A U-turn on Double Descent: Rethinking Parameter Counting in Statistical Learning

Alicia Curth et al (2023)

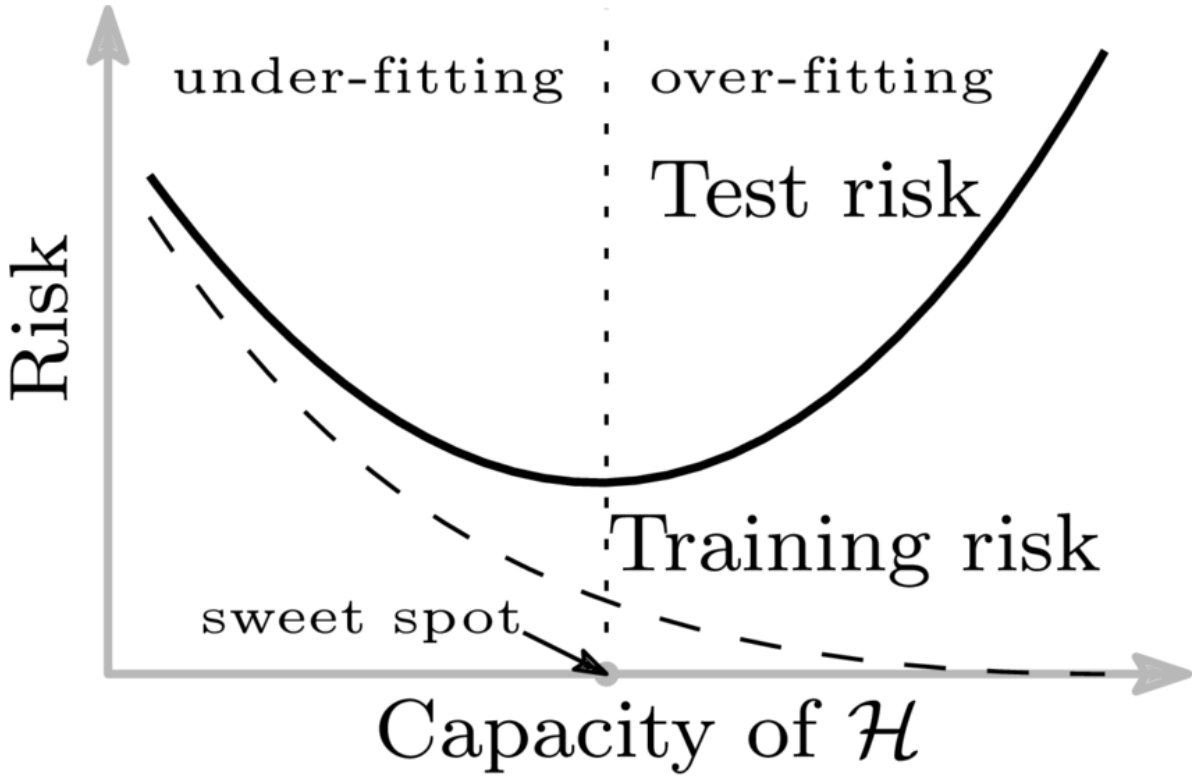
presented by Insung Kong

Seoul National University

2024 1/8

Outline

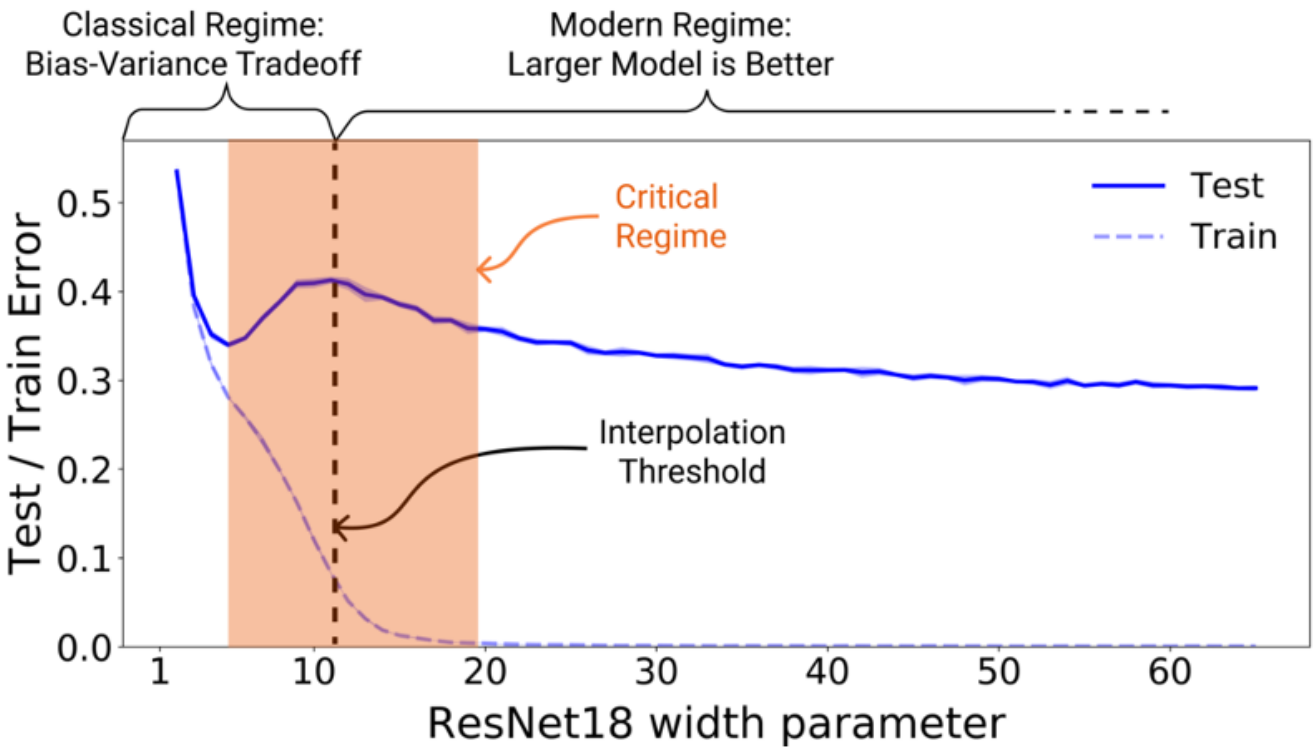
Backgrounds



- Traditional U-shape
