LEARNING "LEARNING CURVES"

Author1, Author2
Affiliation
Univ
City
{Author1, Author2}email@email

Author3
Affiliation
Univ
City
email@email

ABSTRACT

***K*eywords** meta-learning · expected performance · learning-curves · functional analysis

1 Introduction

The asymptotic generalization performance of a learner is of utmost importance since practical machine learning is bound to a finite setting. A learning curve shows the generalization performance of a learner concerning training set size. Learning curves have numerous applications in machine learning pipelines (decrease data collection cost, model selection, hyperparameter tuning, etc.). The training set size creates an evident bottleneck since the computational effort that goes into training increases with the training set size. (Although, numerous methods try to circumvent that bottleneck both for shallow and deep machine learning models [1, 2].) This computational bottleneck prevents a reliable learning curve generation. Thus, the need for interpolation and extrapolation of these curves gains importance.

Historically learning curves are mostly modeled by parametric models, namely exponential, power law, etc., to extrapolate the generalization performance concerning training set size. (See [3] for detailed references to these parametric methods.) However, for all of these parametric models, there exists an intrinsic monotonicity assumption. As shown in the detailed survey [4], expecting a learner to have a monotone generalization error decrease with increased training set size is not reasonable for a given dataset. This is why we propose a data-driven approach for learning curve extrapolation, hoping that will allow us to break free from the monotonicity assumptions stemming from parametric models.

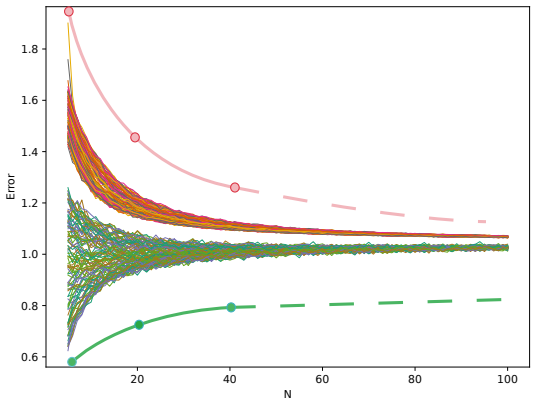

Figure 1: Learning curve extrapolation visualization for both training and test performance with an error measure concerning the training set size N.

Fortunately, extensive learning curve databases are getting publicly available which facilitates a data-driven approach to extrapolating learning curves. A data-driven modeling approach to learning curves requires the treatment of learning curve creation as a learning problem in itself. To be precise we will consider a setting where there are available full learning curves from different learners on different datasets, different learners from the same dataset, and a learner for the same dataset with changing hyper-parameters. In addition to full learning curves, we are given a few data points (here few data points refer to *error measure-training set size* pairs). The aim will be to extrapolate this curve to increased training set sizes given the generalization performance of the smaller training set sizes of a learner.

This paper utilizes functional analysis to extract valuable meta-information that is coming from a database of learning curves via Functional Principal Component Analysis (FPCA) and linearly combines these components while minimizing the squared loss over the observed data points.

2 Related Work

- Database creation.
- Non-monotonicity of learning curves
- Learning curve approximation methods.
- Surveys.
- Meta-learning learning curves. (Leiden People)
- Have not seen anyone doing something like this yet. But talk about the semi-parametric kernel ridge and its applications.
- Maybe ties to structured learning and meta-learning if it does not hinder the flow of the paper.

3 Problem Setting

To formalize our supervised learning problem setting let us start by introducing generic input and output spaces as $\mathbb{X}$ and $\mathbb{Y}$. A learning algorithm $\mathcal{A}$ takes i.i.d samples $\mathcal{D}_N := (x, y)_{i=1}^N$ (also represented by$\mathcal{A}(\mathcal{D}_N)$), from an unknown distribution $\mathcal{P}_{\mathcal{D}_N}$ over $\mathbb{X} \times \mathbb{Y}$ and produces a hypothesis h from hypothesis class $\mathbb{H}$ defined by the learner $\mathcal{A}$. Then, the prediction of a learner can be represented as $\hat{y} = h(x) \in \mathbb{Y}$. Moreover, the performance of the learner is measured by a loss function $\mathcal{L}(y, \hat{y})$. Thus, the expected loss or risk $\mathcal{R}$ of a hypothesis generated by a learning algorithm over the true distribution $\mathcal{P}_{\mathcal{D}_N}$ is given as:

$$\mathcal{R}(h) = \int \mathcal{L}(y, \hat{y}) d\mathcal{P}_{\mathcal{D}_N}. \tag{1}$$

