Lab Talk #4

April 3, 2023

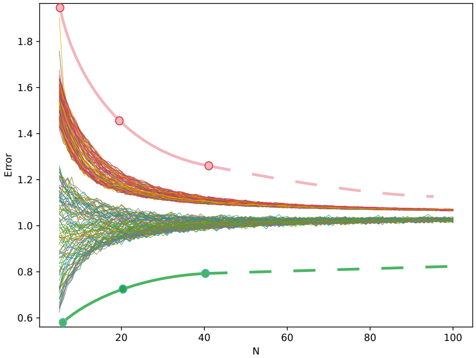
Ozgur Taylan Turan

"Learning" Learning Curves

Introduction

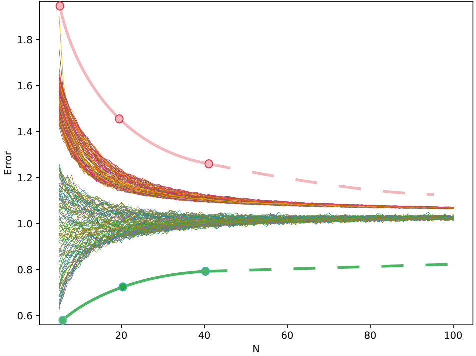
Learning Curves

- Not *Training Curves*
- Generalization Performance for a given N number of training points



Learning Curves

- Not *Training Curves*
- Generalization Performance for a given N number of training points



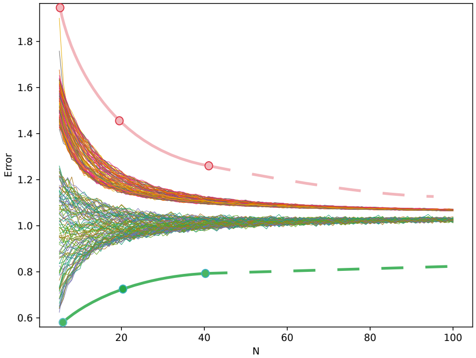
Learning Curves "formally"

- $\bar{\mathcal{R}}(\mathcal{A}, N) = \underset{\mathcal{D}_N}{\mathbb{E}} \mathcal{R}(\mathcal{A}(\mathcal{D}_N)))$

$\mathcal{D}_N := (x, y)_{i=1}^{N}$

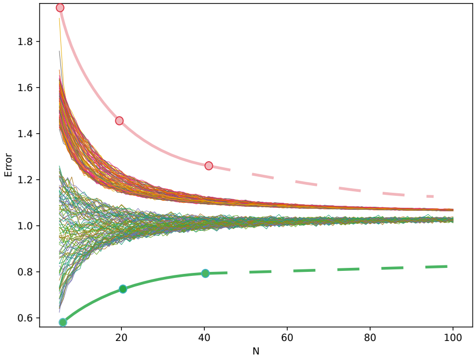
$\mathcal{A}(\mathcal{D}_N) \to h$

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



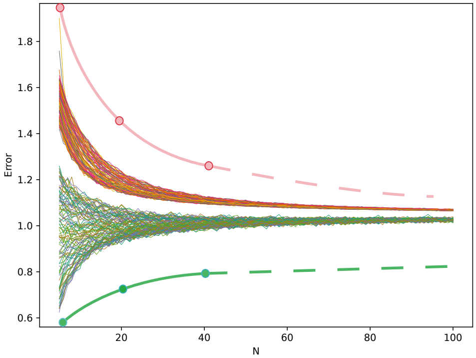
Learning Curves Importance

- How much data is enough?
- What is you generalization performance for a given N?
- Model Selection -> Hyper-parameter selection...



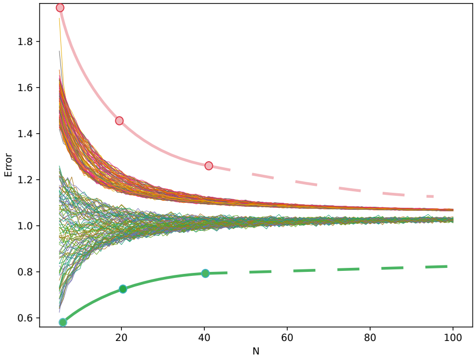
Learning Curves Importance

- How much data is enough?
- What is you generalization performance for a given N?
- Model Selection -> Hyper-parameter selection...



