Learning to Limit Data Collection via Scaling Laws: Supplementary Material

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

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

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

Michèle Finck
webmaster@marysville-ohio.com
University of Tübingen
Germany

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

ACM Reference Format:
Divya Shanmugam, Fernando Diaz, Samira Shabanian, Michèle Finck, and Asia J. Biega. 2022. Learning to Limit Data Collection via Scaling Laws: Supplementary Material. In *2022 ACM Conference on Fairness, Accountability, and Transparency (FAccT '22), June 21–24, 2022, Seoul, Republic of Korea.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3531146.3533148

1 EXPERIMENTAL DETAILS

We include relevant details to our experimentation below and provide code to reproduce all results.

1.1 Dataset Pre-Processing

For MovieLens-20M, we filter out users with fewer than 100 movies rated to produce a dataset of 51869 users and 26654 unique movies. MovieLens-L randomly samples 5000 users from this set and MovieLens-S samples 1000 users. For GoogleLocal, we filter out users with fewer than 150 ratings to produce a dataset of 1571 users and 275402 unique places. GoogleLocal-L randomly samples 1500 users from this set and GoogleLocal-S samples 500 users. Table 1 summarizes the dimensionality and sparsity of the resulting datasets.

1.2 Parameter Fitting

In line with past work, our method and each baseline is subject to the same parameter fitting approach: weighted non-linear least squares (NLSw). NLSw weights each sample by $(1/\sqrt{n})$, where n is the sample size. This is done to account for the heteroscedastic relationship between performance and dataset size.

1.3 Implementation

We fit performance curves using the OLS implementation included in the Python statsmodels library [Seabold and Perktold 2010] and the Levenberg-Marquardt algorithm via the least-squares implementation in the Python sklearn library [Pedregosa et al. 2011]. All experiments were run on a 2.8 GHz Intel Core i7 processor with 16GB of available RAM on MacOS High Sierra.

1.4 Choice of thresholds

We evaluate each method at three thresholds for the return on additional data: 5e-7, 2e-7, and 5e-8. We select these thresholds because they translate to thresholds that are distributed over the data collection process (see Figure **??** to observe typical range of empirical returns for MovieLens-L and GoogleLocal-L).

1.5 Implementation of Oracle

To make our calculation of the slope using for the *Oracle* baseline explicit, recall that we generate $|\mathcal{A}|/q$ subsamples over the course of data collection. To estimate the return in MSE at subsample i, we divide the difference in empirical performance at subsample $i-2$ and subsample $i+2$ by their difference in sample sizes, or $4q$. This produces an approximation of the true return that is sufficiently smooth; we experiment with a different smoothing procedure (estimation via subsample $i-1$ and subsample $i+1$) later in the supplemental material.

2 SUPPORTING FIGURES

2.1 Minimization on Returns: Consistent Results

We plot consistent results in Figure 1. These results align with those discussed in the body of the paper; while our method appears noisier, it reflects the noise of the empirically derived return in additional data. If one were to minimize with respect to a less noisy metric than performance given additional examples, this curve may look smoother. This presents yet another tradeoff between the baselines and the proposed method: while our method produces stopping points most accurate to the empirical return on additional data, the baselines offer a "smoother" decision rule.

2.2 Comparisons to Other Curve Models

2.2.1 Ablation of Stages in Piecewise Power Law. FIDO uses a piecewise power law technique that consists of three power laws, or one

