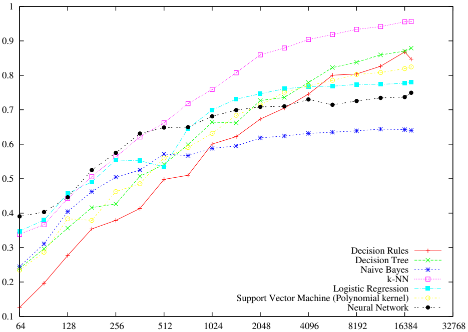
Learning Curves for Decision Making in Supervised Machine Learning

Felix Mohr, Jan N. van Rijn

Universidad de La Sabana - Colombia
Leiden University - the Netherlands

Motivation

All of us know curves like these ...



But how to systematically integrate these into decision making for data acquisition or improving efficiency and quality in model induction?

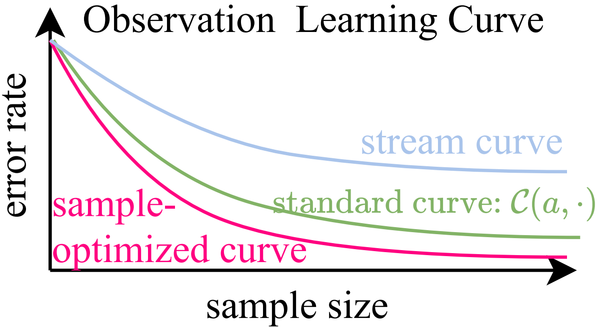
Universidad de La Sabana
Universiteit Leiden

Overview

This talk is based on our recent survey paper [Mohr and van Rijn, 2022]: "Learning Curves for Decision Making in Supervised Machine Learning".

Overview

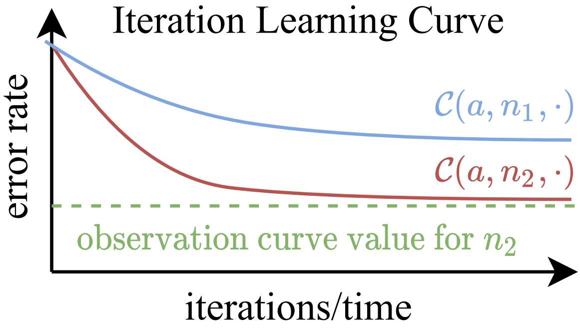
Observation Learning Curves



$$\mathcal{C}(a, s) = \mathbb{E}_{|d_{tr}|=s}\left[\text{out-of-sample performance of } a(d_{tr})\right],$$

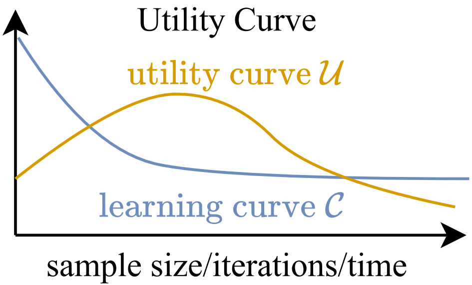
where a is an algorithm that learns a model $a(d_{tr})$ from some data d_{tr}.

Iteration Learning Curves

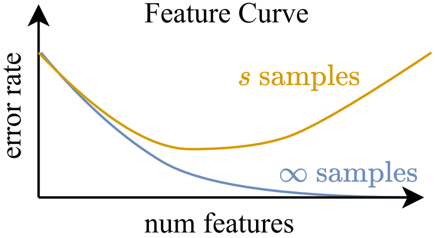


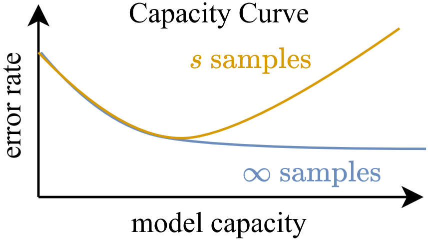
$$\mathcal{C}(a, n, s) = \mathbb{E}_{|d_{tr}|=n}\left[\text{OOS score of } a(d_{tr}) \text{ after } s \text{ iterations}\right]$$

Utility Curves



Other Performance Curves





Empirical Learning Curves

Learning curves are unknown: The OOS cannot be computed (and much less its expected value across different setups).

Remedy as usual: Estimate $\mathcal{C}(a, s)$ or $\mathcal{C}(a, n, s)$ on a concrete set of validation data points.

We refer to the considered values for s as anchors.

Empirical learning curves can be expensive to compute, and there have been papers solely on the analysis of empirical learning curves [Perlich et al., 2003].

Empirical Learning Curves

Recommended Resources

LCDB (github.com/fmohr/lcdb) provides API access to

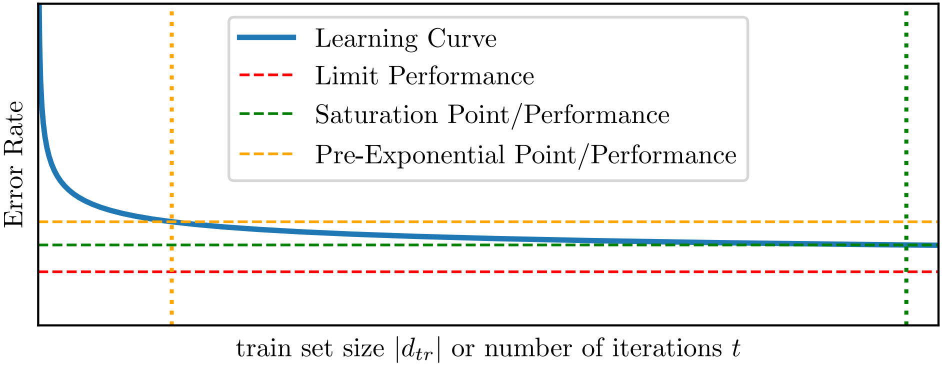
- accuracy/error/F1/log-loss/AUROC learning curves for
- 40 learners (default hyperparameters) on 240 datasets with
- at least 10 train/validation/test folds

LCDB is the largest and most flexible database for observation learning curves.