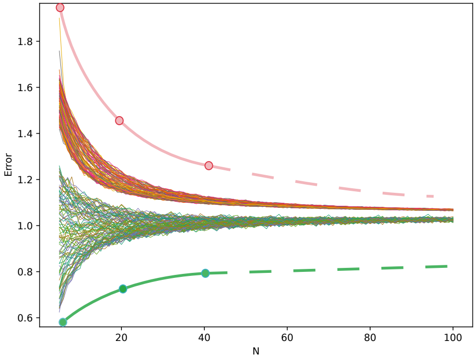
Learning Curves Importance

- How much data is enough?
- What is you generalization performance for a given N?
- Model Selection -> Hyper-parameter selection...



Learning Curves Extrapolation

- Parametric curve fitting (*e.g.* power law, exponential and logarithmic models)
- Marco's presentation about parametric fitting being really tough!
- Non-monotonic learning curves.

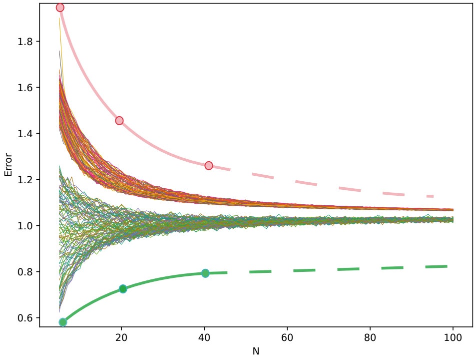


1

[1] M. Loog and T. J. Viering. "A Survey of Learning Curves with Bad Behavior: Or How More Data Need Not Lead to Better Performance". In: (), p. 16

Learning Curves Extrapolation

- Parametric curve fitting (*e.g.* power law, exponential and logarithmic models)
- Marco's presentation about parametric fitting being really tough!
- Non-monotonic learning curves.

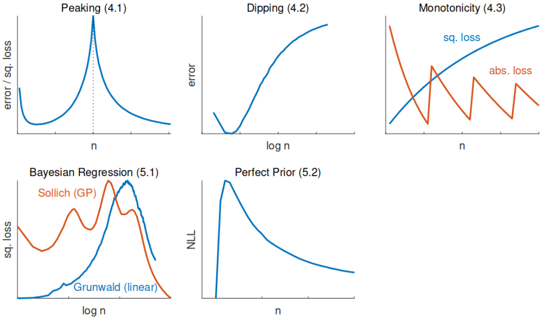


1

[1] M. Loog and T. J. Viering. "A Survey of Learning Curves with Bad Behavior: Or How More Data Need Not Lead to Better Performance". In: (), p. 16

Learning Curves Extrapolation

- Parametric curve fitting (*e.g.* power law, exponential and logarithmic models)
- Marco's presentation about parametric fitting being really tough!
- Non-monotonic learning curves.



[1]

[1] M. Loog and T. J. Viering. "A Survey of Learning Curves with Bad Behavior: Or How More Data Need Not Lead to Better Performance". In: (), p. 16

Research Questions

Question 1. What is the performance gain of a completely data-driven learning curve extrapolation compared to conventional parametric learning curve fitting?

Question 2. Performance of a data-driven learning curve for non-monotone curves?

Problem Definition

Learning Problem

For a learning task T_i

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

- $\varepsilon \sim \mathcal{N}(0, \sigma^2)$
- $\mathbb{H}$ Reproducing Hilbert Space
- $\mathcal{M}$ Model
- $\tilde{\mathcal{M}}$ Model with additional functions
- ψ_p additional available functions

Learning Problem

With the objective as,

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

Kernel Ridge learner is obtained via *Non-parametric Representer Theorem*[1]

- $\mathcal{M}(\cdot) = \sum_i^{\mathcal{Z}} \alpha_i k(\cdot, \mathcal{D}_{\mathrm{N}_i})$

- $\varepsilon \sim \mathcal{N}(0, \sigma^2)$
- $\mathbb{H}$ Reproducing Hilbert Space
- $\mathcal{M}$ Model
- $\tilde{\mathcal{M}}$ Model with additional functions
- ψ_p additional available functions

[1] B. Schölkopf and A. J. Smola (2002). *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Adaptive Computation and Machine Learning. Cambridge, Mass: MIT Press. 626 pp. ISBN: 978-0-262-19475-4

Learning Problem

With the objective as,

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

If you assume $\tilde{\mathcal{M}} = \mathcal{M} + h$ where $h \in span\{\psi_p\}$ and $\{\psi_p\}_{p=1}^{\mathcal{Z}}$ via *Semi-parametric Representer Theorem* [1]

- $\tilde{\mathcal{M}}(\cdot) = \sum_i^Q \alpha_i k(\cdot, N_i) + \sum_j^M \beta_j \psi_j(\cdot)$

