows us to learn the performance curve effectively as data is acquired and more accurately than existing approaches. Moreover, the method easily adapts to different underlying feature acquisition algorithms.

Finally, we propose how the data collection phases might be used to guide decisions concerning when to keep collecting data and how much is enough. We conclude the paper by demonstrating impact analyses which ought to be performed when selecting underlying components for minimized data collection, such as active feature acquisition (AFA) methods. We find that AFA can lead to unequal, concentrated data collection from few users, in addition to decreased minimization performance for evolving user communities or when sensitive features are excluded from initial data collection.

Related Work

Beyond the related work on data minimization discussed in the Introduction, our technical approach is closely related to empirical research on performance curve estimation. This line of work examines the relationship between dataset size and model performance. The literature considers a large spread of metrics for model performance, including sensitivity (Hajian-Tilaki 2014), error rates (Gürses, Troncoso, and Diaz 2015), accuracy (Cho et al. 2015; Kolachina et al. 2012), and confidence (Dobbin, Zhao, and Simon 2008; Kalayeh and Landgrebe 1983). Each work assumes a power law relationship. Alternatives to the power law curve have been considered and shown to be comparable in accuracy (Domhan, Springenberg, and Hutter 2015; Kolachina et al. 2012).

The work most closely related to our own is that of Tae and Whang (2020), in which the authors build on performance curve research to present a selective data collection framework. The performance curves are used to identify classes which require more data to achieve equitable error rates. Tae and Whang (2020) assume a power law relationship throughout the data collection process and address the small-data region by using weighted non-linear least squares. In contrast, we model regions of the performance curve separately, and most importantly, use the performance curve to *identify an appropriate budget*.

Ideas broadly related to data minimization exist across fields in machine learning; we include an extended discussion about active feature acquisition (Schein et al. 2002; Vu et al. 2007; Huang et al. 2018; Sutherland, Póczos, and Schneider 2013; Mavroforakis et al.; Bhargava, Ganti, and Nowak 2017), sample complexity (Ehrenfeucht et al. 1989; Yan, Chaudhuri, and Javidi 2019; Balcan, Hanneke, and Vaughan 2010; Heckel and Ramchandran 2017), and sufficient sample sizes (Suresh and Chandrashekara 2012; Manski and Tetenov 2016) in the appendix.

Interpreting Limitation in Data Minimization

Legal Perspective The GDPR requires that no more data than necessary to achieve a declare *processing purpose* (in line with the purpose limitation principle) be processed. Data minimization is both a key principle under Article 5(1)(c) GDPR as well as a component of data protection by design and by default. It can be broken down into three distinct requirements: adequacy, relevance and necessity. The data that is processed must be relevant, requiring that only data that is pertinent to achieve the purpose is processed. Personal data must also be adequate, meaning that no irrelevant data should be processed. In some circumstances, adequacy may require that more data is processed, such as where the available data is not suitable to make inferences about an underrepresented group of users. Finally, personal data can only be processed where it is necessary to achieve the declared purpose. Thus, where the same result can be achieved without processing certain data, that data ought to be minimized out. In accordance with data minimization, where there is a choice between using ordinary personal data and sensitive personal data the former ought to be given preference, and raw data should only be processed where it is not feasible to pseudonymise that data.

Computational Interpretation We build on the approach put forward by Biega et al. (2020) in which the processing purpose is associated with *improvements in model performance metrics*. This interpretation raises an open question of what it means to minimize data in relation to such a metric-based purpose. The main conceptual proposal in this paper is to maximally limit data while reaching target *fraction of the (predicted) best possible model performance*, either globally or for each data subject. We term this interpretation *relative-performance-based data minimization*.

Scenario and notation We formalize this interpretation as follows. We assume a scenario where a data processor operates a service (a model M, such as a classifier or 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 users $\mathcal{U}$. The acquired data is used to train M and make predictions for each user $u \in \mathcal{U}$. We further assume that, when the data collection starts, 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. Performance of M is measured on a held-out test set using performance metric $\sigma(M, \mathcal{I} \cup H(\mathcal{P}, n))$. User-specific performance of M is evaluated on a held-out test set of user ratings for u, and termed $\sigma_u(M, \mathcal{I} \cup H(\mathcal{P}, n))$.

Finally, g, which is a fraction of performance gain, is a parameter dictating when the stopping criterion should be enforced. The parameter can be set globally, or individually for each user. The fraction of performance gain is a relative performance metric, scaled between the model performance

given no additional data and model performance given all available data.