LCBench (github.com/automl/LCBench) provides API access to

- (balanced) accuracy of 50 episode iteration learning curves
- on a train/validation/test folds for
- 2000 NN architectures on 35 datasets

Terminology



Well Behaved Learning Curves

Ideally, learning curves had some nice properties such as

- monotonicity (improvements cannot get lost)
- convexity (improvements occur at a systematic rate)

Are learning curves well behaved in this sense?

- Yes, mostly! (LCDB)
- No (empirical evidence on the nasty Double Descent, divergence, ...).

Depends on the type of learning/performance curve and the learner.

This is *one* reason why making the distinction is important when fitting curve models ...

Modelling a Learning Curve

Objective: Derive a model of the *true* learning curve based on the *empirical* learning curve.

The empirical learning curve is the result of sampling from a *stochastic process* that underlies *heteroscedastic noise* σ_s^2 stemming from randomness in data splits and the learning algorithm itself.

It is typically assumed that this stochastic process follows the distribution

$$f(s) \sim \mathcal{N}(\mu_s, \sigma_s^2) = \mu_s + \mathcal{N}(0, \sigma_s^2),$$

where μ_s is the (true) average generalization performance of the learner at anchor s.

Modelling a Learning Curve

Point-Wise Models

In the simplest case, one just estimates the mean μ_s of the curve.

The noise σ_s^2 is just ignored.

One of the most commonly used model classes is the Inverse Power Law (IPL):

$$\hat{\mu}_s = \alpha + \beta s^{-\gamma},$$

where $\alpha, \beta, \gamma > 0$ (for descending curves, e.g., error rates).

However, a dozen of model classes have been proposed (Vierig and Loog 2022).

Modelling a Learning Curve

The Inference Problem

For any parametric model, we will have parameters $\beta_1, .., \beta_m$ to describe the behavior of μ_s. The center of attention is the likelihood

$$\mathbb{P}(\beta_1, .., \beta_m \mid D),$$

where $D = \{(s_1, y_1), .., (s_n, y_n)\}$ is the set of observations of the learning curve.

For a point-wise model, we ask for the $\arg\max$ of this expression, i.e., the MLE for the parameters to obtain the *most likely* values for $\beta_1, .., \beta_m$.

This assignment of $\beta_1, .., \beta_m$ can be computed rather efficiently with non-linear regression methods such as the LM algorithm.

Modelling a Learning Curve

What is the best model fit?

Even for a single model class, the best parameters depend on the objective.

- if the goal is to *explain* a learning curve on an observed range, one can apply standard regression.
- if the goal is to *extrapolate* a learning curve, the parameters obtained from standard regression are typically sub-optimal (since they give too much weight on initial parts of the curve).

It has been (empirically) shown that, on the same datasets, a model can be optimal w.r.t. the first question but sub-optimal w.r.t. the second one.

We are not aware of extrapolation approaches that explicitly consider this issue.

Modelling a Learning Curve

Modeling Uncertainty

The point-wise estimate does not consider any type of uncertainty.

We can be uncertain about (at least) three things:

1. the gap between the predicted performance $\hat{f}(s)$ and the true value μ_s at some anchor s,
2. whether the current estimates of θ are the best we can get *within* our fixed model class, and
3. uncertainty about whether or not the model class itself is appropriate.

Research papers often do not specify what type of uncertainty they look at; this becomes only clear from the context.

Modelling a Learning Curve

Aleatoric vs Epistemic Uncertainty

It is sensible to make the now common distinction between aleatoric and epistemic uncertainty.

- The aleatoric (problem-inherent) uncertainty is σ_s^2. An estimate comes for free in CV but is missing in simple hold-out methods.
- The epistemic (sample-based) uncertainty depends on the number N of observations from which $\beta_1, .., \beta_m$ are estimated.

The epistemic uncertainty can be reduced by gathering more observations.

This is similar to GPs in which more data leads to reduced uncertainty in the kernel-neighborhood of the observations.

Modelling a Learning Curve

Range Estimates

Range estimates define lower and upper bounds for the estimate of μ_s.

... not necessarily (and usually not) the inf or sup of μ_s but simply express any type of interval considered to meaningfully express uncertainty.

... often used to quantify the aleatoric uncertainty, e.g., by estimating μ_s as above and adding confidence bands at each anchor s (even the known ones).

Modelling a Learning Curve

Distribution Estimates

The parameters $\beta_1, .., \beta_m$ describe the behavior of μ_s, so the likelihood

$$\mathbb{P}(\beta_1, .., \beta_m \mid D)$$

quantifies the epistemic uncertainty about μ_s.

Thanks to Bayes we have that

$$\mathbb{P}(\beta_1, .., \beta_m \mid D) \propto \mathbb{P}(D \mid \beta_1, .., \beta_m)\mathbb{P}(\beta_1, .., \beta_m),$$

which is intractable but can be sampled from, e.g., via MCMC.

The aleatoric uncertainty σ_s^2 is not considered here. However, an estimate $\hat{\sigma}_s^2$ could be *used* if D had non-aggregated observations at anchors.

Modelling a Learning Curve

Distribution Estimates

By sampling S samples from the posterior $\mathbb{P}(\beta_1, .., \beta_m \mid D)$, one can collect predictions $z_1, .., z_S = f_{(\beta_1,..,\beta_m)}(s)$ for any anchor s.

This yields

$$\hat{\mu}_{\mathbb{P},s} = \frac{1}{S}\sum_{i=1}^{S} z_i \quad \text{and, from this, the variance} \quad \hat{\sigma}^2_{\mathbb{P},s} = \frac{1}{S}\sum_{i=1}^{S}(z_i - \hat{\mu}_s)^2$$

The subscript $\mathbb{P}$ emphasizes that this is an estimate of the variance of $\mathbb{P}$ and not of the aleatoric uncertainty σ_s^2.

