Revisiting Model Complexity in the Wake of Overparameterized Machine Learning

Pratik Patil

University of California, Berkeley

MDS 2024

Based on joint work with Jin-Hong Du and Ryan Tibshirani

`https://pratikpatil.io/papers/model-complexity.pdf`

Overparametrization in machine learning

Modern machine learning models typically fit a huge number of parameters. Such overparameterization seems to be useful for:

- Representation: allows rich, expressive models for diverse real data
- Optimization: simple, local optimization methods often find near-optimal solutions to empirical risk minimization problem
- Generalization: despite overfitting, models generalize well in practice

This talk is about generalization aspect in overparameterized learning.

Overparametrization in machine learning

Modern machine learning models typically fit a huge number of parameters. Such overparameterization seems to be useful for:

- Representation: allows rich, expressive models for diverse real data
- Optimization: simple, local optimization methods often find near-optimal solutions to empirical risk minimization problem
- Generalization: despite overfitting, models generalize well in practice

This talk is about generalization aspect in overparameterized learning.

Overparametrization in machine learning

Modern machine learning models typically fit a huge number of parameters. Such overparameterization seems to be useful for:

- Representation: allows rich, expressive models for diverse real data
- Optimization: simple, local optimization methods often find near-optimal solutions to empirical risk minimization problem
- Generalization: despite overfitting, models generalize well in practice

This talk is about generalization aspect in overparameterized learning.

Overparametrization in machine learning

Modern machine learning models typically fit a huge number of parameters. Such overparameterization seems to be useful for:

- Representation: allows rich, expressive models for diverse real data
- Optimization: simple, local optimization methods often find near-optimal solutions to empirical risk minimization problem
- Generalization: despite overfitting, models generalize well in practice

This talk is about generalization aspect in overparameterized learning.