Relative-performance-based data minimization We minimize in reference to a relative performance rather than an absolute performance because it is difficult to identify a good goal performance without understanding (1) model performance in the absence of additional data and (2) model performance in the presence of all available data. Under the considered scenario, the data minimization task is to collect the minimum number n of feature-value pairs from queryable set $\mathcal{P}$ such that the fraction of performance gain achieved equals at least g:

$$\frac{\sigma(M, \mathcal{I}) - \sigma(M, \mathcal{I} \cup H(\mathcal{P}, n))}{\sigma(M, \mathcal{I}) - \sigma(M, \mathcal{I} \cup \mathcal{P})} \geq g \qquad (1)$$

where $\sigma(M, \mathcal{I})$ is the starting performance, $\sigma(M, \mathcal{I} \cup H(\mathcal{P}, n))$ is the current performance, and $\sigma(M, \mathcal{I} \cup \mathcal{P})$ is target performance achievable with all the available data. Note that in this paper we assume that the model performance increases as we collect more data, and the starting and target performance correspond to the worst and best performance, accordingly. This is a common assumption in the performance curve literature (Kolachina et al. 2012; Domhan, Springenberg, and Hutter 2015), but there are cases in which additional data may hurt model performance. In these settings, a different family of curve models should be used.

Stages of Data Collection

We define a performance-based stopping criterion assuming an existence of a relationship between dataset size and performance. Crucially, this relationship is not linear. We plot the true relationship for the GoogleLocal-L recommendation dataset (Pasricha and McAuley 2018) in Figure 1 and contrast it with a figure from Hestness et al. (2017) that identifies three stages of data collection. The stages 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 the performance.
3. *The irreducible error region*, where the collection of more data does not lead to model improvements.

We make two observations from Figure 1. The first is that the stages of data collection identified by Hestness et al. (2017) in machine translation exist for recommendation datasets. This is true for both the small and large sample of GoogleLocal. We include preprocessing details for the dataset in the Experimental Set-Up section.

The second observation is that a mere 10% of each dataset lands data collection outside of the small data region for both of the datasets. 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 the power law curve produces underestimates when predicting error given additional data. In later sections, we show that modeling each stage separately mitigates this effect.

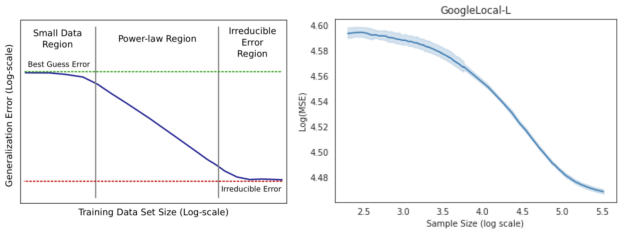

Figure 1: Model performance over the course of data collection. On the left, a figure from Hestness et al. (2017) plots the stages of data collection. On the right, we plot model performance over data collection from GoogleLocal-L. Graphs for GoogleLocal-S, MovieLens-L, and MovieLens-S are in the appendix and show the same trend.

Implications for Data Collection Practice From a *practical* perspective, the data collection stages could be used to decide when collecting more data is necessary for reliable model performance (the small data region), when a user should decide whether to trade more data for better performance (the power-law region), and when the collection can stop (the irreducible error 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 more personal data in the third stage would be hard to justify as personal data is not "necessary" to improve the model and meet its underlying purpose.

Minimized Data Collection Method

The method accepts four parameters: feature acquisition algorithm H, model M, performance metric σ and a goal fraction of performance gain $g \in [0, 1]$. There are three steps: (1) H acquires a portion of the available data (*Data Collection*), (2) the method fits a performance curve to the new data and updates the predicted fraction of performance gain (*Curve Fitting*), and (3) Steps 1 and 2 repeat until the predicted fraction of performance gain exceeds g (*Stopping Criterion Evaluation*). We outline these steps in more detail below and provide pseudo-code in the appendix.

Data collection In this step, a feature acquisition algorithm H collects q observations from the pool of available observations $\mathcal{P}$. Smaller q translates to more conservative data collection processes and to more accurate estimates of the stopping criterion at the expense of decreased efficiency (smaller q means that Curve Fitting and Stopping Criterion Evaluation are executed at smaller intervals). In practice, one might choose to set a larger q early on during data processing, and decrease it as the data processing continues.

Curve fitting The key novel idea underpinning this step is to predict the *lower bound* and *upper bound* on model performance—given a set of previously collected data $\mathcal{A}$—with *separate* power law curves, denoted f_l and f_u respectively. To simulate the conditions of varying performance, we fit those curves on samples of $\mathcal{A}$ of different sizes. More specifically, given the current data increase parameter q, we

generate $\frac{|\mathcal{A}|}{q}$ samples such that the size of each consecutive sample increases by q, where $|\mathcal{A}|$ denotes the cardinality of $\mathcal{A}$. The function to predict the lower bound learns from the half of the samples of the smallest size, while the function to predict the upper bound is informed by the half of the samples of the largest size. For each sample, we evaluate model performance on $\mathcal{V}$, and use the resulting pairs of values (sample size and performance on the validation set) to fit the performance curves.