A reliable learning curve of a learner is obtained by averaging over many $\mathcal{D}_N$ for varying N. Then, the average risk of a learner is given as:

$$\bar{\mathcal{R}}(\mathcal{A}, N) = \mathop{\mathbb{E}}_{\mathcal{D}_N} \mathcal{R}(\mathcal{A}(\mathcal{D}_N))). \tag{2}$$

An individual learning curve $\mathcal{C} : \mathbb{R} \to \mathbb{R}$ depends on $\mathcal{A}$ and $\mathcal{P}_{\mathcal{D}_N}$. This allows us to represent this as a supervised learning problem. Let us assume that predicting a learning curve is the task $\mathcal{T}_i$ that we are interested in and we are given training size and corresponding risk values from this curve $\mathcal{Z} := (N_i, \mathcal{R}_i)_{i=1}^Q$ and learning curves $(\mathcal{C}_i)_{i=1}^M$. Note that the curves are not analytical but in a discretized point-wise available. As mentioned before the cost of obtaining a learning curve increases as the N increases. Thus, we assume that $N_i \subset [0, k]$ where $k \in \mathbb{Z}^+$, but limited to be small, so that the cost of obtaining $\mathcal{R}(\mathcal{A}(\mathcal{D}_{N_i}))$ is not large. In other words, we are trying to model $\mathcal{C}$ by training a learner $\mathcal{M} : \mathbb{R} \to \mathbb{R}$ with samples $\mathcal{Z}$ and $(\mathcal{C}_i)_{i=1}^M$.

It should be noted that in most cases the true distribution of the data is unknown, thus the empirical version of the risk is obtained by creating one hold-out set and several cross-validation splits. Moreover, additional splits might be necessary if the hyper-parameters of the learner are to be tuned as well.

4 Methods

4.1 Semi-Parametric Kernel Ridge

For a learning task T_i of a one-dimensional regression problem in which the input-output relation is given by.

$$\mathcal{R} = \mathcal{C}(N) + \varepsilon, \tag{3}$$

where ε is taken to be standard normal.

Kernel Ridge Regression is obtained by utilizing the *Nonparametric Representer Theorem* [5]. This theorem assumes that g is a strictly increasing function and $\mathcal{L}$ is a monotonic loss function. (We will assume mean squared loss $\mathcal{L} := \sum_i^Q (\mathcal{M}(N_i) - \mathcal{R}_i)^2$ throughout the paper.), Then,

$$\hat{\mathcal{M}} = \min_{\mathcal{M} \in \mathbb{H}} \mathcal{L}(\mathcal{M}, \mathcal{R}) + g(||\mathcal{M}||_{\mathbb{H}}) \tag{4}$$

has the solution of the form, $\mathcal{M}(\cdot) = \sum_i^{\mathcal{Z}} \alpha_i k(\cdot, \mathcal{D}_{N_i})$. Where $\mathbb{H}$ is the Reproducing Hilbert Space and k is an arbitrary kernel. Until this point only available samples $\mathcal{Z}$ are being utilized. To incorporate the other learning curves $(\mathcal{C}_i)_{i=1}^M$ the *Semi-parametric Representer Theorem* [5] can be utilzed. This theorem assumes that the underlying function can be modeled as $\tilde{\mathcal{M}} = \mathcal{M} + h$ where $h \in span\{\psi_p\}$ and $\{\psi_p\}_{p=1}^{\mathcal{Z}}$ are real-valued functions. Then the supervised learning problem with squared loss is given by:

$$\hat{\tilde{\mathcal{M}}} = \min_{\tilde{\mathcal{M}} \in \mathbb{H}} \mathcal{L}(\tilde{\mathcal{M}}, \mathcal{R}) + g(||f||_{\mathbb{H}}). \tag{5}$$

And, it has the solution in the form of $\tilde{\mathcal{M}}(\cdot) = \sum_i^Q \alpha_i k(\cdot, N_i) + \sum_j^M \beta_j \psi_j(\cdot)$. This expression allows us to incorporate information coming from other available learning curves.

Further assuming $g(||f||_{\mathbb{H}}) := \lambda ||f||^2$, the optimal solutions for the Semiparametric Representer Theorem is given by:

$$\hat{\boldsymbol{\alpha}} = (\mathbf{K}\mathbf{K} + \lambda \mathbf{K} - \mathbf{K}\boldsymbol{\psi}(\boldsymbol{\psi}^{\mathrm{T}}\boldsymbol{\psi})^{-1}\boldsymbol{\psi}^{\mathrm{T}}\mathbf{K})^{-1}(\mathbf{K} - \mathbf{K}\boldsymbol{\psi}(\boldsymbol{\psi}^{\mathrm{T}}\boldsymbol{\psi})^{-1}\boldsymbol{\psi})\boldsymbol{\mathcal{R}} \tag{6}$$