- Model complexity $\uparrow \implies$ Bias $\downarrow$, Variance $\uparrow$

Double Descent

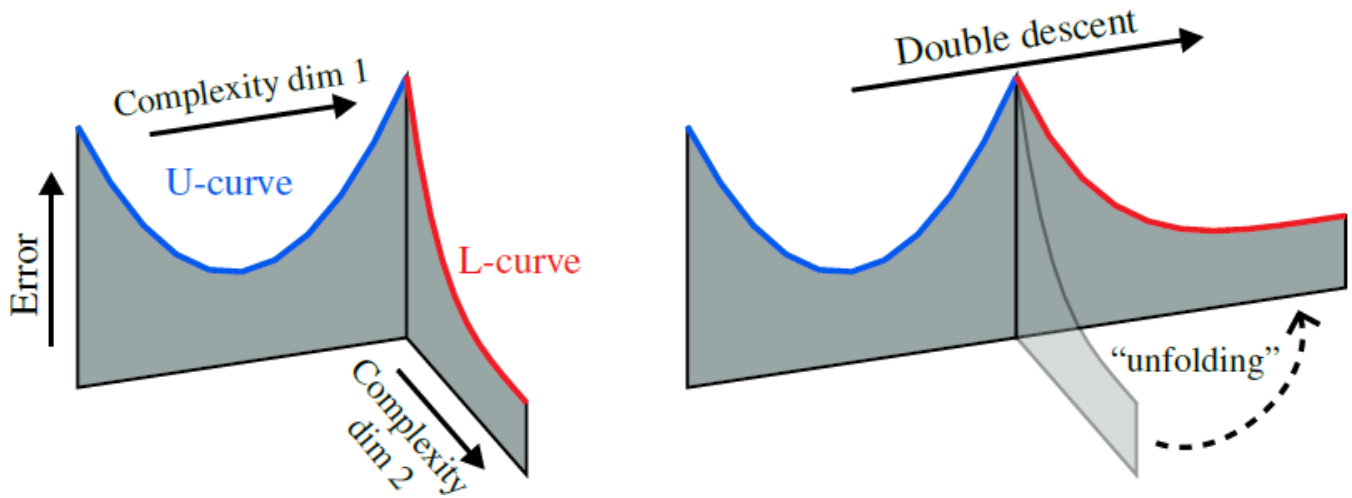
- In deep learning,



- Moreover, Belkin et al. (2019) demonstrate that double descent ubiquitously appears across many non-deep learning methods such as **trees**, **boosting** and even **linear regression**. (cited 1000+)