For both f_l and f_u, we assume a power law relationship and learn the parameters A and b for the following equation:

$$f(x) = Ax^{-b} \tag{2}$$

We linearize this equation to produce:

$$\log(f(x)) = -b\log(A) - b\log(x) \tag{3}$$

Past work shows that linearizing the equation produces more stable performance curve estimates (Xiao et al. 2011). One can fit the learning curve using an ordinary least squares (OLS) method. The intuition behind learning two separate curves is straightforward: it is likely that the beginning and end of data collection lie in separate portions of the data collection curve, and the same power law curve does not adequately describe both portions. In the event they do not, there is no harm to modeling them separately.

We choose to train f_u on the largest sample sizes because the function is ultimately used to forecast model performance given additional data and this extrapolation is best informed by larger sample sizes, which are more likely to lie in the same data collection phase.

The choice to train f_l to estimate performance given $\mathcal{I}$ is more subtle. One could use the validation set performance of M given $\mathcal{I}$. However, this performance is a point estimate of the true value. By training f_l, we allow our estimate to be informed by sample sizes close to $|\mathcal{I}|$.

Stopping criterion evaluation Given the number of acquired observations, we use the learning curves to estimate the current fraction of total performance and compare it with the user-specified goal. We calculate the current fraction of performance gain as follows:

$$\hat{p} = \frac{f_l(|\mathcal{I}|) - f_u(|\mathcal{A}|)}{f_l(|\mathcal{I}|) - f_u(|\mathcal{I} \cup \mathcal{P}|)} \tag{4}$$

Data collection stops when $\hat{p}$ exceeds g. We use the magnitude of $\mathcal{P}$ in our experiments, but in practice, data processors may not know the magnitude of $\mathcal{P}$ and may assume it to be arbitrarily large, since model performance ultimately flattens out.

Experimental Set-Up

Datasets We perform experiments on two real-world datasets in the recommender system domain: MovieLens-20M (Harper and Konstan 2015) and GoogleLocal (He, Kang, and McAuley 2017; Pasricha and McAuley 2018). We sample each dataset at two sizes, to examine how results generalize across number of users and sparsity levels. Dataset statistics can be found in Table 1 and preprocessing pipelines can be found in the appendix.

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

Table 1: Each dataset is subject to the same initial, validation, and test splits, where each split is 10% of the total ratings. Each split is stratified across users. The remaining 70% of the data is the queryable rating set $\mathcal{Q}$. We produce 5 random splits of each dataset according to these divisions. All results are reported over the 5 splits.

Baselines Methods relating sample size to performance commonly assume a power law model (Hestness et al. 2017; Tae and Whang 2020). We include *Linear* to test the assumption of a power law relationship, *Initial* to determine the benefit of updating the learning curve as data is acquired, and non-linear least squares *NLS* to represent the traditional approach to fitting learning curves. Tae and Whang (2020) and Figueroa et al. (2012) use a weighted version of *NLS* to account for heteroscedastic relationship between performance and dataset size, which we term *NLSw*. Implementation details for each baseline are included below.

- *Linear*: Recommends the collection of $X\%$ of data to deliver $X\%$ of possible performance.
- *Initial*: Fits power law curve (Eq. (3)) on subsamples of $\mathcal{I}$ using non-linear least squares.
- *NLS*: Fits power law curve (Eq. (3)) on subsamples of $\mathcal{A}$ using non-linear least squares. This baseline fits the learning curve as data is collected.
- *NLSw*: Fits power law curve (Eq. (3)) on subsamples of $\mathcal{A}$, weighted by $\frac{1}{\sqrt{N}}$, where N is the sample size. This is to account for the higher variance in performance from smaller samples.

Hyperparameters We assume the model M operated by the data processor is a FunkSVD (Funk 2006) recommendation system. We use the same hyperparameters q and r across all experiments: We set the number of latent features in FunkSVD to be 30, and q is 2% of the number of queryable entries. We assume random feature acquisition unless otherwise stated.

Evaluation metrics We evaluate performance using the standard recommendation evaluation metric, mean-squared error (MSE). Additionally, we consider the fraction of performance gain (defined previously), and per-user MSE.

Implementation We fit performance curves using the OLS implementation included in the Python statsmodels library (Seabold and Perktold 2010). All experiments were run on a 2.8 GHz Intel Core i7 processor with 16GB of available RAM on MacOS High Sierra.