*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

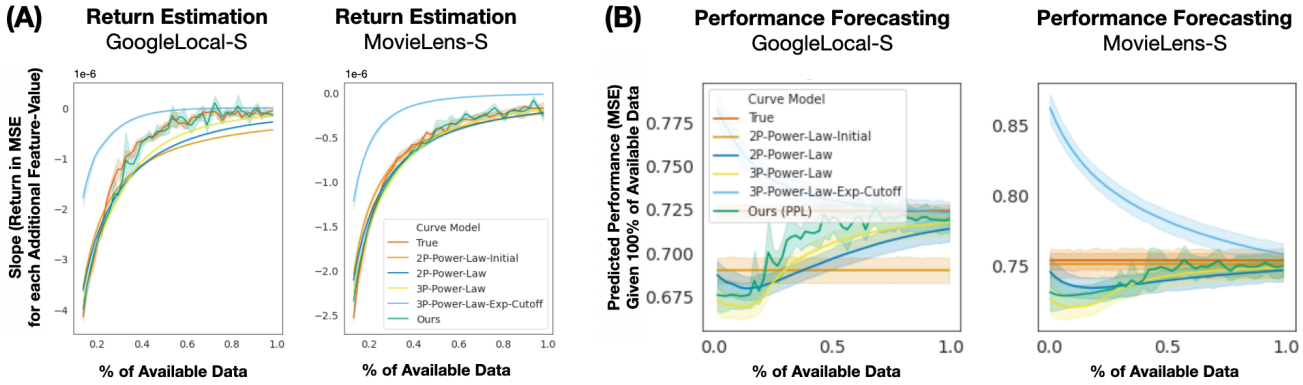

Figure 1: Evaluation of performance curves over the course of data collection. We plot the diminishing returns results for MovieLens-S and GoogleLocal-S visually here. Note that the margin with which we outperform baselines is smaller in the case of MovieLens-S; this is due to the smaller diminishing returns region, which can be seen in Figure 4.

for each stage of data collection outlined by [Hestness et al. 2017]. In this section, we provide an ablation study of the necessity of each stage by testing FIDO with one power law, two power laws, and three power laws (Table 1). In short, the use of three power laws produces more accurate stopping criteria *on average*, but the stopping criteria are not significantly different from modeling two stages. This is because the "small data region" is negligibly small in most cases, and performance is not affected by modeling the small data region and the power law region as one power law.

2.2.2 Nonparametric Regression. Each of the baselines considered are parametric approximations of the relationship between dataset size and performance. Past work in learning curve estimation considers the value of nonparametric approaches. Here, we explore the value of such an approach and find that the estimated performance curves are much noisier than those produced by parametric curve models. Table 2 shows that FIDO achieves more accurate stopping criteria across datasets and thresholds, by a large margin.

2.3 Performance Curves for GoogleLocal-S and MovieLens-S

We include the performance curves for each dataset considered in this work in Figure 3. We include the performance curves for MovieLens-L and GoogleLocal-L, as shown in Figure **??** for ease of comparison.

2.4 Performance Curves for Active vs. Random Feature Acquisition

We include the true performance curve using *Stability* as a feature acquisition algorithm in Figure 4. Note the noise introduced over runs compared to the performance curves produced by random feature acquisition (Figure **??**).

2.5 Illustrative Performance Curve Fits

We measure the quality of the performance curve fit via metrics relevant to our data minimization objectives (extrapolations and slope estimation) in the body of the paper and support these results with a visualization of performance curve fits (Figure 5). We plot each method's performance curve fit halfway through data collection from GoogeLocal-L. Examining the predicted performance curve (orange) for each baseline, we can see that it departs from the true performance curve most towards the end of data collection, due to differences in modeling the stage of diminishing returns.

3 ALTERNATE DATA MINIMIZATION OBJECTIVE: MINIMIZATION BY RELATIVE PERFORMANCE

In the body of the paper, we consider one data minimization objective: minimization based on the return of additional data. FIDO is flexible to other performance-based objectives as well. In this section, we explore one such objective: minimization based on the *relative performance*, which stops data collection based on a percentage of the model's performance given infinite data.

Definition. Minimization in reference to a relative performance rather than an absolute performance may be preferable 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. Parameter g represents the goal *fraction* of performance gain and dictates 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. Under the considered scenario, the data minimization task is to collect the minimum number n of user-feature-value triples from queryable set $\mathcal{P}$ such that the fraction of performance gain achieved equals at least $g \in [0, 1]$:

$$\frac{\sigma_M - \sigma_M}{\sigma_M - \sigma_M} \geq g \tag{1}$$

Results. Minimization by relative performance requires approximating the percentage of performance gain using the performance curve learned in Step 2 and ceasing data collection once the predicted performance gain exceeds user-specified goal g. The proposed

Dataset	Threshold	FIDO (1 Stage)	FIDO (2 Stages)	FIDO (3 Stages)	Oracle
GoogleLocal-L	-5.0e-07	0.32 ± 0.01	0.29 ± 0.02	0.29 ± 0.02	0.27 ± 0.01
GoogleLocal-L	-2.0e-07	0.61 ± 0.01	0.42 ± 0.03	0.42 ± 0.03	0.41 ± 0.02
GoogleLocal-L	-5.0e-08	1.00 ± 0.00	0.65 ± 0.04	0.68 ± 0.05	0.68 ± 0.05
GoogleLocal-S	-5.0e-07	0.71 ± 0.01	0.42 ± 0.06	0.42 ± 0.06	0.47 ± 0.03
GoogleLocal-S	-2.0e-07	1.00 ± 0.00	0.48 ± 0.09	0.51 ± 0.14	0.58 ± 0.06
GoogleLocal-S	-5.0e-08	1.00 ± 0.00	0.56 ± 0.04	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
MovieLens-L	-2.0e-07	0.27 ± 0.00	0.24 ± 0.01	0.25 ± 0.02	0.23 ± 0.01
MovieLens-L	-5.0e-08	0.76 ± 0.01	0.52 ± 0.05	0.53 ± 0.05	0.53 ± 0.02
MovieLens-S	-5.0e-07	0.53 ± 0.01	0.46 ± 0.05	0.46 ± 0.05	0.45 ± 0.03
MovieLens-S	-2.0e-07	1.00 ± 0.00	0.67 ± 0.12	0.68 ± 0.14	0.79 ± 0.08
MovieLens-S	-5.0e-08	1.00 ± 0.00	0.97 ± 0.07	0.99 ± 0.03	1.00 ± 0.00