If the *epistemic* uncertainty is considered a Gaussian, then this yields a full characterization of $\mathbb{P}$.

Overview

Decision Situations

There are at least three situations in which learning curves aid decision making:

1. Data Acquisition: The acquisition of how many additional labels is (economically) reasonable?
2. Early Stopping: Stop model training as soon as limit/saturation performance is reached.
3. Early Discarding: Stop model training as soon as it can be recognized that the limit/saturation performance will not be at least τ.

Learning Curves for Data Acquisition

For a single learner, learning curves *based on training sizes* give insights into the possible limit performance.

Max/min-aggregating this over a portfolio of (high-variance) learners gives insights into the intrinsic noise of the data.

If this capacity curve has plateaud, then the noise level has been reached, and additional instances will not help improve performance.

Otherwise, we can try to predict the (utility) saturation point to plan the acquisition of new labels.

Learning Curves for Early Stopping

Early Stopping means interrupting the training process of a learner if the (observation or iteration) learning curve has converged.

There is no notion of a baseline here.

"Early" since without this interruption training might have continued because

- more data is available, or
- other stopping criteria are not yet satisfied.

Examples:

- iteratively increase training sizes until no improvement occurs
- use validation data to detect a stall learning process

Learning Curves for Early Discarding

Early Discarding means interrupting the training process of a learner if it will not improve upon some baseline τ.

τ is typically the currently best solution during model selection.

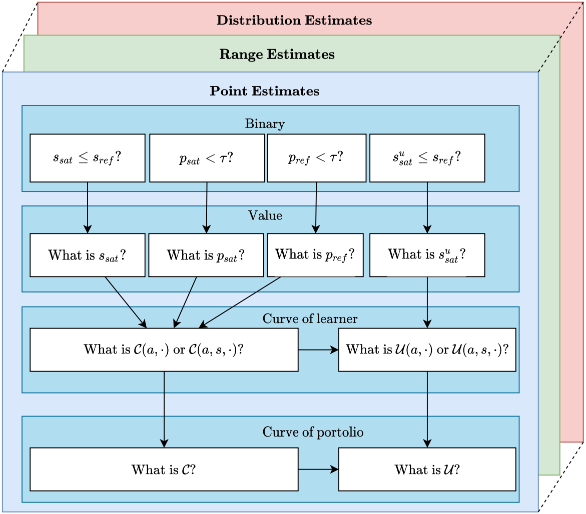
What is coloquially meant by "best solution" is the *learner* that is best when training a model on a given *dataset size*, typically between 70% and 90% of the available data.

Learning Curves for Early Discarding

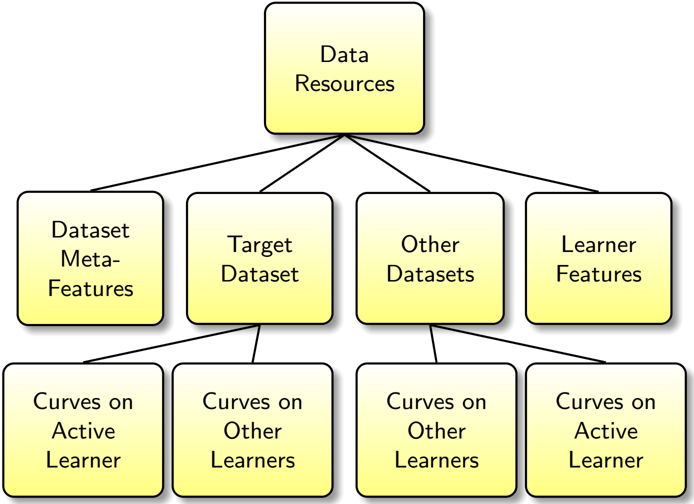
An important sub-classification of decision problems is how fidelity can change during the model selection process:

- Horizontal Model-Selection: The (finite) set of available learners is known in advance, and their learning curves are grown simultaneously (Successive Halving).
- Vertical Model-Selection: Learning curves of a *stream* of learners are grown one by one (LCCV).
- Diagonal Model-Selection: The perspectives can mix (Hyperband, Freeze-Thaw BO, Fabolas).

Questions to ask about Learning Curves



Resources Available for Decision Making



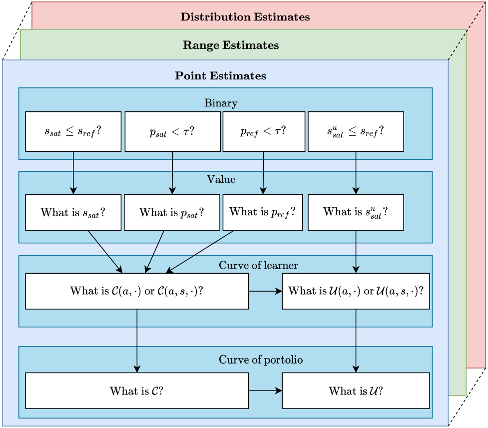
Overview

Background on Learning Curves

Learning Curves for Decision Making

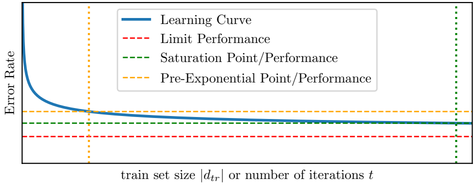
Literature Review

Overview of Literature



- Structured based on the complexity of the solution
- Various works use a complex model to answer a simple question (which is not wrong)
- Typically have the potential to answer more complex questions
- Goal: give an overview of what has already been done throughout the years
- Disclaimer: Will be a time journey, some core ideas come from old papers but form the basis of current works
- Aim at works that explicitly utilize learning curves

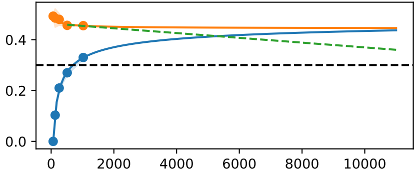
Identification of Saturation Performance p_{sat}