Experiments for Method Evaluation

Forecasting performance A successful performance-based stopping criterion relies on accurately forecasting performance given additional data. In this section, we evaluate our method's estimate of performance given the entirety of

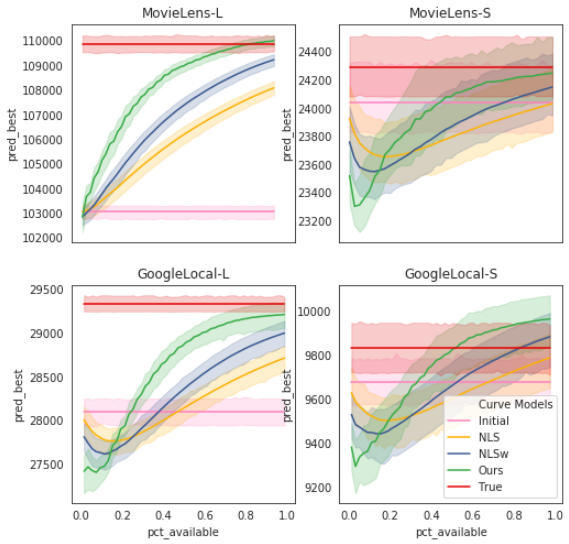

Figure 2: Predicted performance, $f_u(M, \mathcal{I} \cup \mathcal{P})$, over the course of data collection. Our method (green) matches the true performance (red) most closely at all stages of data collection. Figures best displayed in color.

$\mathcal{P}$. We show that our method significantly outperforms baselines in this task.

Figure 2 plots our results. Each method underestimates error given $\mathcal{P}$. This is in line with prior work in machine translation, which shows the underestimation of test error using curve-fitting approaches (Kolachina et al. 2012). Across datasets and sample sizes, our method estimates this quantity most accurately. This is because our method trains exclusively on larger sample sizes, ignoring smaller sample sizes completely. Importantly, this departs from the common understanding of the performance curve, which uses a single power law curve to characterize the data collection process.

Between the baselines, *NLSw* improves upon *NLS* because it down-weights smaller sample sizes. While *Initial* does not adapt to sample sizes beyond the initialized data, its estimate for the performance given $\mathcal{P}$ remains accurate for MovieLens-S. Taking a closer look, we see that this makes sense—MovieLens-S contains a larger power-law region, allowing accurate extrapolations from curves fit to smaller sample sizes.

Predicting fraction of performance How accurate is the predicted fraction of performance gain? To investigate, we look at the predicted fraction of performance by each method over the course of data collection and plot the estimates in Figure 3. We include the true relationship in red.

The predicted performance gain for all methods is more accurate at higher and lower fractions. Our method, plotted in green, comes closest to the true relationship during all stages of data collection. The baselines produce very similar estimates for fraction of performance gain over the course of data collection. *NLS* and *NLSw* produce similar estimates because their differing characteristic - down-weighting smaller sample sizes - disappears with larger sample sizes, as in the case of recommendation datasets.

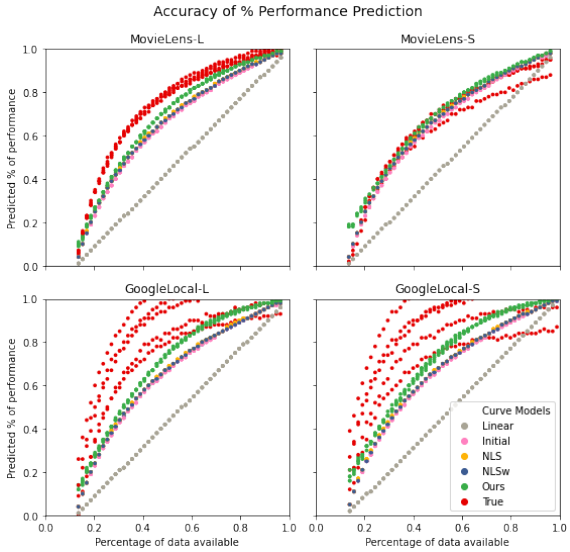

Figure 3: Comparison of our method's predicted fraction of performance $\hat{p}$ (green) to baseline predictions and the true fraction of performance p (red).

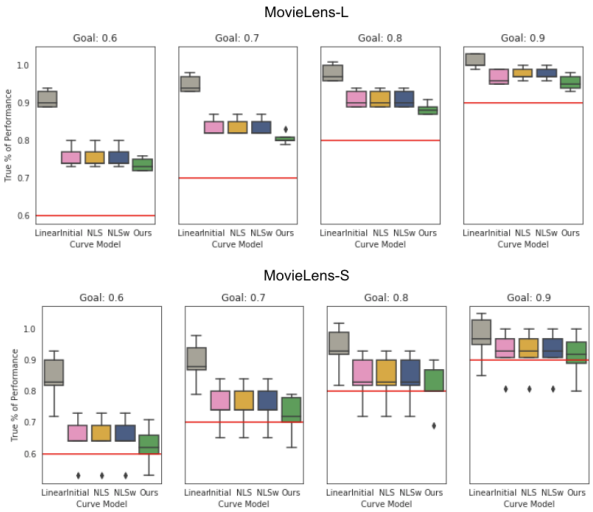

Figure 4: True performances achieved using stopping criteria given different g and different data collection methods. We see that across g, the true performance using our method (green) is closest to the goal g (red).