Table 1: Ablation of the Number of Stages in FIDO's Performance Curve Model. We vary the number of stages we assume to exist in the performance curve between 1 (assuming the entire performance curve can be described by a power law) and 3 (aligning with the number of stages described by [Hestness et al. 2017]. FIDO achieves the closest stopping criterion (on average) to Oracle when the framework models three stages of data collection.

Dataset	Threshold	NonParam	FIDO	Oracle
GoogleLocal-L	-5.0e-07	0.28 ± 0.01	0.29 ± 0.02	0.27 ± 0.01
GoogleLocal-L	-2.0e-07	0.30 ± 0.01	0.42 ± 0.03	0.41 ± 0.02
GoogleLocal-L	-5.0e-08	0.30 ± 0.01	0.68 ± 0.05	0.68 ± 0.05
GoogleLocal-S	-5.0e-07	0.32 ± 0.01	0.42 ± 0.06	0.47 ± 0.03
GoogleLocal-S	-2.0e-07	0.32 ± 0.01	0.51 ± 0.14	0.58 ± 0.06
GoogleLocal-S	-5.0e-08	0.32 ± 0.02	0.59 ± 0.08	0.64 ± 0.10
MovieLens-L	-5.0e-07	0.15 ± 0.01	0.13 ± 0.00	0.13 ± 0.00
MovieLens-L	-2.0e-07	0.25 ± 0.01	0.25 ± 0.02	0.23 ± 0.01
MovieLens-L	-5.0e-08	0.28 ± 0.01	0.53 ± 0.05	0.53 ± 0.02
MovieLens-S	-5.0e-07	0.39 ± 0.01	0.46 ± 0.05	0.45 ± 0.03
MovieLens-S	-2.0e-07	0.39 ± 0.01	0.68 ± 0.14	0.79 ± 0.08
MovieLens-S	-5.0e-08	0.39 ± 0.01	0.99 ± 0.03	1.00 ± 0.00

Table 2: Comparison of FIDO to non-parametric regression. FIDO achieves more accurate stopping criteria than nonparametric regression across datasets and thresholds.

method ceases data collection at a point closest to the goal percentage compared to baselines and its performance offers insight into the reliability of relative-performance-based stopping criteria.

We evaluate each stopping criterion across multiple datasets and goal fractions of performance $g \in [.5, .6, .7, .8, .9]$. The plots show that lower goals are more difficult to adhere to, producing larger gaps between the true performances accomplished by each method and the goal performance (plotted in red). While the method performs significantly better than baselines in some cases (particularly for higher goal percentages on GoogleLocal-L), we cannot confidently conclude that the proposed method produces a significantly better stopping criteria across datasets. This suggests the selection of the appropriate performance curve model may depend both on the dataset and the data minimization objective.

It is worth noting that most methods, save for *3P-PL-Exp*, collect more data than is necessary for the goal performance, producing true performance gains 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.

4 ADDITIONAL RELATED WORK

4.1 Active Feature Acquisition (AFA).

AFA concerns the intelligent collection of feature-values given a fixed budget. There are several well-known AFA techniques: (i) using matrix completion with the assumption of low rank [Bhargava et al. 2017; Huang et al. 2018; Mavroforakis et al. 2017] (ii) estimating the expected accuracy improvement for tasks such as clustering or classification [Melville et al. 2005; Vu et al. 2007] (iii) applying techniques to address ice-start [Gong et al. 2019] and cold-start problems [Schein et al. 2002] and (iv) using variational techniques to approximate the posterior distribution [Sutherland et al. 2013]. The performance of AFA is often evaluated based on a fixed budget. FIDO lies adjacent to this area in that the framework aims to learn a performance-based budget *given* a feature acquisition algorithm.