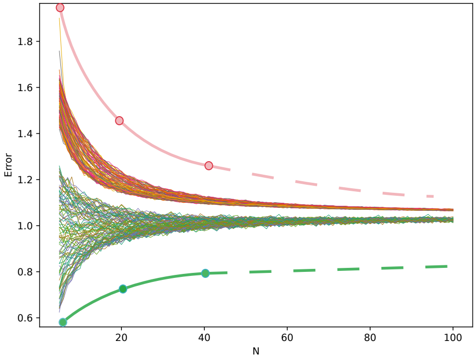
- $\varepsilon \sim \mathcal{N}(0, \sigma^2)$
- $\mathbb{H}$ Reproducing Hilbert Space
- $\mathcal{M}$ Model
- $\tilde{\mathcal{M}}$ Model with additional functions
- ψ_p additional available functions

[1] B. Schölkopf and A. J. Smola (2002). *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Adaptive Computation and Machine Learning. Cambridge, Mass: MIT Press. 626 pp. ISBN: 978-0-262-19475-4

Learning Problem

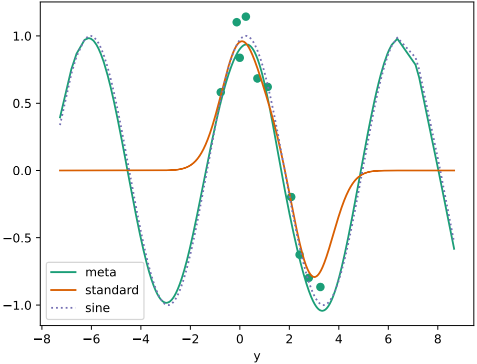
If you assume $\tilde{\mathcal{M}} = \mathcal{M} + h$ where $h \in span\{\psi_p\}$ and $\{\psi_p\}_{p=1}^{\mathcal{Z}}$ via *Semi-parametric Representer Theorem* [1]

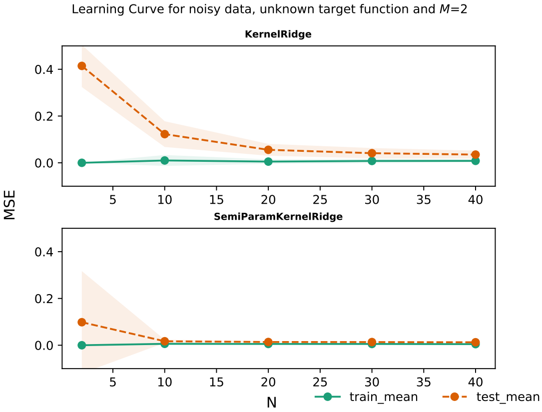
- $\tilde{\mathcal{M}}(\cdot) = \sum_i^Q \alpha_i k(\cdot, N_i) + \sum_j^M \beta_j \psi_j(\cdot)$



[1] B. Schölkopf and A. J. Smola (2002). *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Adaptive Computation and Machine Learning. Cambridge, Mass: MIT Press. 626 pp. ISBN: 978-0-262-19475-4

Learning Problem





Initial Questions

Significance of α

Significance of α

- $\mathcal{M}_1 := \sum_j^M \beta_j \psi_j(\cdot)$
- $\mathcal{M}_2 := \sum_i^Q \alpha_i k(\cdot, N_i) + \sum_j^M \beta_j \psi_j(\cdot)$
- $\mathcal{M}_3 := \sum_i^Q \alpha_i k(\cdot, N_i)$

Statistical Hypothesis Testing

- t test
- f test
- chi square test
- Wilcovon Rank-Sum
- $\vdots$
- Cramer Von Misses

Significance of α

- $\mathcal{M}_1 := \sum_j^M \beta_j \psi_j(\cdot)$
- $\mathcal{M}_2 := \sum_i^Q \alpha_i k(\cdot, N_i) + \sum_j^M \beta_j \psi_j(\cdot)$
- $\mathcal{M}_3 := \sum_i^Q \alpha_i k(\cdot, N_i)$

Controlled Environment (σ, $\boldsymbol{psi}$ look at the Extrapolation+Interpolation Error populations. Informative $\boldsymbol{\psi}$

- All combinations are significantly different $\rightarrow \sigma = 0$
- $\mathcal{M}_1 - \mathcal{M}_2$ not different, but other combinations are different $\rightarrow sigma \neq$

Significance of α

Final verdict: Go with the more flexible method!

Selection of ψ

- Using the raw curves
- Extracting information from learning curves...