Performance of stopping criterion In this section, we examine the effects of using the proposed minimized data collection method. In particular, we compare the true performance achieved by our method with baselines introduced previously in Figure 4.

We evaluate each stopping criteria across multiple datasets and goal fractions of performance. We see that the proposed method accomplishes a true performance closest to the goal performance g across all datasets and goals considered. Lower goals are more difficult to adhere too, producing larger gaps between the true performances accomplished by each method and the goal performance (plotted in red).

It is worth noting that each method collects more data

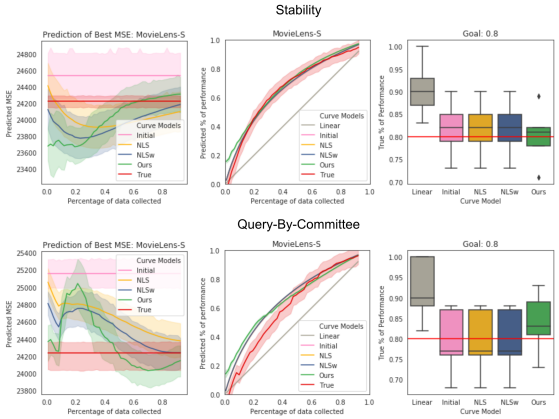

Figure 5: Robustness of minimized data collection method to different initialization assumptions. Plots for *Stability* (top row) and *QBC* Show that our method exceeds or matches the ability of baselines to predict performance (left), predict fraction of performance (center), and produce an accurate stopping criterion (right).

than is necessary for the goal performance, producing true performances that exceed g. This is a direct result of each method's underestimation of the error on the test set given the queryable set $\mathcal{P}$, as discussed in earlier sections. Minimized data collection must be conscious of this behavior, because such algorithms are liable to collect too much data for a given goal.

Robustness to Feature Acquisition Algorithms

Thus far, we have considered data collection where observations are queried randomly from $\mathcal{Q}$. AFA methods improve upon this approach by instead querying feature values based on their uncertainty (Huang et al. 2018; Freund et al. 1997; Chakraborty et al. 2013) or contribution to a downstream task (Melville et al. 2005; Vu et al. 2007). 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 experimental conditions (Munjal et al. 2020). In this section, we examine the performance of minimized data collection in the context of AFA algorithms. We find that our performance curve estimates are still more accurate than existing approaches given different feature acquisition algorithms.

Setup We consider two popular AFA methods: *Stability* and *Query-by-Committee* (*QBC*). *QBC* (Chakraborty et al. 2013) produces feature uncertainty estimates from the variance over matrix imputations by three approaches (k-NN, EM, and SVD), and *Stability* (Huang et al. 2018) estimates the uncertainty of a feature by its variance over SVD reconstructions of different ranks. Each algorithm requests the highest variance feature-values. For *Stability*, we follow the approach of (Huang et al. 2018) and set the ranks to be $[1, 2, 3]$.

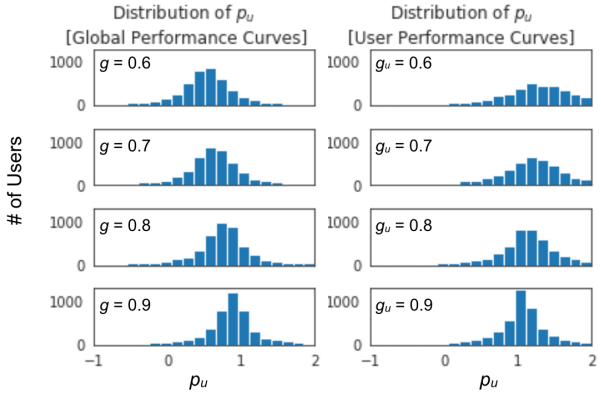

Figure 6: User-specific performance metrics when minimized data collection learns one curve for all users (left) and a curve for each user (right). Unsurprisingly, we see that the mode of user-specific performance increases as g increases. Interestingly, this is not the case when learning user-specific performance curves and suggests future areas of work for learning accurate user-specific performance curves.