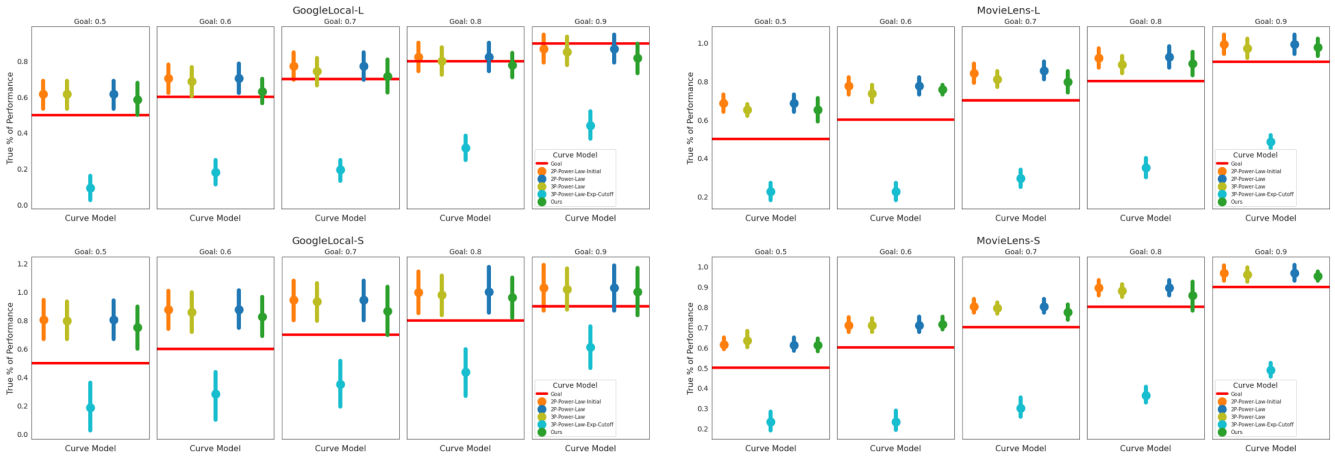

Figure 2: Performance on Relative Performance Minimization The proposed method (green) lands closest to the goal percentage of relative performance on average, compared to existing methods. While the piecewise power law produces a power law curve that is significantly more accurate than others in *forecasting* performance, the estimates of relative performance remain comparable to baselines.

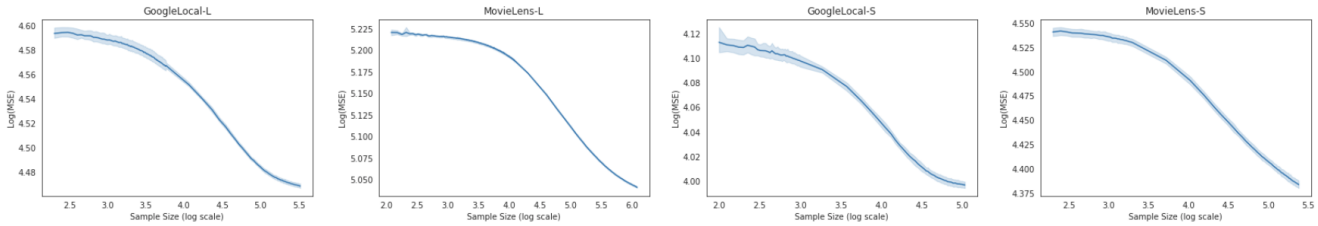

Figure 3: Performance Curves given Random Feature Acquisition Each graph represents the improvement in performance over the course of data collection. Note the smaller diminishing returns region for MovieLens-S.

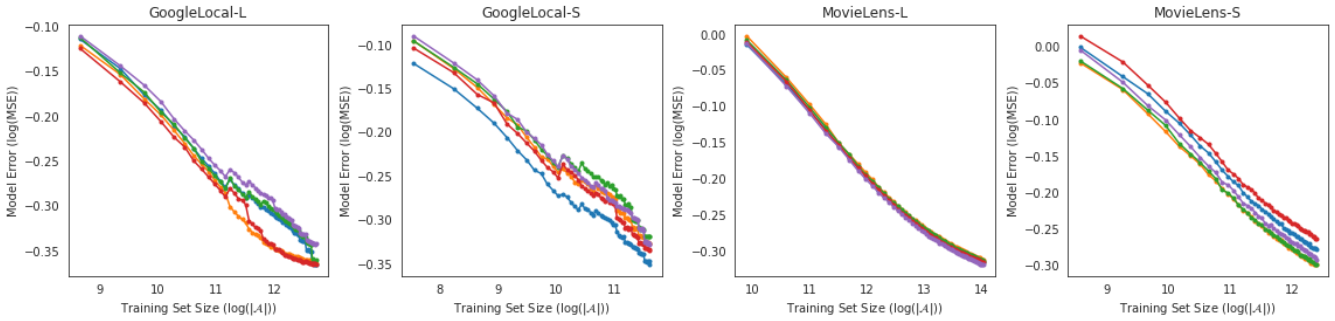

Figure 4: Performance curves using AFA method *Stability*. Each line represents an individual split of the dataset.