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{\psi}^{\mathrm{T}}\boldsymbol{\psi})^{-1}(\boldsymbol{\psi}^{\mathrm{T}}\boldsymbol{\mathcal{R}} - \boldsymbol{\psi}^{\mathrm{T}}\mathbf{K}\hat{\boldsymbol{\alpha}}), \tag{7}$$

where $\boldsymbol{\alpha} \in \mathbb{R}^{Q \times 1}$, $\boldsymbol{\beta} \in \mathbb{R}^{M \times 1}$, $\boldsymbol{\psi} \in \mathbb{R}^{Q \times M}$, $\mathbf{K} \in \mathbb{R}^{Q \times Q}$ and $\boldsymbol{\mathcal{R}} \in \mathbb{R}^{Q \times 1}$. All the bold symbols represent the concatenated versions of all data points and functions. One might question the need for additional learning curve information or the benefit obtained from the samples. These questions are addressed in Sections 7.1 and 7.2 respectively. By looking at the learning curves and statistical testing of the proposed learning algorithm.

Now, one other question can be related to the selection of functions ψ_j. One obvious choice is to use all the available curves, however, with an increasing number of curves the computational complexity of the solution procedure increases. Moreover, the possible noise coming from all the learning curves might hinder the final prediction of the learning curve that is of interest. Thus, we propose Functional Principle Component Analysis (FPCA) as it allows smoothing and selecting only important modes of variation of the underlying learning curves. The reasoning behind FPCA utilization can be seen in Section 7.3.

5 Results and Discussion

5.1 Learning Monotone Learning Curves

5.1.1 Learning curve prediction for changing hyper-parameters of the same model and same dataset

- The results of this experiment I have already shown.
- It is possible to extrapolate with little error
- Needs to write the experimentation one last time in a reproducible way.

5.1.2 Learning curve prediction for tuned hyper-parameter of the same model and different datasets

- We do not have it yet but after the Section 5.1.1 experiments are written again it will be easy to obtain.
- I suspect this will be a bit more problematic as the tuned curves have a bit different variations.

5.2 Learning Non-monotone Learning Curves

- We have linear ridge regression for varying lambdas which is non-monotonic.
- This method was able to fairly accurately predict even bimodal curves. (Bimodality comes from the different dimensional datasets.
- I will take these from Tom. After he comes back he will prepare the already existing non-monotonic examples.

5.3 Performance against baselines

- Here the prediction performance for power-law etc. will be provided.
- The difference between the non-monotonic and monotonic predictions will be highlighted.

6 Conclusion

- We can accurately predict learning curves with a data-driven approach.
- Talk about the caveats: Limitations of the approach, time complexity, choices that have to be made like, how many components, FPCA limitations, and the drawbacks you might observe during the re-experimentation.
- Talk about possible extensions like Gaussian Process extension for adding uncertainty to our predictions, might get over-confidence though from my experience with multi-fidelity regression. One additional remark would be to penalize the β's. This might pave the way for using the raw learning curves in the prediction without being affected by the noisiness of it. But an additional burden on the model selection can make things a bit difficult again.

Acknowledgments

References

[1] Siyuan Ma and Mikhail Belkin. Diving into the shallows: A computational perspective on large-scale shallow learning.

[2] J. Kiefer and J. Wolfowitz. Stochastic Estimation of the Maximum of a Regression Function. 23(3):462–466.

[3] Felix Mohr, Tom J Viering, Marco Loog, and Jan N van Rijn. LCDB 1.0: An Extensive Learning Curves Database for Classification Tasks.

[4] Marco Loog and Tom J Viering. A Survey of Learning Curves with Bad Behavior: Or How More Data Need Not Lead to Better Performance. page 16.

[5] Bernhard Schölkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Adaptive Computation and Machine Learning. MIT Press.

7 Appendix

7.1 Learning Curves for Semiparameteric Kernel Ridge

With Synthetic Data.

- Show the increased performance compared to standard Kernel Ridge.
- Small training set size benefit.
- Extrapolation capabilities.

7.2 Hypothesis Testing for the Semiparametric Kernel Ridge

With Synthetic Data.

- Only available test for know seems to be Cramer Von Mises statistical test. The reason is we do not know the underlying distribution between resulting risks of the different hypotheses. Moreover, the variance is not the same from the initial experimentation which eliminates almost all the other statistical testing methods.
- Here if I have time, might add the work that Rickard and I talked about.

7.3 FPCA for information extraction from the available learning curves

With Real Data.

- Explain and show the time benefit.
- Explain and show the smoothing benefit.
- How to decide on how many components to use.
- Mention mean adjustment as it is found to be critical in utilization.