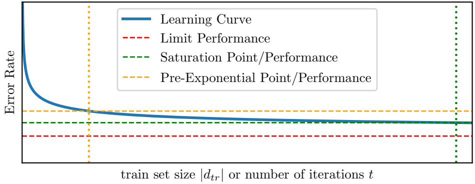
- Observation curves: What is the best performance a given learner will obtain, regardless of the amount of data
- Use case: Determine the effect of more data (data acquisition)
- Iteration curves: What is the best performance a given learner will obtain, regardless of the number of iterations (e.g., epochs)?
- Use case: discard the learner if $p_{sat} < \tau$ (early discarding)

Identification of Saturation Performance p_{sat}

- Well-studied from a theoretical viewpoint
- Insight: symmetrical behaviour between the training error and the validation error
- Cortes et al. [1994] estimated the saturation performance by averaging the train performance and test performance, once the train performance drops



Identification of Saturation Point s_{sat}



- Observation curves: Minimal amount of data that obtains p_{sat}
- Iteration curves: minimal amount of iterations (e.g., epochs) that obtain p_{sat}
- Use cases: early stopping, data acquisition
- Retrospective approaches vs. projective approaches

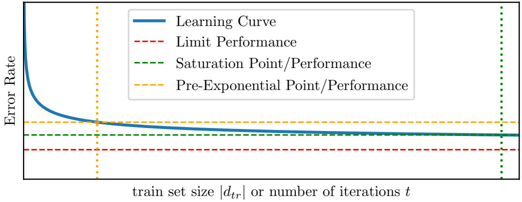
Identification of Saturation Point – retrospective approaches

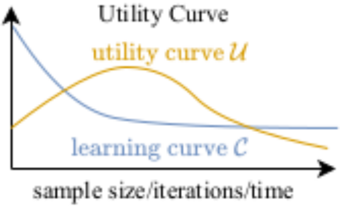
- Iteration-based curves vs observation-based curves
- John and Langley [1996] propose the Probably Close Enough (PCE) measure, based on probability and distance from the full dataset: $P(acc(N) - acc(N_i)) < \epsilon$
- Provost et al. [1999] are the first to incorporate a geometric schedule b^k (e.g., $64, 128, 256, \ldots$) as opposed to an arithmetic schedule
- Additionally, they propose a dynamic programming approach to calculate the optimal sampling strategy
- Linear regression with local sampling (LRLS)
- Finally, they formally prove that the geometric schedule is asymptotical optimal
- No meta-data was used

Identification of Saturation Point – projective approaches

- Goal: try to determine saturation point before actually running on that anchor
- Make use of meta-data (learning curve performance data of a given algorithm/hyperparameter configuration on previous datasets)
- Leite and Brazdil [2004] utilize a geometric schedule $(91, 128, 181, 256, \ldots)$, and run the algorithm on early anchors
- Based on these early anchors, they utilize a k-NN algorithm to identify the most closely related datasets and determine the saturation point on these
- Leite and Brazdil [2004] experiment with various measures to aggregate the saturation point from the related datasets

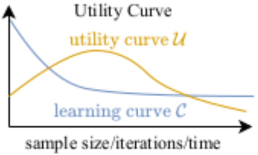
Identification of Utility-based stopping point





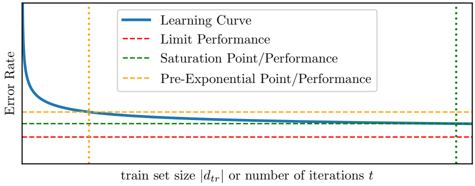
- Optimize utility of a certain cost concerning a certain model performance
- Use cases in data acquisition (cost of labelling) and early stopping (CPU cost)

Identification of Utility-based stopping point



- Very similar to progressive sampling by Provost et al. [1999]
- Stop sampling once the utility degrades
- Main complication: unifying scale for model performance and cost (training cost or acquisition cost)
- Weiss and Tian [2008] define an explicit notion of utility, where the user has to determine the cost of acquiring labels and the cost of miss-classification

Performance Bounding at Fixed Point(s)



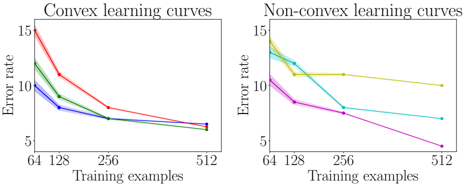
- Optimize utility of a certain cost concerning a certain model performance
- Use case: early discarding
- See also: Successive Halving [Jamieson and Talwalkar, 2016], Hyperband [Li et al., 2017], but: no learning curves
- Easier problem than performance prediction (regression) ...
- ... but higher expectations for the correctness of a statement

Data Allocation using Upper Bounds

- [Sabharwal et al., 2016] aim to answer the question, given a set of learners, which one should be evaluated next
- For each learner, the learning curve across the last two anchors is extrapolated linearly (using the most optimistic slope that is permitted by the sampling uncertainty at each anchor)
- The learner that is projected to be the best at the final anchor will be allocated the double amount of data
- The method is repeated until one algorithm reaches the final anchor
- Disadvantages: designed to work with a fixed set of algorithms (horizontal algorithm configuration)

Learning-curve based cross-validation

- LCCV aims to be a learning-curve-based version of cross-validation for algorithm configuration (vertical algorithm configuration)
- Once it has determined an ‘incumbent’ (on full data), it will construct learning curves for new configurations
- Build upon the assumption that learning curves are convex [Mohr and van Rijn, 2021]

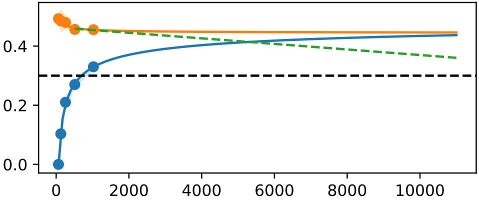


Learning-curve based cross-validation

Similar to DAUB, it will extrapolate a learning curve using the most optimistic extrapolation [Mohr and van Rijn, 2021]