Results Our method remains superior to the baselines in predicting the performance given $\mathcal{Q}$, predicting the fraction of performance gain, and enforcing a stopping criterion for both *Stability* and *QBC*. Plots for these comparisons can be found in Figure 5. It is worth noting that margin at which the proposed data collection outperforms existing approaches is larger for *Stability*. This may be attributed to committee-based approach of *QBC*; predicting the behavior of a committee including three distinct feature acquisition models is more difficult than just one, as is the case with *Stability*.

The second observation we can draw from these experiments is that existing approaches are more competitive given *Stability* and *QBC*. The performance curves for these feature acquisition algorithms can be found in the supplement and each suggest that the power law region is expanded by AFA algorithms.

Effects on Per-User Performance

Previous sections discuss minimized data collection in terms of global system performance. In this section, we examine the method'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.

Setup We contrast the performance of our method in terms of the user fraction of performance p_u in two settings. In the first, we replicate the setting discussed in previous sections and learn a performance curve for the whole dataset. In the second, the data collection method learns a performance curve *per user* and applies a stopping criterion based on goal g to each curve. We calculate user fractions of performance p_u by evaluating $\sigma(M, \mathcal{I} \cup H(\mathcal{P}, n))$ exclusively on data from user u. This produces a p_u for each user.

Results Note that the axes for each histogram extend from -1 to 2. This is because more data does not necessarily trans-

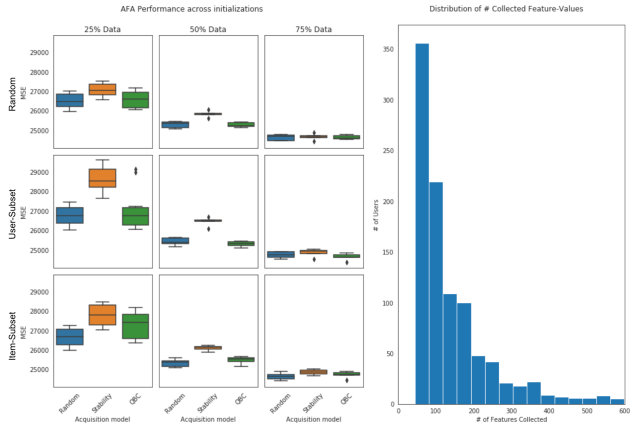

Figure 7: We show that the performance achieved during data collection depends on both the AFA algorithm employed and the initialization conditions (left). Error bars are reported over 5 random initializations. We also show that a small portion of users bear the majority of the data collection burden in a histogram of the quantity of features acquired per user by *Stability* from MovieLens-S halfway through data collection.

late 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 could still hurt user performance if this data is not representative. In this setting, the assumption of monotonically increasing performance over data collection does not hold, and accordingly, the minimized data collection method does not perform as well. One key takeaway from this experiment is that the fraction of performance may not be an appropriate metric for data minimization on a per user level. We include comments on how the proposed data collection method may be further adapted to the per user setting in the appendix.

Impact of Active Feature Acquisition on Users

Our minimized data collection method relies on an underlying feature acquisition algorithm and AFA is a natural choice for limited data collection. In this section, we illuminate several impacts of standard AFA algorithms on data subjects.

Disparate data collection burden First of all, AFA algorithms collect different number of features from different users. Figure 7 (right), plots a histogram of the quantity of collected data over users for AFA algorithm *Stability*, for dataset MovieLens-S (similar trends exist for other setups).

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. Thus, the overall behavior of AFA in the context of data minimization raises questions of both the user fairness as well as user agency. Should users be able to decide whether they would like to become high-collection users in exchange for higher performance? How marginal must this improvement be before to no longer justify data collection?

Sensitivity to initial system data In Figure 7 (left), we examine the dependence of minimized recommendation 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 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.

Several observations have important consequences for the practice of data minimization. Note that 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, non-random AFA begin decreasing in performance. This observation is consequential in cases where (i) the population of data subjects is evolving (initialization data would not contain the data of users who join at a later time), and (ii) the data processor is not initially allowed to collect certain feature values because of other constraints (such as the feature being sensitive).

Discussion

This work addresses the lack of technical interpretations of the legal requirement of data minimization. Our findings offer takeaways for researchers, data processors, and legal scholars. We propose a method for data minimization during data collection that relies on adaptively learning the relationship between dataset size and performance and accounts for three distinct stages of data collection. Each data collection stage is characterized by quantitatively different contributions of new data to the system performance, with the second stage allowing users to trade data for improved performance, and the last stage characterized by no significant performance increase and thus offering a definite stopping point. Moreover, we find that, because of these quantitative changes in data collection characteristics, algorithms learning performance-based stopping criteria tend to overestimate the the amount of data necessary to meet a target performance. Last but not least, we demonstrate that AFA methods may incur unequal distribution of collected data across users, and decreased performance for evolving communities or in cases where certain features are excluded from initial data collection.