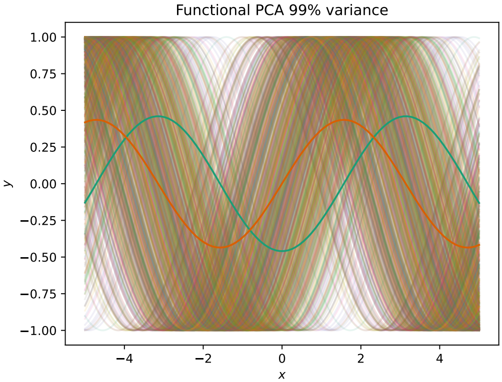
- Functional ($\mathbb{R}^d \rightarrow \mathbb{R}$) Analysis
- equal spacing, same region etc...

Selection of ψ

- Using the raw curves
- Extracting information from learning curves...

- Functional ($\mathbb{R}^d \rightarrow \mathbb{R}$) Analysis
- equal spacing, same region etc...

Selection of ψ



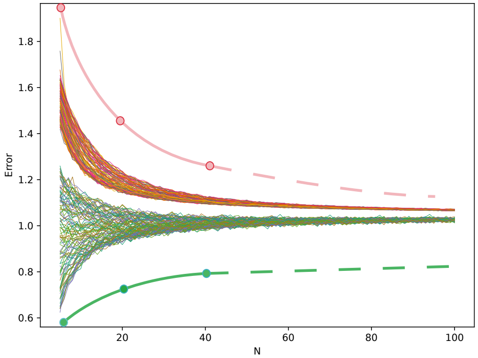
- Functional ($\mathbb{R}^d \to \mathbb{R}$) Analysis
- equal spacing, same region etc...

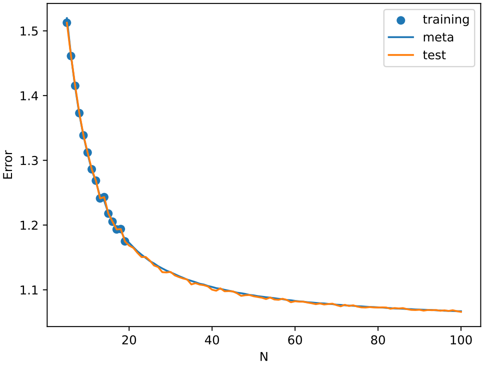
Selection of ψ

- Functional ($\mathbb{R}^d \rightarrow \mathbb{R}$) Analysis
- equal spacing, same region etc...

Final verdict: functional PCA usage is especially important for noisy curves due to smoothing introduced!

How does it look now?





Curve Fitting Problems

- Marco's talk
- $p(x, a, b, c) := a * e^{(b*x)} + c$

- De Facto $\rightarrow$ Levenberg–Marquardt method (Gradient Descent + Gauss Newton)
- Playing with the internal optimizer parameters. (Not helping!)

Curve Fitting Problems

- Marco's talk
- $p(x,a,b,c) := a * e^{(b*x)} + c$

- De Facto $\rightarrow$ Levenberg–Marquardt method (Gradient Descent + Gauss Newton)
- Playing with the internal optimizer parameters. (Not helping!)

Curve Fitting Problems

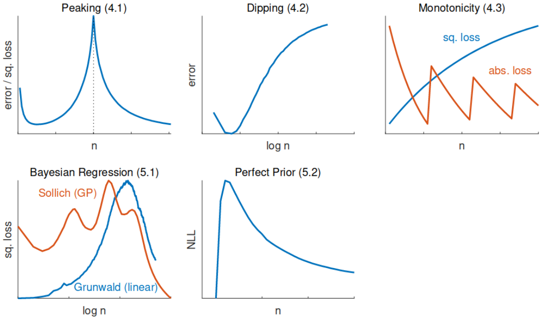
- Marco's talk
- $p(x, a, b, c) := a * e^{(b*x)} + c$

- De Facto $\rightarrow$ Levenberg–Marquardt method (Gradient Descent + Gauss Newton)
- Playing with the internal optimizer parameters. (Not helping!)

Final verdict: Try various optimizers and various runs get the best one!

Plans

To Do



[1]

- Generating learning curve data
- Comparing performance of semi-parametric kernel ridge to parametric curve fitting in vari/us settings. (*e.g.* changing hyper-parameters, $\mathcal{P}_{X,Y}$, and learners...)

[1] M. Loog and T. J. Viering. "A Survey of Learning Curves with Bad Behavior: Or How More Data Need Not Lead to Better Performance". In: (), p. 16

Thanks!