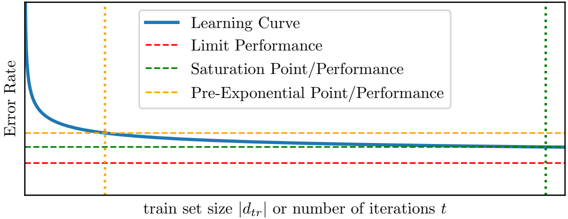
Early discarding is based on two criteria:

- Optimistic extrapolation does not yield improvement over incumbent
- Train error exceeds validation error of incumbent



More conservative than successive halving

Performance Prediction at Fixed Point(s)

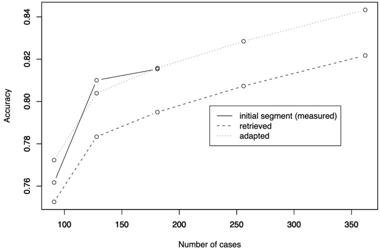


- At its core a regression problem
- Note that: Predicting the saturation performance is a special case of performance prediction at fixed points
- Various types of meta-data: implicit dataset features, explicit dataset features and algorithm features

Implicit or Explicit Dataset Context

Very similar to the work of Leite and Brazdil [2004] predicting the saturation point, Leite and Brazdil [2005] aim to predict the performance at a given point utilizing a dataset with learning curves

A learning-curve-based distance measure is utilized to select the *k* most similar datasets



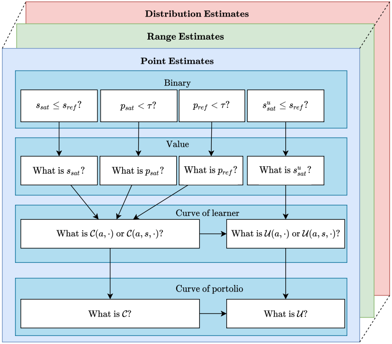
Learning curves can be quite different, and a measure was developed to scale the curves of the current dataset to other datasets

This work was extended by Leite and Brazdil [2010] to use explicit meta-features

Generalization With an Explicit Algorithm Context

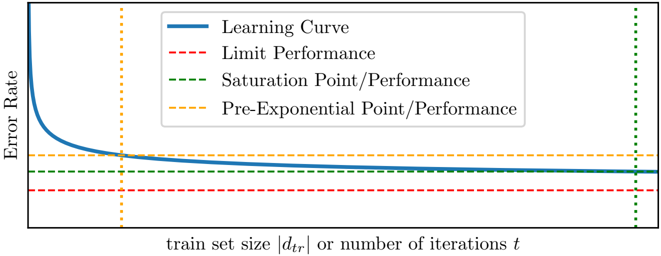
- Baker et al. [2018] utilize a learning curve model to predict for a given configuration whether it will be competitive with the best found configuration so far
- Their model is specialized in neural networks and includes beyond learning curve data also (simplistic) data of the configuration (network width, network depth)
- Long et al. [2020] build upon this work and extend it with additional n-gram features. They report a better Spearman correlation score

Quick recap ...



Next, we will see methods that model the entire learning curve

Performance Prediction at Any Point



- Aims to model the entire learning curve
- Often used to do performance prediction at a given point
- Recall the various learning curve models (e.g., inverse power law)
- Distinguishing factor: how to deal with uncertainty

Point estimates

- Various parametric-models can be used for learning curve modelling
- (to the best of our knowledge) Cortes et al. [1993] were the first to use the inverse power law for modelling learning curves
- John and Langley [1996] introduce the notion of 'Probably Close Enough', but also work with point estimates

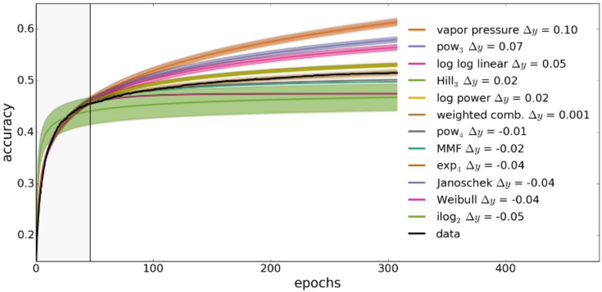
Range Estimates

- Mukherjee et al. [2003] model learning curves for 8 DNA datasets, explicitly modelling a 25 and 75-percentile curve based on (MC)CV-folds
- Koshute et al. [2021] use the inverse power law to predict the minimum anchor point on which a learner must be trained to reach near-saturation performance
- They do this by fitting a learning curve model on the lower confidence bounds of confidence intervals of known anchors
- This model is used to determine the anchor where the performance is within an ϵ-distance from the saturation performance
- DAUB [Sabharwal et al., 2016] and LCCV [Mohr and van Rijn, 2021] use a similar strategy to utilize range estimates in the model

Estimate Distributions

- Domhan et al. [2015] take into account uncertainty about the model itself
- The approach assumes learning curves to be instances of a parametric model that is a linear combination of known model classes, such as the inverse power law, and others
- Monte-Carlo Markov Chains are used to estimate the posterior distribution
- This was used for early discarding of configurations that would not be competitive
- Use a probability to determine whether with a certain probability a configuration can be pruned