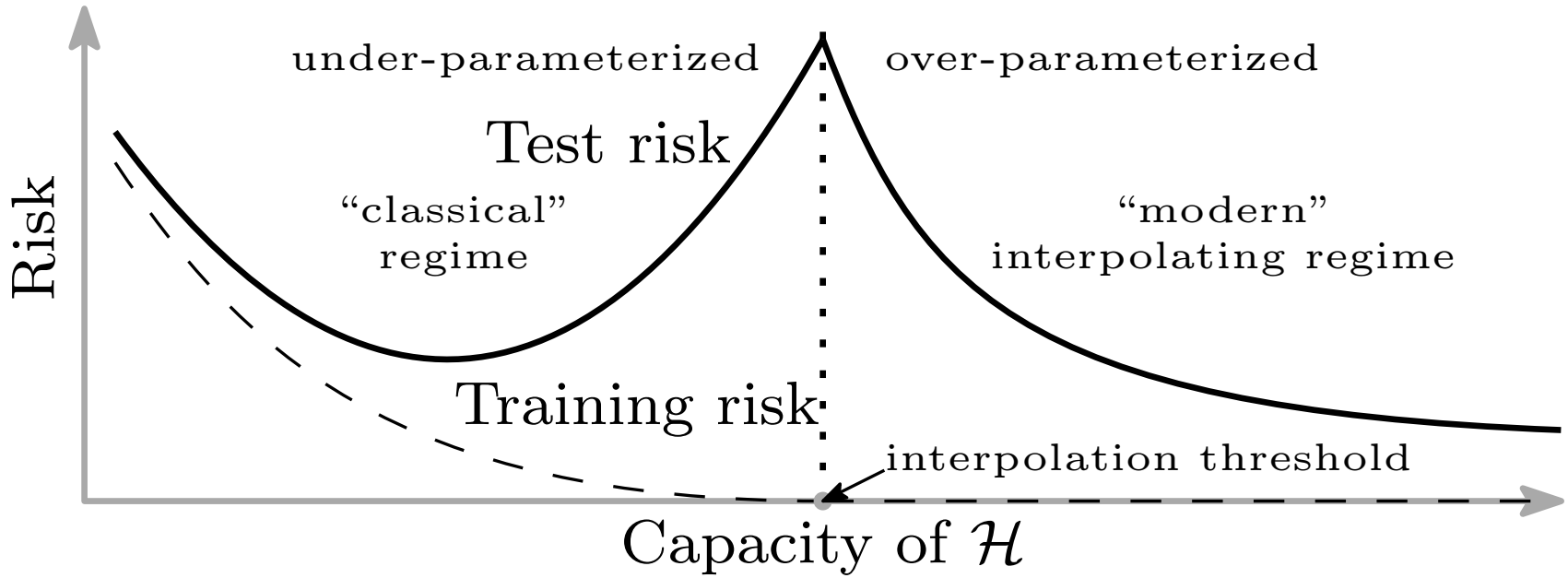
Overparametrization in machine learning

Modern machine learning models typically fit a huge number of parameters. Such overparameterization seems to be useful for:

- Representation: allows rich, expressive models for diverse real data
- Optimization: simple, local optimization methods often find near-optimal solutions to empirical risk minimization problem
- Generalization: despite overfitting, models generalize well in practice

This talk is about generalization aspect in overparameterized learning.

The double/multiple descent phenomenon

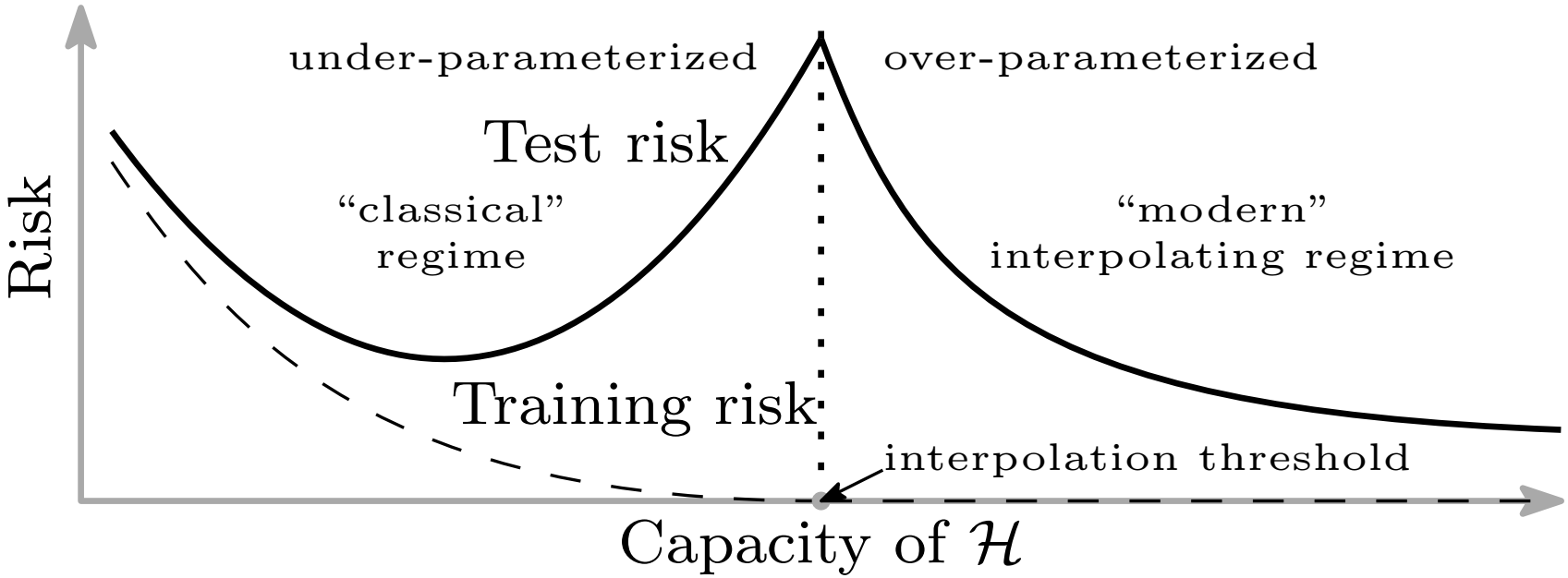


"Reconciling modern machine learning practice and the bias variance tradeoff"
Belkin, Hsu, Ma, Mandal, 2018

- The phenomenon is dubbed "double descent" in the risk curve.
- This trend holds for many model classes including linear regression, kernel regression, random forest, boosting, neural networks, etc.

