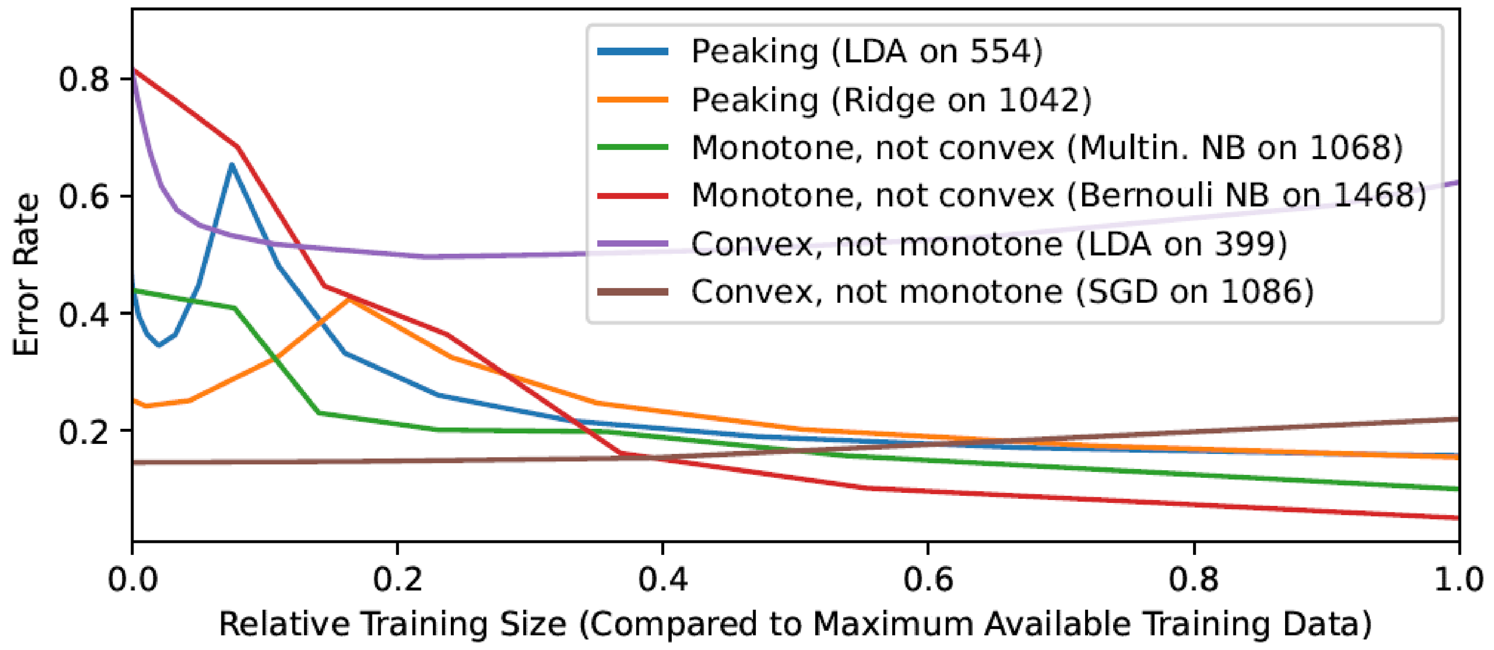


LCDB 1.0: An Extensive Learning Curves Database for Classification Tasks

Felix Mohr*, Tom J Viering**, Marco Loog**, Jan N van Rijn‡

* Universidad de La Sabana

** TU Delft

‡ Universiteit Leiden

1 What are Learning Curves?

Plots **generalization error** C versus **training set size** s
Use them to:

1. Estimate the value of collecting more training. By extrapolating a learning curve.

2. Speed up training. If the curve doesn't improve with s anymore, stop adding more data to save time.

3. Faster model selection. By extrapolating the curve of learners, we can rule out bad learners early [1].

2 Why study Learning Curves?

Learning curves can have **surprising shapes** in artificial settings, such as curves with **local minima /maxima**. How widespread is this in practice?