Contribution

- In Part 1, Curth et al. (2023) show that for non-deep double descent, there is implicitly more than one complexity axis along which the parameter count grows



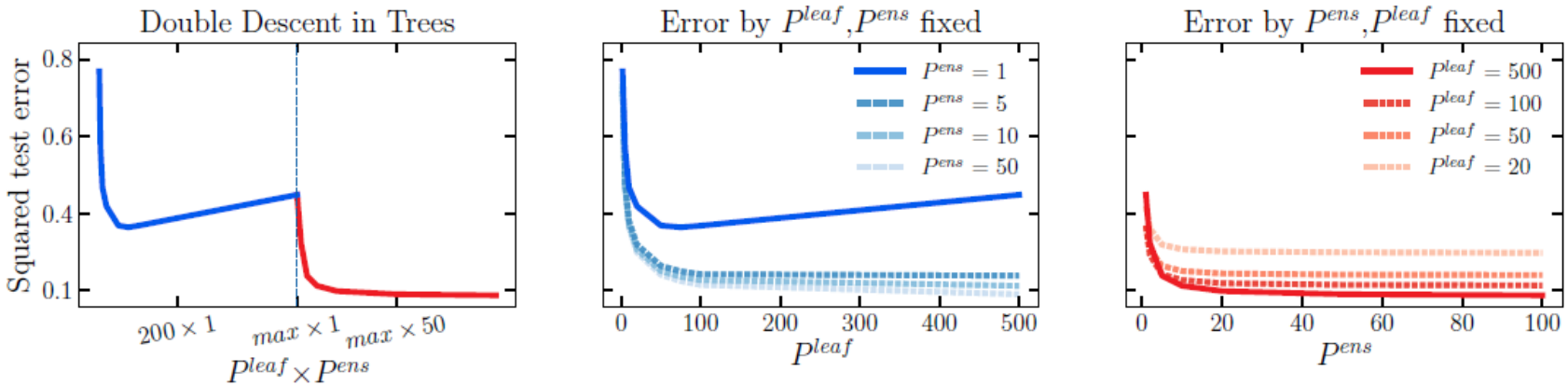
- In Part 2, Curth et al. (2023) propose a generalized measure for the effective number of parameter (for smoothers)

Outline

Understanding double descent in trees

- P^{leaf} : **the maximum allowed number of terminal leaf nodes**
- In experiments of Belkin et al. (2019), the number of model parameters is initially controlled through P^{leaf}.
- However, P^{leaf} for a single tree cannot be increased past n, which is when every leaf contains only one instance.
- P^{ens} : **the number of different trees grown to full depth**, where each tree will generally be distinct due to the randomness in features considered for each split.
- When $P^{ens} > 1$, this is actually an ensemble of trees (i.e. a random forest without bootstrapping)