Is there an equivalent "reparameterization" of "overparameterization"?

The double/multiple descent phenomenon

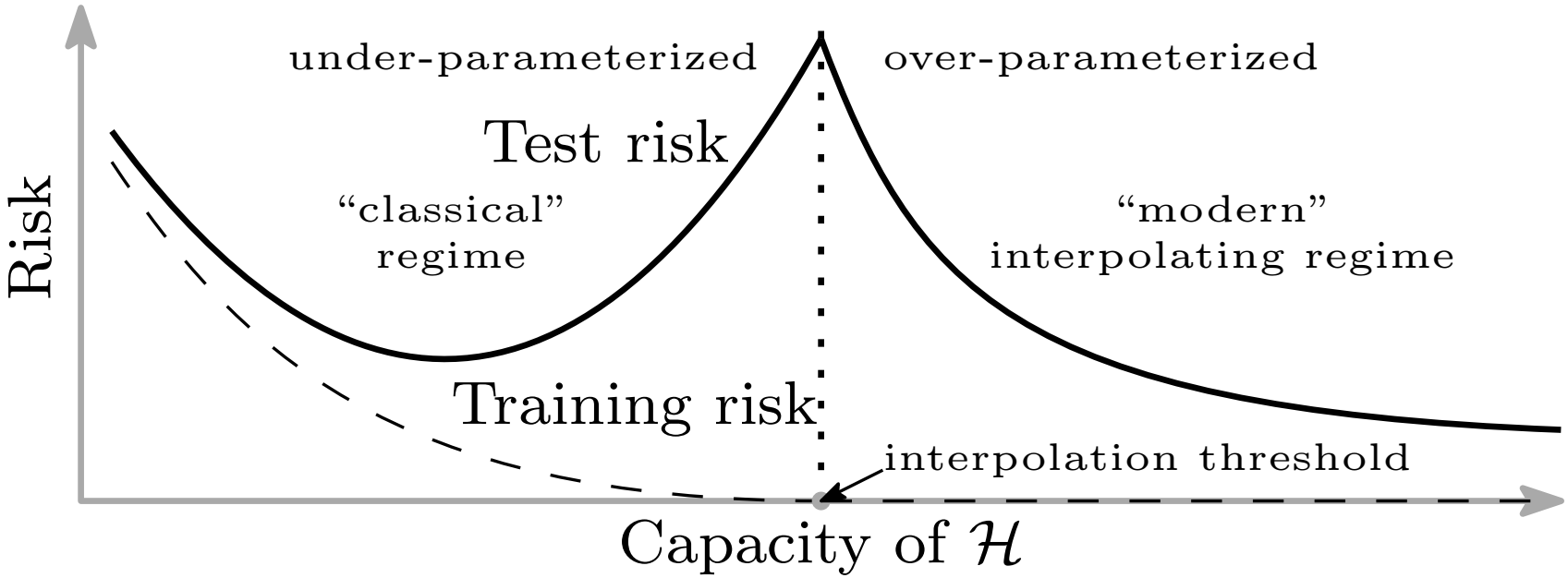


"Reconciling modern machine learning practice and the bias variance tradeoff"
Belkin, Hsu, Ma, Mandal, 2018

- The phenomenon is dubbed "double descent" in the risk curve.
- This trend holds for many model classes including linear regression, kernel regression, random forest, boosting, neural networks, etc.

Is there an equivalent "reparameterization" of "overparameterization"?