While our paper provides new methods, insights and practical guidelines for implementing data minimization in practice in automated decision-making and profiling systems, several open questions remain. Those include, for instance, designing better per-user minimization methods, automatic detection of data collection stages, and studying the technical interplay between data minimization and other data protection principles.

References

Balcan, M.-F.; Hanneke, S.; and Vaughan, J. W. 2010. The true sample complexity of active learning. *Machine learning* 80(2-3): 111–139.

Bhargava, A.; Ganti, R.; and Nowak, R. 2017. Active Positive Semidefinite Matrix Completion: Algorithms, Theory and Applications. volume 54 of *Proceedings of Machine Learning Research*, 1349–1357. Fort Lauderdale, FL, USA: PMLR.

Biega, A. J.; Potash, P.; Daumé, H.; Diaz, F.; and Finck, M. 2020. Operationalizing the Legal Principle of Data Minimization for Personalization. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, 399–408.

Binns, R.; and Gallo, V. 2019. Data minimisation and privacy-preserving techniques in AI systems. URL https://ico.org.uk/about-the-ico/news-and-events/ai-blog-data-minimisation-and-privacy-preserving-techniques-in-ai-systems/.

Chakraborty, S.; Zhou, J.; Balasubramanian, V.; Panchanathan, S.; Davidson, I.; and Ye, J. 2013. Active Matrix Completion. 81–90. ISBN 978-0-7695-5108-1. doi: 10.1109/ICDM.2013.69.

Cho, J.; Lee, K.; Shin, E.; Choy, G.; and Do, S. 2015. Medical Image Deep Learning with Hospital PACS Dataset. *CoRR* abs/1511.06348. URL http://arxiv.org/abs/1511.06348.

Chow, R.; Jin, H.; Knijnenburg, B.; and Saldamli, G. 2013. Differential data analysis for recommender systems. In *Proceedings of the 7th ACM conference on Recommender systems*, 323–326.

Datatilsynet, Norwegian Data Protection Authority. 2018. Artificial Intelligence and Privacy. URL https://www.datatilsynet.no/globalassets/global/english/ai-and-privacy.pdf.

Dobbin, K. K.; Zhao, Y.; and Simon, R. M. 2008. How Large a Training Set is Needed to Develop a Classifier for Microarray Data? *Clinical Cancer Research* 14(1): 108–114.

Domhan, T.; Springenberg, J. T.; and Hutter, F. 2015. Speeding up automatic hyperparameter optimization of deep neural networks by extrapolation of learning curves. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

Ehrenfeucht, A.; Haussler, D.; Kearns, M.; and Valiant, L. 1989. A general lower bound on the number of examples needed for learning. *Information and Computation* 82(3): 247–261.

Figueroa, R. L.; Zeng-Treitler, Q.; Kandula, S.; and Ngo, L. H. 2012. Predicting sample size required for classification performance. *BMC Medical Informatics and Decision Making* 12(1). doi:10.1186/1472-6947-12-8. URL https://doi.org/10.11862F1472-6947-12-8.

Freund, Y.; Seung, H. S.; Shamir, E.; and Tishby, N. 1997. Selective sampling using the query by committee algorithm. *Machine learning* 28(2-3): 133–168.

Funk, S. 2006. Netflix update: Try this at home. *https://sifter.org/ simon/journal/20061211.html* .

Galdon Clavell, G.; Martín Zamorano, M.; Castillo, C.; Smith, O.; and Matic, A. 2020. Auditing Algorithms: On Lessons Learned and the Risks of Data Minimization. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, 265–271.

GDPR. 2016. REGULATION (EU) 2016/679 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL. *Official Journal of the European Union* .

Gürses, S.; Troncoso, C.; and Diaz, C. 2015. Engineering privacy by design reloaded. In *Amsterdam Privacy Conference*, 1–21.

Hajian-Tilaki, K. 2014. Sample size estimation in diagnostic test studies of biomedical informatics. *Journal of Biomedical Informatics* 48: 193 – 204. ISSN 1532-0464. doi: https://doi.org/10.1016/j.jbi.2014.02.013. URL http://www.sciencedirect.com/science/article/pii/S1532046414000501.

Harper, F. M.; and Konstan, J. A. 2015. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)* 5(4): 1–19.

He, R.; Kang, W.-C.; and McAuley, J. 2017. Translation-based recommendation. In *Proceedings of the eleventh ACM conference on recommender systems*, 161–169.

Heckel, R.; and Ramchandran, K. 2017. The sample complexity of online one-class collaborative filtering. *arXiv preprint arXiv:1706.00061* .