No consensus on the shape of learning curves [2]. Best modelled by a **power law, exponential, …?**

3 Database Highlights

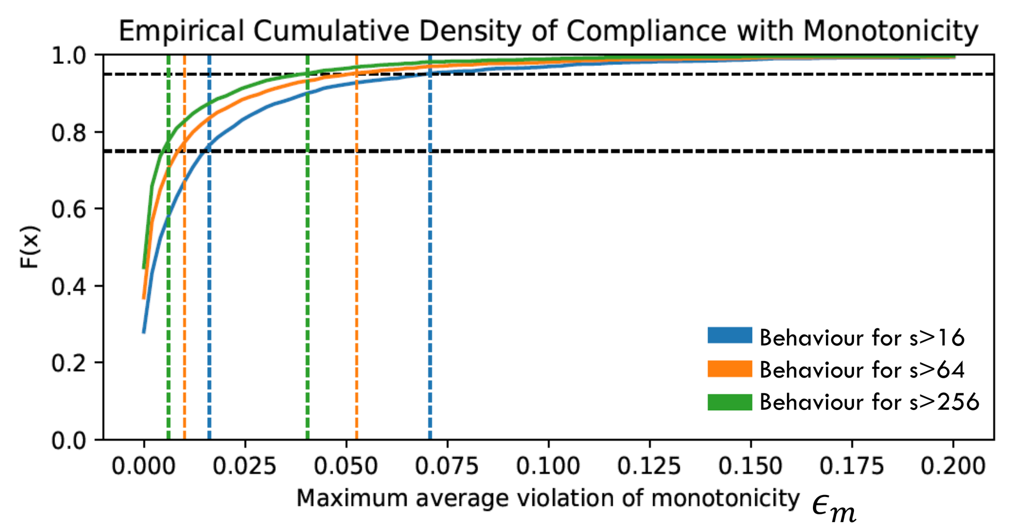
We publish a large database of learning curves:

- 20 learners on 246 datasets
- Getting our data: `pip install lcdb`
- Precomputed error rate, F1, AUC ROC, log loss
- Provide all predictions (can compute any metric)
- Bootstrapping: 25 train / validation / test splits
- Training set sizes $s_i = \left[2^{(7+i)/2}\right]$, $s = [16, 23, 32, \ldots]$

4 Preliminary Findings

A Are error rate curves monotone?

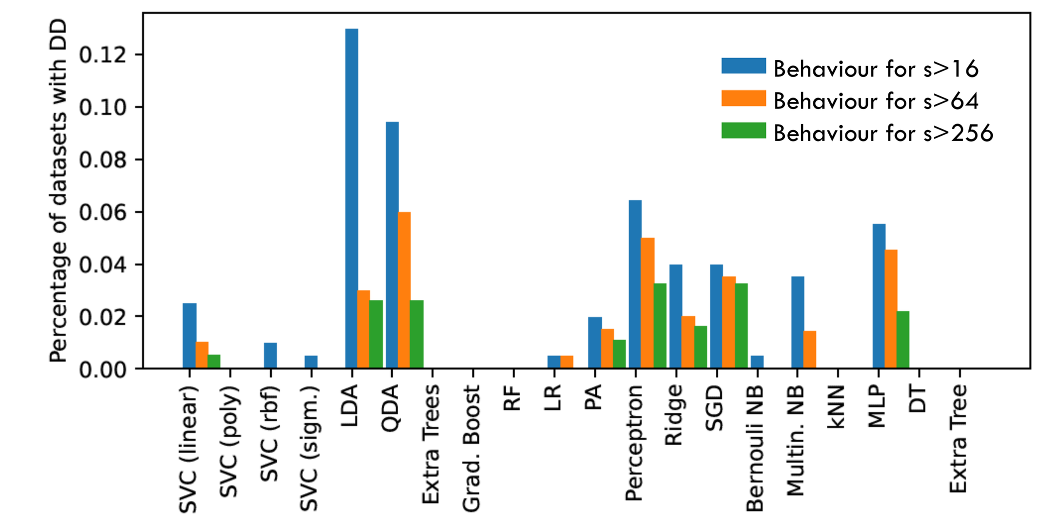
Define $\epsilon_m = \max_i \{\max(0, \hat{C}(s_{i+1}) - \hat{C}(s_i))\}$



Averaged over the whole database most curves seem monotone according to ϵ_m

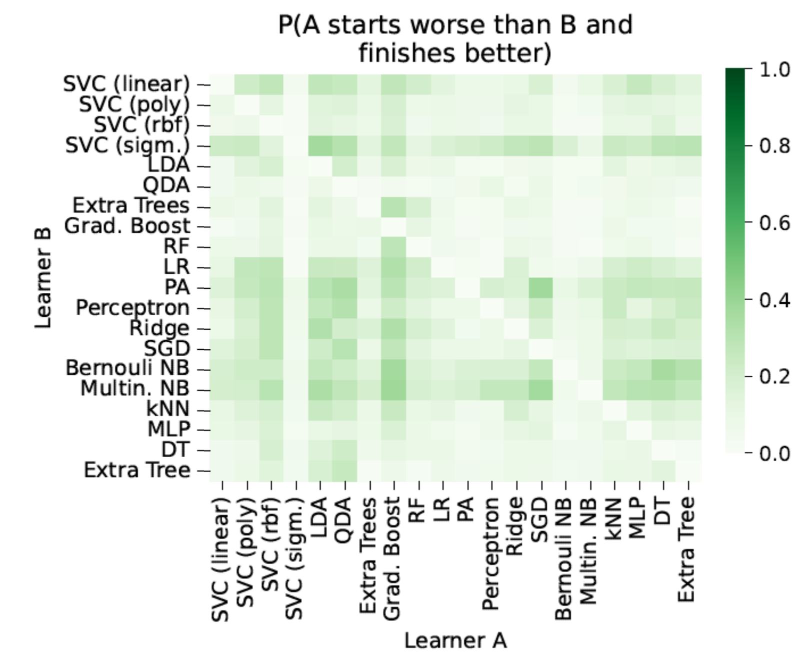
B Local maxima in error rate?