The double/multiple descent phenomenon

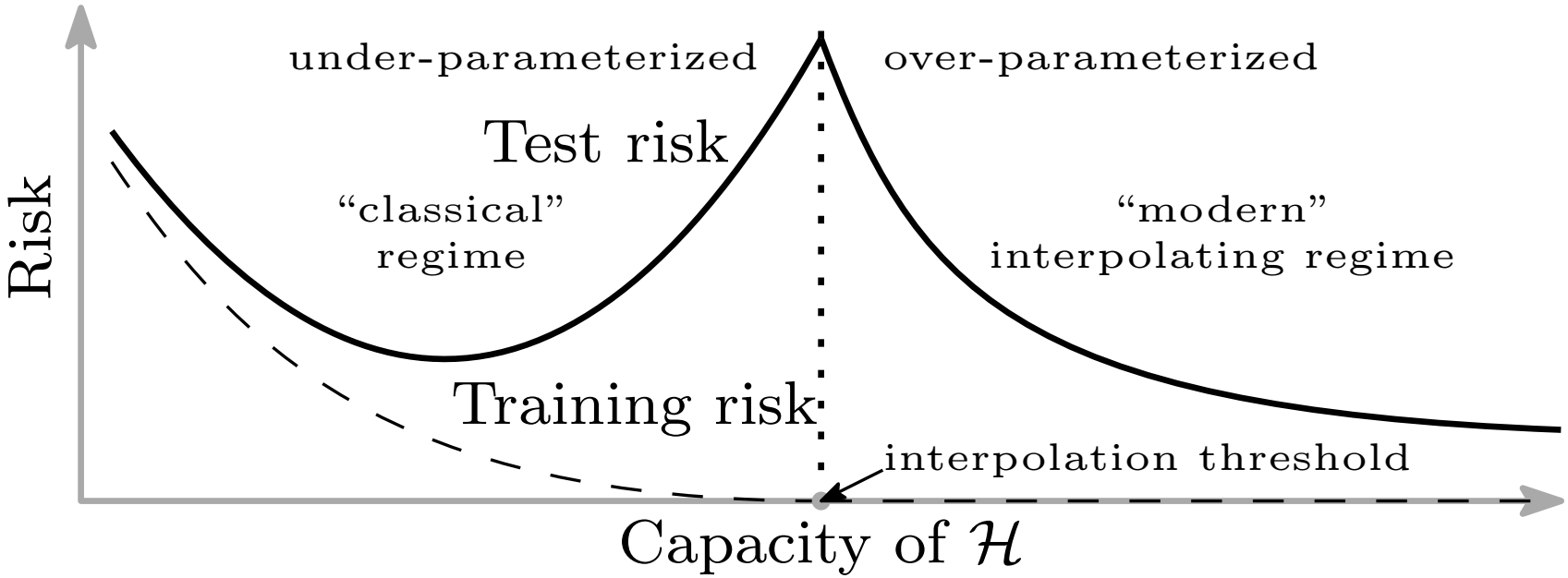


"Reconciling modern machine learning practice and the bias variance tradeoff"
Belkin, Hsu, Ma, Mandal, 2018

- The phenomenon is dubbed "double descent" in the risk curve.
- This trend holds for many model classes including linear regression, kernel regression, random forest, boosting, neural networks, etc.

Is there an equivalent "reparameterization" of "overparameterization"?