Understanding double descent in trees



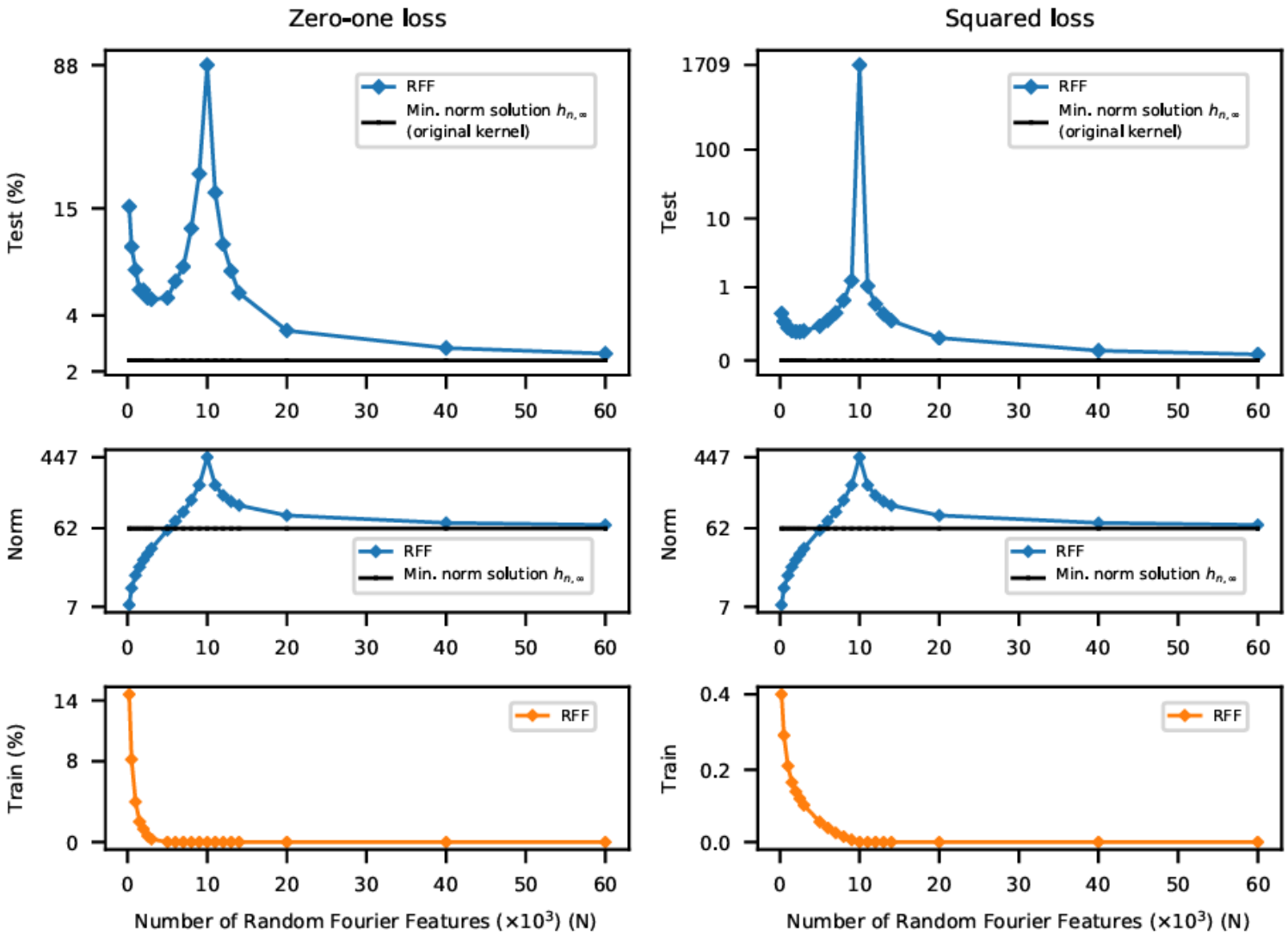
- Left : evidence of double descent given by Belkin et al. (2019)
- Center : fixed P^{ens}, error exhibits U-shape
- Right : fixed P^{leaf}, error exhibits L-shape

Understanding double descent in linear regression

- $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathbb{R}^d$: input vectors
- In order to flexibly control the number of model parameters, Belkin et al. (2019) apply basis expansions using random Fourier features (RFF).
- P^ϕ : the number of raw model parameters
- For $p \in [P^\phi]$, $\phi_p(\boldsymbol{x}_i) = \text{Re}\left(\exp^{\sqrt{-1}\boldsymbol{v}_p^T \boldsymbol{x}_i}\right)$ where $\boldsymbol{v}_p \overset{\text{iid}}{\sim} \mathcal{N}\left(\boldsymbol{0}, \frac{1}{5^2} \cdot \boldsymbol{I}_d\right)$.
- For $n \times P^\phi$ random design matrix $\boldsymbol{\Phi}$, obtain
  - Least square solution if $P^\phi \leq n$
  - Min-norm solution if $P^\phi > n$

Understanding double descent in linear regression

- Results by Belkin et al. (2019)



Proposition 1 (Min-norm least squares as dimensionality reduction.)