4.2 Sample Complexity.

Sample complexity corresponds to the number of samples required for a learning algorithm to achieve an error rate of ϵ with a probability of $(1-\delta)$. Literature on sample complexity takes a theoretical approach to the central task of our work: tying data collection to model performance. Earlier work quantifies sample complexity assuming random sample acquisition [Ehrenfeucht et al. 1989], while more recent work studies sample complexity for active learning methods [Balcan et al. 2010; Yan et al. 2019] and recommendation systems [Heckel and Ramchandran 2017]. Hestness et al. [2017] note the gap between theoretical guarantees and empirical trends for performance curves and highlight this as an area for future work.

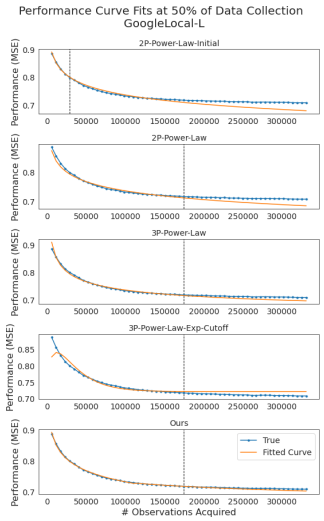

Figure 5: Performance curve fits halfway through data collection for our method and the 4 baselines considered. The black dotted line denotes the amount of data provided to each performance curve fit—note that *2P-Power-Law-Initial* sees no more than the initial dataset.

REFERENCES

Maria-Florina Balcan, Steve Hanneke, and Jennifer Wortman Vaughan. 2010. The true sample complexity of active learning. *Machine learning* 80, 2-3 (2010).

Aniruddha Bhargava, Ravi Ganti, and Rob Nowak. 2017. Active Positive Semidefinite Matrix Completion: Algorithms, Theory and Applications, Aarti Singh and Jerry Zhu (Eds.), Vol. 54. PMLR, 1349–1357.

Andrzej Ehrenfeucht, David Haussler, Michael Kearns, and Leslie Valiant. 1989. A general lower bound on the number of examples needed for learning. *Information and Computation* 82, 3 (1989), 247–261.

Wenbo Gong, Sebastian Tschiatschek, Richard Turner, Sebastian Nowozin, José Miguel Hernández-Lobato, and Cheng Zhang. 2019. Icebreaker: Element-wise Active Information Acquisition with Bayesian Deep Latent Gaussian Model. In *NeurIPS-33*.

Reinhard Heckel and Kannan Ramchandran. 2017. The Sample Complexity of Online One-Class Collaborative Filtering. *CoRR* abs/1706.00061 (2017).

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).

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.

Charalampos Mavroforakis, Dóra Erdös, Mark Crovella, and Evimaria Terzi. 2017. Active Positive-Definite Matrix Completion. In *SDM'17*. 264–272.

Prem Melville, Maytal Saar-Tsechansky, Foster Provost, and Raymond Mooney. 2005. An Expected Utility Approach to Active Feature-Value Acquisition. In *ICDM-05*. IEEE, 745–748.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

Andrew I. Schein, Alexandrin Popescul, Lyle H. Ungar, and David M. Pennock. 2002. Methods and Metrics for Cold-Start Recommendations. In *ACM(25) SIGIR*. Association for Computing Machinery, 253–260.

Skipper Seabold and Josef Perktold. 2010. statsmodels: Econometric and statistical modeling with python. In *9th Python in Science Conference*.

Dougal J. Sutherland, Barnabás Póczos, and Jeff Schneider. 2013. Active Learning and Search on Low-Rank Matrices. In *ACM SIGKDD-13*. 212–220.

Duy Vu, Mikhail Bilenko, Maytal Saar-tsechansky, and Prem Melville. 2007. Intelligent Information Acquisition for Improved Clustering.

Songbai Yan, Kamalika Chaudhuri, and Tara Javidi. 2019. The label complexity of active learning from observational data. In *NeurIPS*. 1810–1819.