Hestness, J.; Narang, S.; Ardalani, N.; Diamos, G.; Jun, H.; Kianinejad, H.; Patwary, M.; Yang, Y.; and Zhou, Y. 2017. Deep Learning Scaling is Predictable, Empirically. *arXiv preprint arXiv:1712.00409* .

Huang, S.-J.; Xu, M.; Xie, M.-K.; Sugiyama, M.; Niu, G.; and Chen, S. 2018. Active feature acquisition with supervised matrix completion. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 1571–1579.

ICO, Information Commissioner's Office. 2017. Big Data, Artificial Intelligence, Machine Learning and Data Protection. URL https://ico.org.uk/media/for-organisations/documents/2013559/big-data-ai-ml-and-data-protection.pdf.

Kalayeh, H.; and Landgrebe, D. A. 1983. Predicting the required number of training samples. *IEEE transactions on pattern analysis and machine intelligence* (6): 664–667.

Kolachina, P.; Cancedda, N.; Dymetman, M.; and Venkatapathy, S. 2012. Prediction of learning curves in machine translation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 22–30.

Krause, A.; and Horvitz, E. 2010. A utility-theoretic approach to privacy in online services. *Journal of Artificial Intelligence Research* 39: 633–662.

Manski, C. F.; and Tetenov, A. 2016. Sufficient trial size to inform clinical practice. *Proceedings of the National Academy of Sciences* 113(38): 10518–10523.

Mavroforakis, C.; Erdös, D.; Crovella, M.; and Terzi, E. ???? *Active Positive-Definite Matrix Completion*, 264–272. doi: 10.1137/1.9781611974973.30. URL https://epubs.siam.org/doi/abs/10.1137/1.9781611974973.30.

Melville, P.; Saar-Tsechansky, M.; Provost, F.; and Mooney, R. 2005. An expected utility approach to active feature-value acquisition. In *Fifth IEEE International Conference on Data Mining (ICDM'05)*, 4–pp. IEEE.

Munjal, P.; Hayat, N.; Hayat, M.; Sourati, J.; and Khan, S. 2020. Towards Robust and Reproducible Active Learning Using Neural Networks. *arXiv* arXiv–2002.

Pasricha, R.; and McAuley, J. 2018. Translation-based factorization machines for sequential recommendation. In *Proceedings of the 12th ACM Conference on Recommender Systems*, 63–71.

Schein, A. I.; Popescul, A.; Ungar, L. H.; and Pennock, D. M. 2002. Methods and Metrics for Cold-Start Recommendations. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '02, 253–260. New York, NY, USA: Association for Computing Machinery. ISBN 1581135610. doi:10.1145/564376.564421. URL https://doi.org/10.1145/564376.564421.

Seabold, S.; and Perktold, J. 2010. statsmodels: Econometric and statistical modeling with python. In *9th Python in Science Conference*.

Senarath, A.; and Arachchilage, N. A. G. 2018. Understanding Software Developers' Approach towards Implementing Data Minimization. *arXiv preprint arXiv:1808.01479* .

Suresh, K.; and Chandrashekara, S. 2012. Sample size estimation and power analysis for clinical research studies. *Journal of human reproductive sciences* 5(1): 7.

Sutherland, D. J.; Póczos, B.; and Schneider, J. 2013. Active Learning and Search on Low-Rank Matrices. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, 212–220.

Tae, K. H.; and Whang, S. E. 2020. Slice Tuner: A Selective Data Collection Framework for Accurate and Fair Machine Learning Models. *arXiv preprint arXiv:2003.04549* .

Vincent, N.; Hecht, B.; and Sen, S. 2019. "Data Strikes": Evaluating the Effectiveness of a New Form of Collective Action Against Technology Companies. In *The World Wide Web Conference*, 1931–1943. ACM.

Vu, D.; Bilenko, M.; Saar-tsechansky, M.; and Melville, P. 2007. Intelligent Information Acquisition for Improved Clustering.

Wen, H.; Yang, L.; Sobolev, M.; and Estrin, D. 2018. Exploring recommendations under user-controlled data filtering. In *Proceedings of the 12th ACM Conference on Recommender Systems*, 72–76. ACM.

Xiao, X.; White, E. P.; Hooten, M. B.; and Durham, S. L. 2011. On the use of log-transformation vs. nonlinear regression for analyzing biological power laws. *Ecology* 92(10): 1887–1894.

Yan, S.; Chaudhuri, K.; and Javidi, T. 2019. The label complexity of active learning from observational data. In *Advances in Neural Information Processing Systems*, 1810–1819.

Zarsky, T. Z. 2017. Incompatible: The GDPR in the Age of Big Data. *Seton Hall Law Review* 47(4): 2.