For a full rank matrix $X \in \mathbb{R}^{n \times d}$ with $n < d$ and a vector of targets $\boldsymbol{y} \in \mathbb{R}^n$, the min-norm least squares solution

$$\hat{\boldsymbol{\beta}}^{MN} = \left\{ \min_{\boldsymbol{\beta}} \|\boldsymbol{\beta}\|_2^2 : \mathrm{X}\boldsymbol{\beta} = \boldsymbol{y} \right\}$$

and the least squares solution

$$\hat{\boldsymbol{\beta}}^{SVD} = \{\boldsymbol{\beta} : \mathrm{B}\boldsymbol{\beta} = \boldsymbol{y}\}$$

using the matrix of basis vectors $B \in \mathbb{R}^{n \times n}$, constructed using the first n right singular vectors of X, are equivalent: i.e.,

$$\boldsymbol{x}^T \hat{\boldsymbol{\beta}}^{MN} = \boldsymbol{b}^T \hat{\boldsymbol{\beta}}^{SVD}$$

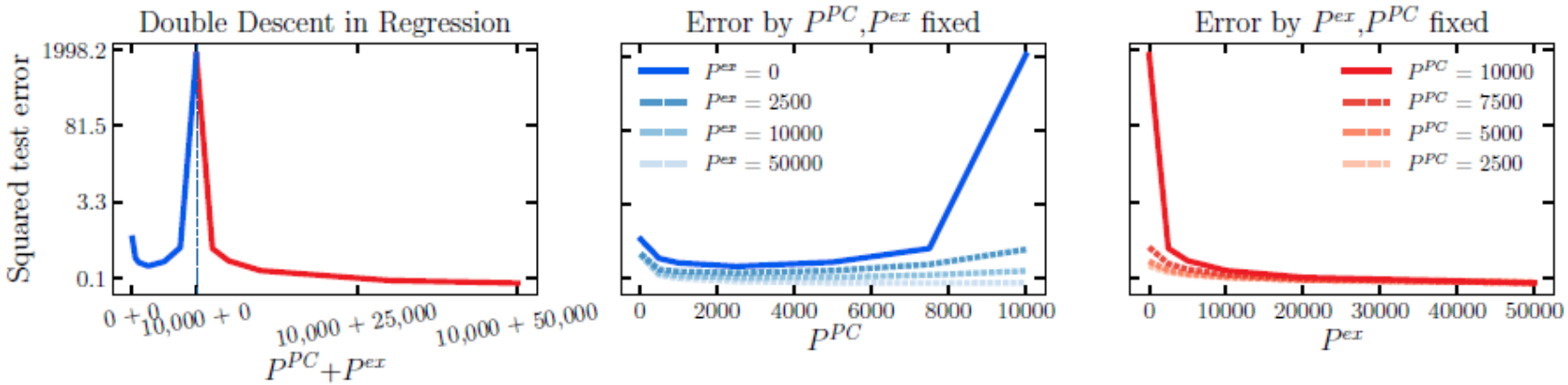
for all $\boldsymbol{x} \in \mathbb{R}^d$ and corresponding basis representation $\boldsymbol{b} \equiv \boldsymbol{b}(\boldsymbol{x})$.

Understanding double descent in linear regression

- When $P^{\phi} < n$, the addition of feature dimensions does correspond to an increase in fitted model parameters.
- When $P^{\phi} > n$, performance gains are better explained as a linear model of fixed size n being fit to an increasingly rich basis constructed in an unsupervised step.