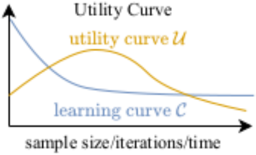
Learning Curve models



Reference name	Formula
vapor pressure	$\exp(a + \frac{b}{x} + c\log(x))$
pow_3	$c - ax^{-\alpha}$
log log linear	$\log(a\log(x) + b)$
Hill_3	$\frac{y_{\max} x^{\eta}}{\kappa^{\eta} + x^{\eta}}$
log power	$\frac{a}{1+\left(\frac{x}{e^{b}}\right)^{c}}$
pow_4	$c - (ax + b)^{-\alpha}$
MMF	$\alpha - \frac{\alpha-\beta}{1+(\kappa x)^{\delta}}$
exp_4	$c - e^{-ax^{\alpha}+b}$
Janoschek	$\alpha - (\alpha - \beta)e^{-\kappa x^{\delta}}$
Weibull	$\alpha - (\alpha - \beta)e^{-(\kappa x)^{\delta}}$
ilog_2	$c - \frac{a}{\log x}$

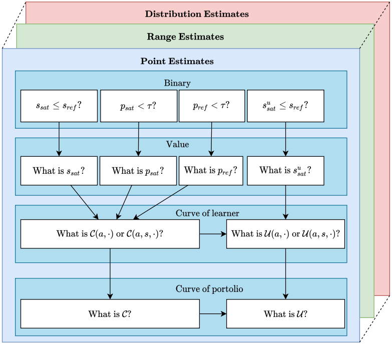
Curve movels used by and figure by Domhan et al. [2015]

Utility Prediction at Any Point



- Last [2007] utilizes the inverse power law to model the performance component in the utility curves
- When the data acquisition costs are known, this approach allows us to projectively calculate the optimal dataset size
- This approach is used by Sarkar et al. [2015] for automated software configuration
- every instance is a parametrization of a software library, and obtaining its label requires the costly execution of a benchmark on such a configuration
- The goal is to understand how many observations need to be acquired to be able to learn a reliable prediction model

Performance at Any Point for Any Learner



- Assumption: by modelling learning curves across learners, the models per learning curve might improve
- Swersky et al. [2014] proposes Freeze-Thaw-(Bayesian) Optimization, using a Gaussian Process to model the asymptotic performance with an exponentially decaying kernel (iteration learning curves)
- Klein et al. [2017a] proposes FABOLOS (observation learning curves), using Gaussian Process to model full learning curves
- Klein et al. [2017b] utilize Bayesian Neural Networks (having d+1 inputs), modelling both the performance and the uncertainty

Freeze-Thaw-(Bayesian) Optimization

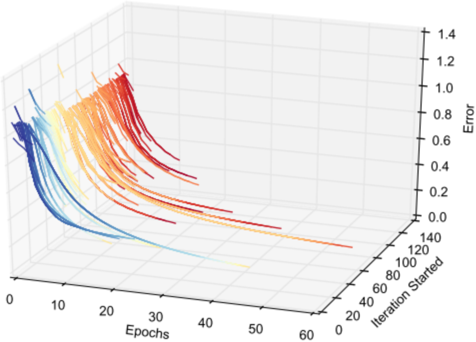


Figure taken from Swersky et al. [2014]

Bayesian Neural Network for Learning Curves

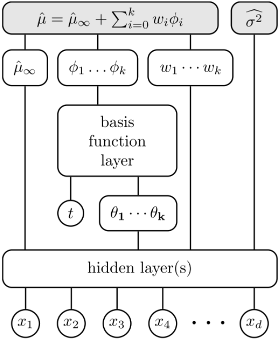


Figure taken from Klein et al. [2017b]

Outlook

Question	Type	LC other DS	DS MF	LC other AL	AL MF	Utility	Estimate Type
p_{sat}	obs.	✗	✗	✗	✗	✗	p
s_{sat}	obs.	✗	✗	✗	✗	✗	r
s_{sat}	obs.	✗	✗	✗	✗	✗	p
s_{sat}	obs.	✓	✗	✗	✗	✗	p
s_{sat}	iter.	✗	✗	✗	✗	✗	p
s_{sat}^{u}	obs.	✗	✗	✗	✗	✓	p
$\mathcal{C}(a, \|d_{tr}\|)$	obs.	✗	✗	✗	✗	✗	p
$\overline{\mathcal{C}(a, \|d_{tr}\|)}$	both	✗	✗	✗	✗	✗	p
$\overline{\mathcal{C}(a, \|d_{tr}\|)}$	obs.	✗	✗	✗	✗	✗	r
$\overline{\mathcal{C}(a, \|d_{tr}\|)}$	obs.	✓	✗	✗	✗	✗	r
$\mathcal{C}(a, \|d_{tr}\|)$	obs.	✓	✗	✗	✗	✗	p
$\mathcal{C}(a, \|d_{tr}\|)$	iter.	✗	✗	✓	✓	✗	p
$\mathcal{C}(a, \|d_{tr}\|)$	obs.	✓	✓	✗	✗	✗	p
$\mathcal{C}(a, \cdot)$	obs.	✗	✗	✗	✗	✗	r
$\mathcal{C}(a, \cdot)$	obs.	✗	✗	✗	✗	✗	p
$\mathcal{C}(a, \cdot)$	iter.	✗	✗	✗	✗	✗	p
$\mathcal{C}(a, \cdot)$	iter.	✗	✗	✗	✗	✗	d
$u(\mathcal{C}(a, \cdot))$	obs.	✗	✗	✗	✗	✓	p
$\mathcal{C}(\cdot, \cdot)$	obs.	✗	✗	✓	✓	✗	d
$\mathcal{C}(\cdot, \cdot)$	both	✗	✗	✓	✓	✗	d