$$peak = \left[\!\left[\exists i < T : \hat{C}(s_i) < \hat{C}(s_{i+1})) \wedge \left(\exists u > i, \forall v \geq u : \hat{C}(s_{v-1}) > \hat{C}(s_v))\right)\right]\!\right]$$



Only some learners peak (local maximum)
Peaking lessens when s is larger

C Do curves cross?



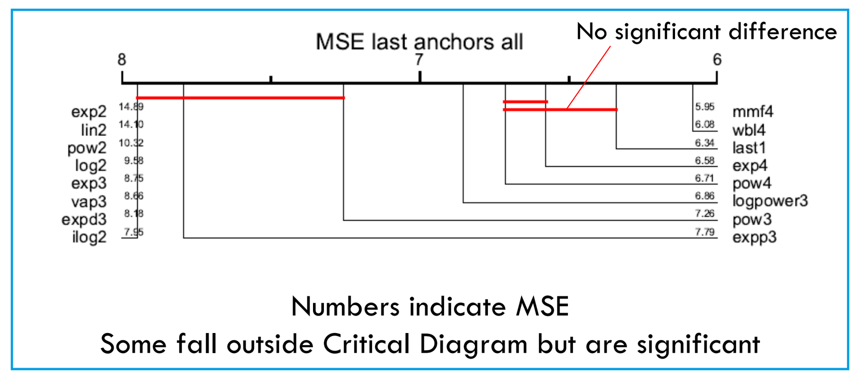
Yes curves cross 10-20% on average

D What shape do error rate curves have?

Experimental setup:

- Extrapolate to last point on the curve (not used in fitting!)
- Average 25 learning curves to smoothen curves
- Performance in Mean Squared Error (MSE)
- Repeat fitting 5 times and keep best fit by MSE
- Discard bad fits (e.g. MSE > 1, did not converge, etc.)
- Average over all datasets, learners, number of points used for fitting

Model	Formula	Model	Formula
last1	a	vap3	$\exp\left(a + \frac{b}{x} + c\log(x)\right)$
pow2	$-ax^{-b}$	expp3	$c - \exp\left((-b+x)^a\right)$
log2	$-a\log(x) + b$	expd3	$c - (-a+c)\,exp(-bx)$
exp2	$a\exp(-bx)$	logpow3	$a/\left((x\exp(-b))^c + 1\right)$
lin2	$ax + b$	pow4	$a - b(d+x)^{-c}$
ilog2	$-a/\log(x) + b$	mmf4	$(ab + cx^d)/(b + x^d)$
pow3	$a - bx^{-c}$	wbl4	$-b\exp(-ax^d) + c$
exp3	$a\exp(-bx) + c$	exp4	$c - \exp(-ax^d + b)$



Findings:

- Surprisingly: mmf4 and wbl4 perform best
- While prior studies found exp2/exp3/pow2/pow3 the best
- More points used for fitting: models with more parameters perform better
- In line with bias-variance trade-off

5 Discussion

Error rate: seemingly monotone, without too many local maxima, but do cross.

Extrapolation: mmf4, wbl4 perform best. Often excluded in prior studies. Other results agree with Brumen [3].

2% discarded fits: need for robust fitting!

6 Version 2.0?

- Support for pipelines (to implement feature scaling)
- Hyperparameter tuned models (we use defaults)
- Use monotonic training sets so that $s_1 \subset s_2 \subset s_3 \ldots$ (currently all training sets are sampled independently)

Suggestions for V2.0? Write them here on the poster:

[1] Mohr, F., van Rijn, J.N.: Learning curves for decision making in supervised machine learning - A survey. CoRR **abs/2201.12150** (2022)
[2] Viering, T.J., Loog, M.: The shape of learning curves: a review. CoRR **abs/2103.10948** (2021)
[3] Brumen, B., Cernezel, A., Bosnjak, L.: Overview of machine learning process modelling. Entropy **23**(9) (2021)