Different Approaches to Fitting and Extrapolating the Learning Curve

Donghwi Kim and Tom Viering

Delft University of Technology

Abstract. Learning curves plot performance of a model against its training set size. Extrapolating this curve can estimate how much data is needed to achieve a desired performance. To extrapolate a learning curve, we fit two parametric models (log2, pow4) to the learning curve. We are the first study to compare learning curve extrapolation performance of existing curve fitting techniques such as Newton's method and Levenberg-Marquardt. We also introduce a smart initialisation based on KMeans to speed up curve fitting and show its potential.

Keywords: learning curve · curve fitting · Levenberg–Marquardt · Newton's method

1 Introduction

In this thesis abstract, we summarise our main findings of [5]. The learning curve plots the performance of the model versus the training set size. Extrapolating the learning curve can give us an idea of how much data is needed to achieve a particular performance [2]. This is especially useful if data collection is expensive [7]. To extrapolate the learning curve, first, a parametric model is fitted to the learning curve. In literature, it is standard to use Levenberg-Marquardt (LM).

Little studies up till now have investigated different techniques for learning curve fitting [7]. We investigate the pros and cons of LM and compare it with Newton's method for curve fitting [3, 4]. It is standard to initialise curve fitting techniques with random values. We also introduce a smart initialisation based on KMeans to speed up learning curve fitting.

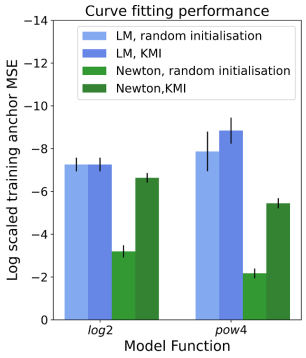
2 KMeans Initialization (KMI)

The crucial assumption behind our initialisation is that learning curves often look similar. This means that parameters that model the learning curve will be similar from one learning curve to the next. This is what we aim to exploit to speed up learning curve fitting. To this end, we use a small database of learning curves. We perform fitting on these learning curves to find the optimal learning curve parameters. We make a small database of these learning curve parameters, and perform K-means clustering on them. The found cluster centres are used as initialisation points for new learning curve fits. Crucially, the learning curves used to build the database, are not used in the evaluation.

3 Experimental Setup

Learning curves for evaluation were generated from 5 datasets from OpenML using 12 SciPy learners with the help of the learning curve database (LCDB) [6, 8, 1]. We used 213 datasets and K = 10 (KMeans) for KMI to derive the initial parameter values. We use parametric models `log2` ($y = a\log(x) + b$) and `pow4` ($y = c - (-ax + b)^{-\alpha}$) to fit the curves [7] (where y is the error rate on the test set, x is the trainset size, a, b, c, α parameters). The MSE is the fitting objective.

4 Results



Extrapolation performance

Initialisation		Random		KMI	
Fitting method		Newton	LM	Newton	LM
Bucket	Model				
0.05 ~0.1	*log2*	0.18	0.22	0.32	0.22
	pow4	0.08	0.67	0.66	0.68
0.1~0.2	*log2*	0.2	0.33	0.38	0.33
	pow4	0.08	0.77	0.72	0.8
0.2~0.4	*log2*	0.29	0.43	0.43	0.43
	pow4	0.12	0.87	0.81	0.89
0.4~0.8	*log2*	0.09	0.68	0.54	0.68
	pow4	0.17	0.94	0.9	0.96

Fig. 1. Left: average curve fitting performance over 5 datasets and 12 learners. Right: average extrapolation performance. Detailed information can be found in [5].

Unsurprisingly, simple learning curve models (with fewer parameters) are easier to fit and take less computation time. Compared to the Newton method, LM has lower MSE on curve fitting on the points used for fitting, but generally takes longer to converge. KMI has a lower MSE on the points used for fitting compared to random initialisation, especially for the complicated learning curve models with many parameters (shown in Fig.1). It was also notable that the curve fitting performance gap between KMI and the random initiation method was greater for the Newton method than LM.

We also evaluated performance for the task of extrapolating the learning curve to a larger training set size (shown in Fig.1). In that case, the learning curve model is not fitted on the point that is being evaluated. Unfortunately, in this case, the models that are easy to fit (with fewer parameters) do not perform well compared to more complex models. We find that LM generally has a lower MSE than Newton. Furthermore, KMI has a lower MSE than random.

5 Conclusion

We have provided some preliminary results showing the potential of KMI instead of randomly initialising the parameter values. Furthermore, LM seems to work overall well for curve fitting, but can be slow. Considering the computation time, Newton's method using KMI may prove to be a viable alternative to LM.

References

1. Giuseppe Casalicchio, Jakob Bossek, Michel Lang, Dominik Kirchhoff, Pascal Kerschke, Benjamin Hofner, Heidi Seibold, Joaquin Vanschoren, and Bernd Bischl. OpenML: An R package to connect to the machine learning platform OpenML. *Computational Statistics*, pages 1–15, 2017.
2. Lewis J. Frey and Douglas H. Fisher. Modeling decision tree performance with the power law. In David Heckerman and Joe Whittaker, editors, *Proceedings of the Seventh International Workshop on Artificial Intelligence and Statistics*, volume R2 of *Proceedings of Machine Learning Research*. PMLR, 03–06 Jan 1999. Reissued by PMLR on 20 August 2020.
3. Henri P Gavin. The levenberg-marquardt algorithm for nonlinear least squares curve-fitting problems. *Department of Civil and Environmental Engineering, Duke University*, 19, 2019.
4. Igor Griva, Stephen Nash, and Ariela Sofer. *Linear and Nonlinear Optimization (2. ed.)*. 01 2008.
5. Donghwi Kim, Tom Viering, and Marco Loog. Different approaches to fitting and extrapolating the learning curve. 2022. Bachelor Thesis available at http://resolver.tudelft.nl/uuid:25107a29-606e-4f76-bfa0-d1884fa4da02.
6. Felix Mohr, Tom J Viering, Marco Loog, and Jan N van Rijn. LCDB 1.0: An extensive learning curves database for classification tasks. In *ECML*, page accepted, 2022.
7. Tom Viering and Marco Loog. The shape of learning curves: a review. *arXiv preprint arXiv:2103.10948*, 2021.
8. Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.