Outlook

Question	Type	LC other DS	DS MF	LC other AL	AL MF	Utility	Estimate Type
p_{sat}	obs.	✗	✗	✗	✗	✗	p
s_{sat}	obs.	✗	✗	✗	✗	✗	r
s_{sat}	obs.	✗	✗	✗	✗	✗	p
s_{sat}	obs.	✓	✗	✗	✗	✗	p
s_{sat}	iter.	✗	✗	✗	✗	✗	p
s_{sat}^{u}	obs.	✗	✗	✗	✗	✓	p
$\mathcal{C}(a, \|d_{tr}\|)$	obs.	✗	✗	✗	✗	✗	p
$\overline{\mathcal{C}(a, \|d_{tr}\|)}$	both	✗	✗	✗	✗	✗	p
$\overline{\mathcal{C}(a, \|d_{tr}\|)}$	obs.	✗	✗	✗	✗	✗	r
$\overline{\mathcal{C}(a, \|d_{tr}\|)}$	obs.	✓	✗	✗	✗	✗	r
$\mathcal{C}(a, \|d_{tr}\|)$	obs.	✓	✗	✗	✗	✗	p
$\mathcal{C}(a, \|d_{tr}\|)$	iter.	✗	✗	✓	✓	✗	p
$\mathcal{C}(a, \|d_{tr}\|)$	obs.	✓	✓	✗	✗	✗	p
$\mathcal{C}(a, \cdot)$	obs.	✗	✗	✗	✗	✗	r
$\mathcal{C}(a, \cdot)$	obs.	✗	✗	✗	✗	✗	p
$\mathcal{C}(a, \cdot)$	iter.	✗	✗	✗	✗	✗	p
$\mathcal{C}(a, \cdot)$	iter.	✗	✗	✗	✗	✗	d
$u(\mathcal{C}(a, \cdot))$	obs.	✗	✗	✗	✗	✓	p
$\mathcal{C}(\cdot, \cdot)$	obs.	✗	✗	✓	✓	✗	d
$\mathcal{C}(\cdot, \cdot)$	both	✗	✗	✓	✓	✗	d

Summary

- Formal definitions about learning curves
- Three decision situations: data acquisition, early stopping and early discarding
- Framework for various questions to be answered by learning curves
- Various concepts from early research are still commonly used in modern papers

Outlook

Question	Type	LC other DS	DS MF	LC other AL	AL MF	Utility	Estimate Type
p_{sat}	obs.	✗	✗	✗	✗	✗	p
s_{sat}	obs.	✗	✗	✗	✗	✗	r
s_{sat}	obs.	✗	✗	✗	✗	✗	p
s_{sat}	obs.	✓	✗	✗	✗	✗	p
s_{sat}	iter.	✗	✗	✗	✗	✗	p
s^u_{sat}	obs.	✗	✗	✗	✗	✓	p
$\mathcal{C}(a, \lvert d_{tr}\rvert)$	obs.	✗	✗	✗	✗	✗	p
$\overline{\mathcal{C}(a, \lvert d_{tr}\rvert)}$	both	✗	✗	✗	✗	✗	p
$\overline{\mathcal{C}(a, \lvert d_{tr}\rvert)}$	obs.	✗	✗	✗	✗	✗	r
$\overline{\mathcal{C}(a, \lvert d_{tr}\rvert)}$	obs.	✓	✗	✗	✗	✗	r
$\mathcal{C}(a, \lvert d_{tr}\rvert)$	obs.	✓	✗	✗	✗	✗	p
$\mathcal{C}(a, \lvert d_{tr}\rvert)$	iter.	✗	✗	✓	✓	✗	p
$\mathcal{C}(a, \lvert d_{tr}\rvert)$	obs.	✓	✓	✗	✗	✗	p
$\mathcal{C}(a, \cdot)$	obs.	✗	✗	✗	✗	✗	r
$\mathcal{C}(a, \cdot)$	obs.	✗	✗	✗	✗	✗	p
$\mathcal{C}(a, \cdot)$	iter.	✗	✗	✗	✗	✗	p
$\mathcal{C}(a, \cdot)$	iter.	✗	✗	✗	✗	✗	d
$u(\mathcal{C}(a, \cdot))$	obs.	✗	✗	✗	✗	✓	p
$\mathcal{C}(\cdot, \cdot)$	obs.	✗	✗	✓	✓	✗	d
$\mathcal{C}(\cdot, \cdot)$	both	✗	✗	✓	✓	✗	d

Summary

- Formal definitions about learning curves
- Three decision situations: data acquisition, early stopping and early discarding
- Framework for various questions to be answered by learning curves
- Various concepts from early research are still commonly used in modern papers

Outlook

- Call for universal benchmark to better compare methods
- Many papers answer a question that is harder than the situation
- None of the papers apply the full potential of meta-data yet

Outlook

Question	Type	LC other DS	DS MF	LC other AL	AL MF	Utility	Estimate Type
p_{sat}	obs.	✗	✗	✗	✗	✗	p
s_{sat}	obs.	✗	✗	✗	✗	✗	r
s_{sat}	obs.	✗	✗	✗	✗	✗	p
s_{sat}	obs.	✓	✗	✗	✗	✗	p
s_{sat}	iter.	✗	✗	✗	✗	✗	p
s_{sat}^{u}	obs.	✗	✗	✗	✗	✓	p
$\mathcal{C}(a, \lvert d_{tr} \rvert)$	obs.	✗	✗	✗	✗	✗	p
$\overline{\mathcal{C}(a, \lvert d_{tr} \rvert)}$	both	✗	✗	✗	✗	✗	p
$\overline{\mathcal{C}(a, \lvert d_{tr} \rvert)}$	obs.	✗	✗	✗	✗	✗	r
$\overline{\mathcal{C}(a, \lvert d_{tr} \rvert)}$	obs.	✓	✗	✗	✗	✗	r
$\mathcal{C}(a, \lvert d_{tr} \rvert)$	obs.	✓	✗	✗	✗	✗	p
$\mathcal{C}(a, \lvert d_{tr} \rvert)$	iter.	✗	✗	✓	✓	✗	p
$\mathcal{C}(a, \lvert d_{tr} \rvert)$	obs.	✓	✓	✗	✗	✗	p
$\mathcal{C}(a, \cdot)$	obs.	✗	✗	✗	✗	✗	r
$\mathcal{C}(a, \cdot)$	obs.	✗	✗	✗	✗	✗	p
$\mathcal{C}(a, \cdot)$	iter.	✗	✗	✗	✗	✗	p
$\mathcal{C}(a, \cdot)$	iter.	✗	✗	✗	✗	✗	d
$u(\mathcal{C}(a, \cdot))$	obs.	✗	✗	✗	✗	✓	p
$\mathcal{C}(\cdot, \cdot)$	obs.	✗	✗	✓	✓	✗	d
$\mathcal{C}(\cdot, \cdot)$	both	✗	✗	✓	✓	✗	d