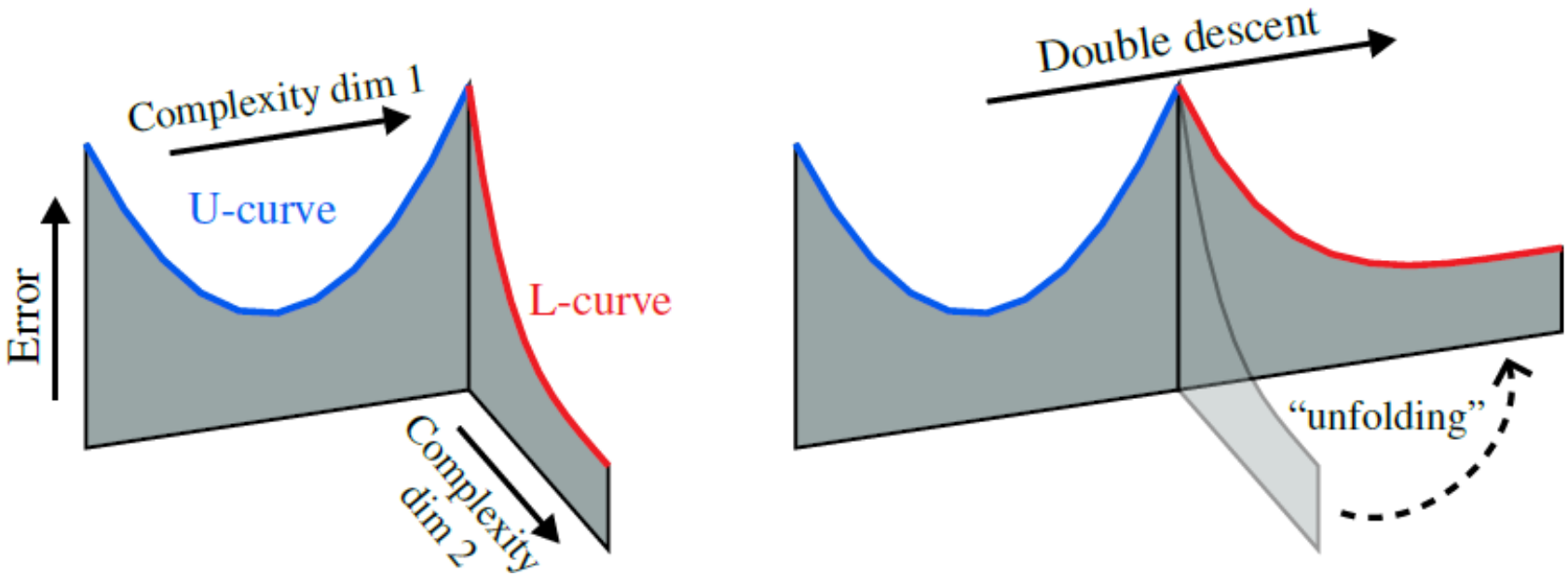
- One can consider selecting the top P^{PC} ($\leq n$) principal components and fitting a linear model to that basis.
- The number of excess features $P^{ex} := P^{\phi} - P^{PC}$ is the number of raw dimensions that only contribute to the creation of a richer basis.

Understanding double descent in linear regression



- Left : evidence of double descent given by Belkin et al. (2019)
- Center : fixed P^{ex}, error exhibits U-shape
- Right : fixed P^{PC}, error exhibits L-shape

Conclusion of part 1



- For non-deep double descent, there is implicitly more than one complexity axis along which the parameter count grows.