Summary

- Formal definitions about learning curves
- Three decision situations: data acquisition, early stopping and early discarding
- Framework for various questions to be answered by learning curves
- Various concepts from early research are still commonly used in modern papers

Outlook

- Call for universal benchmark to better compare methods
- Many papers answer a question that is harder than the situation
- None of the papers apply the full potential of meta-data yet

Get involved

- Learning Curves for Decision Making in Supervised Machine Learning – A Survey [Mohr and van Rijn, 2022] (under review, Arxiv)
- Try out LCDB (`pip install lcdb`) / LCBench

References

B. Baker, O. Gupta, R. Raskar, and N. Naik. Accelerating neural architecture search using performance prediction. In *6th International Conference on Learning Representations, ICLR'18*, 2018.

C. Cortes, L. D. Jackel, S. A. Solla, V. Vapnik, and J. S. Denker. Learning curves: Asymptotic values and rate of convergence. In *Advances in Neural Information Processing Systems 6*, pages 327–334. Morgan Kaufmann, 1993.

C. Cortes, L. D. Jackel, and W. Chiang. Limits in learning machine accuracy imposed by data quality. In *Advances in Neural Information Processing Systems 7*, pages 239–246. MIT Press, 1994.

T. Domhan, J. T. Springenberg, and F. Hutter. Speeding up automatic hyperparameter optimization of deep neural networks by extrapolation of learning curves. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015*, pages 3460–3468. AAAI Press, 2015.

K. G. Jamieson and A. Talwalkar. Non-stochastic best arm identification and hyperparameter optimization. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, AISTATS 2016*, volume 51 of *JMLR Workshop and Conference Proceedings*, pages 240–248. JMLR.org, 2016.

G. H. John and P. Langley. Static versus dynamic sampling for data mining. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*, pages 367–370. AAAI Press, 1996.

A. Klein, S. Falkner, S. Bartels, P. Hennig, and F. Hutter. Fast bayesian optimization of machine learning hyperparameters on large datasets. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017*, volume 54 of *Proceedings of Machine Learning Research*, pages 528–536. PMLR, 2017a.

A. Klein, S. Falkner, J. T. Springenberg, and F. Hutter. Learning curve prediction with bayesian neural networks. In *5th International Conference on Learning Representations, ICLR'17*, 2017b.

P. Koshute, J. Zook, and I. McCulloh. Recommending training set sizes for classification. *arXiv preprint arXiv:2102.09382*, 2021.

M. Last. Predicting and optimizing classifier utility with the power law. In *Workshops Proceedings of the 7th IEEE International Conference on Data Mining (ICDM 2007)*, pages 219–224. IEEE Computer Society, 2007.

R. Leite and P. Brazdil. Improving progressive sampling via meta-learning on learning curves. In *Machine Learning: ECML 2004, 15th European Conference on Machine Learning*, volume 3201 of *Lecture Notes in Computer Science*, pages 250–261. Springer, 2004.

R. Leite and P. Brazdil. Predicting relative performance of classifiers from samples. In *Machine Learning, Proceedings of the Twenty-Second International Conference (ICML 2005)*, volume 119 of *ACM International Conference Proceeding Series*, pages 497–503. ACM, 2005.

R. Leite and P. Brazdil. Active testing strategy to predict the best classification algorithm via sampling and metalearning. In *ECAI 2010 - 19th European Conference on Artificial Intelligence*, volume 215 of *Frontiers in Artificial Intelligence and Applications*, pages 309–314. IOS Press, 2010.

L. Li, K. G. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar. Hyperband: A novel bandit-based approach to hyperparameter optimization. *Journal of Machine Learning Research*, 18:185:1–185:52, 2017.

D. Long, S. Zhang, and Y. Zhang. Performance prediction based on neural architecture features. *Cognitive Computation and Systems*, 2(2):80–83, 2020.

F. Mohr and J. N. van Rijn. Fast and informative model selection using learning curve cross-validation. *arXiv preprint arXiv:2111.13914*, 2021.

F. Mohr and J. N. van Rijn. Learning curves for decision making in supervised machine learning - A survey. *CoRR*, abs/2201.12150, 2022.

S. Mukherjee, P. Tamayo, S. Rogers, R. M. Rifkin, A. Engle, C. Campbell, T. R. Golub, and J. P. Mesirov. Estimating dataset size requirements for classifying DNA microarray data. *Journal of Computational Biology*, 10(2):119–142, 2003.

C. Perlich, F. J. Provost, and J. S. Simonoff. Tree induction vs. logistic regression: A learning-curve analysis. *Journal of Machine Learning Research*, 4:211–255, 2003.

F. J. Provost, D. D. Jensen, and T. Oates. Efficient progressive sampling. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 23–32. ACM, 1999.

A. Sabharwal, H. Samulowitz, and G. Tesauro. Selecting near-optimal learners via incremental data allocation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.

A. Sarkar, J. Guo, N. Siegmund, S. Apel, and K. Czarnecki. Cost-efficient sampling for performance prediction of configurable systems (T). In *30th IEEE/ACM International Conference on Automated Software Engineering, ASE 2015*, pages 342–352. IEEE Computer Society, 2015.

K. Swersky, J. Snoek, and R. P. Adams. Freeze-thaw bayesian optimization. *CoRR*, abs/1406.3896, 2014.

G. M. Weiss and Y. Tian. Maximizing classifier utility when there are data acquisition and modeling costs. *Data Mining and Knowledge Discovery*, 17(2):253–282, 2008.