Outline

Rethinking parameter counting

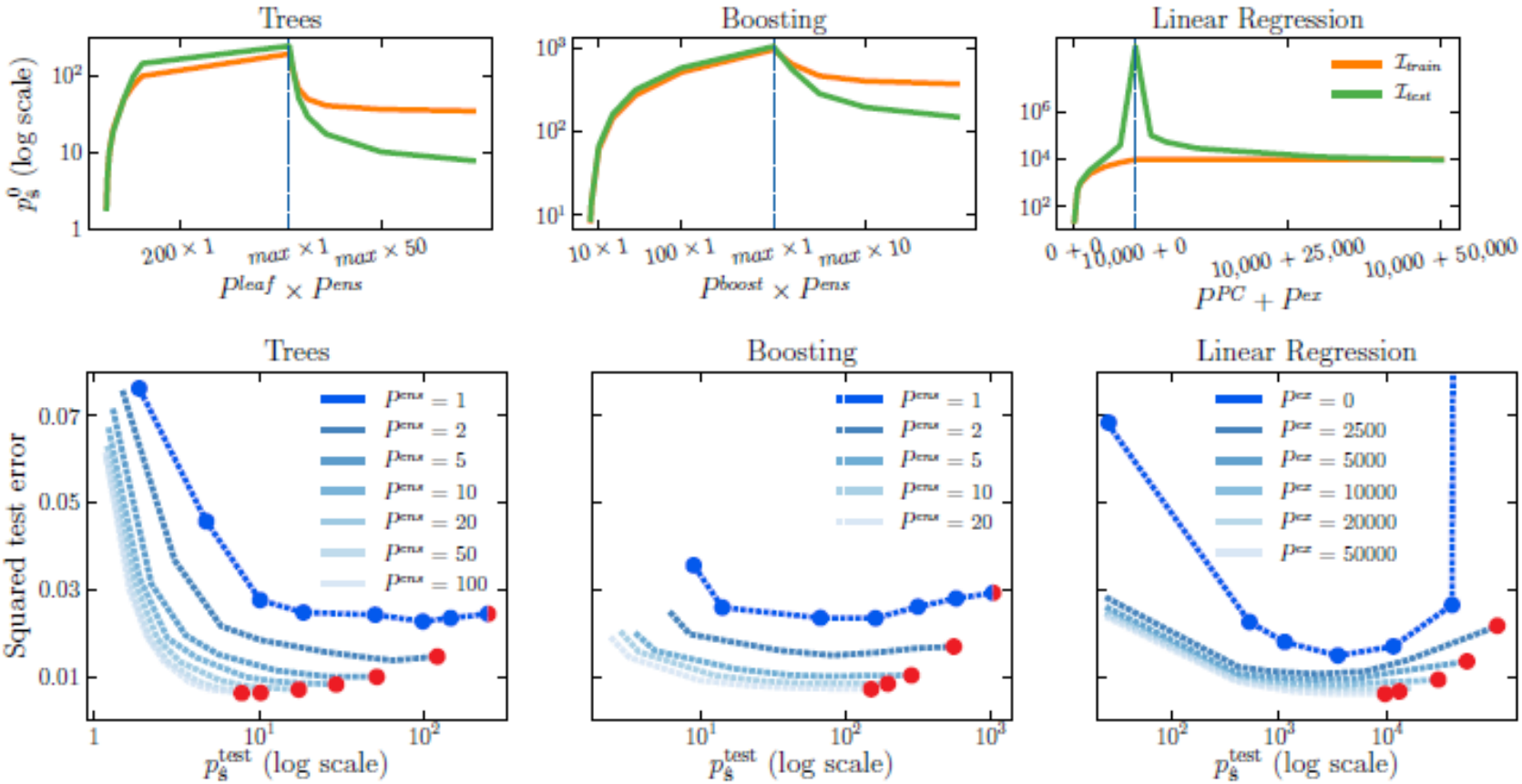
- For train data $\mathcal{D}^{train} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ and new input $\boldsymbol{z} \in \mathcal{X}$, the prediction of a smoother is

$$\hat{f}(\boldsymbol{z}) = \hat{\boldsymbol{s}}(\boldsymbol{z})^\top \boldsymbol{y}_{train},$$

  where $\hat{\boldsymbol{s}}(\boldsymbol{z}) \in \mathbb{R}^n$ and $\boldsymbol{y}_{train} = (y_1, \ldots, y_n)^\top$.
- Previous examples (tree, boosting, linear) are examples of smoothers.
- Curth et al. (2023) adapt the variance based effective parameter definition : for a set of new inputs $\{\boldsymbol{z}_j\}_{j \in \mathcal{I}_0}$

$$p^0_{\hat{\boldsymbol{s}}} \equiv p\left(\mathcal{I}_0, \hat{\boldsymbol{s}}(\cdot)\right) = \frac{n}{|\mathcal{I}_0|} \sum_{j \in \mathcal{I}_0} \left\| \hat{\boldsymbol{s}}\left(\boldsymbol{z}_j\right) \right\|^2$$

Rethinking parameter counting



References I

Belkin, M., Hsu, D., Ma, S., and Mandal, S. (2019). Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854.

Curth, A., Jeffares, A., and van der Schaar, M. (2023). A u-turn on double descent: Rethinking parameter counting in statistical learning. In *Thirty-seventh Conference on Neural Information Processing Systems*.