The double/multiple descent phenomenon

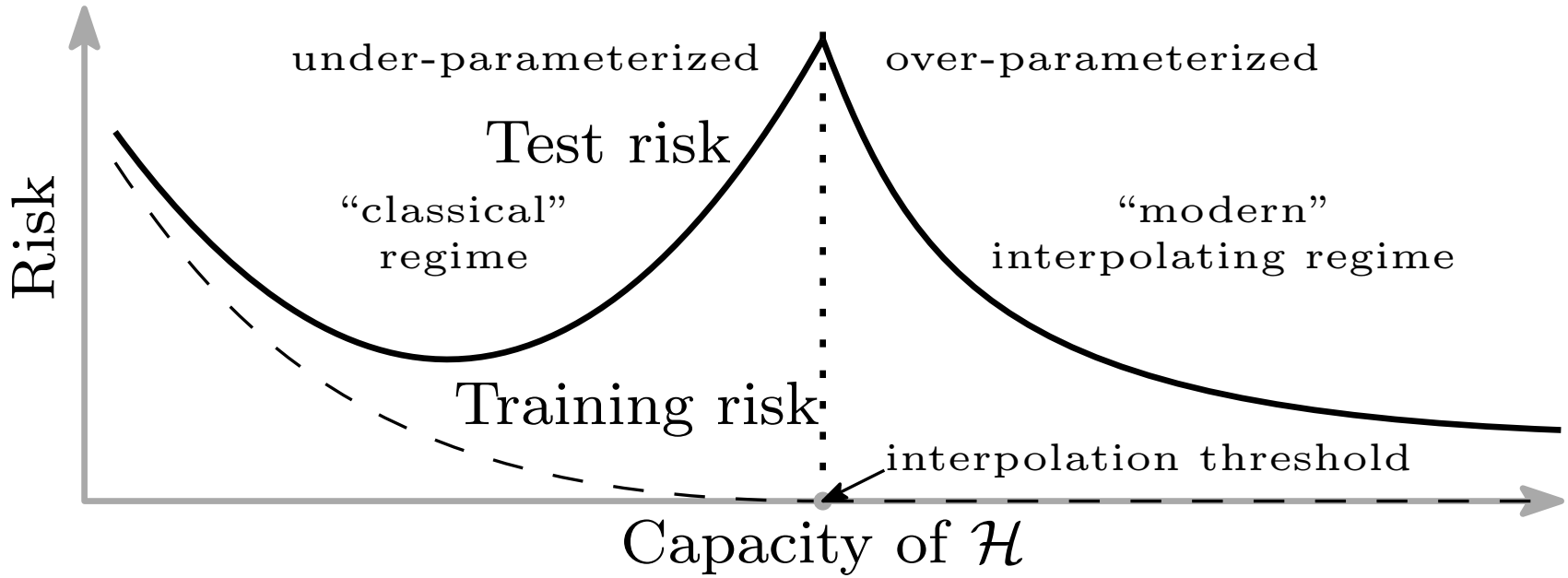


"Reconciling modern machine learning practice and the bias variance tradeoff"
Belkin, Hsu, Ma, Mandal, 2018

- The phenomenon is dubbed "double descent" in the risk curve.
- This trend holds for many model classes including linear regression, kernel regression, random forest, boosting, neural networks, etc.

Is there an equivalent "reparameterization" of "overparameterization"?

The double/multiple descent phenomenon



"Reconciling modern machine learning practice and the bias variance tradeoff"
Belkin, Hsu, Ma, Mandal, 2018

- The phenomenon is dubbed "double descent" in the risk curve.
- This trend holds for many model classes including linear regression, kernel regression, random forest, boosting, neural networks, etc.

Is there an equivalent "reparameterization" of "overparameterization"?

Recent theoretical developments

Understanding generalization of interpolators in simpler settings:

- Linear regression
  - Hastie, Montanari, Rosset, Tibshirani, 2019
  - Belkin, Hsu, Xu, 2019
  - Muthukumar, Vodrahalli, Sahai, 2019
  - Bartlett, Long, Lugosi, Tsigler, 2019
  - Mei, Montanari, 2019
- Kernel regression
  - Liang, Rakhlin, 2018
  - Liang, Rakhlin, Zhai, 2019
- Local methods
  - Belkin, Hsu, Mitra, 2018
  - Belkin, Rakhlin, Tsybakov, 2018
- and many more ...

Recent theoretical developments

Understanding generalization of interpolators in simpler settings:

- Linear regression
  - Hastie, Montanari, Rosset, Tibshirani, 2019
  - Belkin, Hsu, Xu, 2019
  - Muthukumar, Vodrahalli, Sahai, 2019
  - Bartlett, Long, Lugosi, Tsigler, 2019
  - Mei, Montanari, 2019
- Kernel regression
  - Liang, Rakhlin, 2018
  - Liang, Rakhlin, Zhai, 2019
- Local methods
  - Belkin, Hsu, Mitra, 2018
  - Belkin, Rakhlin, Tsybakov, 2018
- and many more ...

Recent theoretical developments

Understanding generalization of interpolators in simpler settings:

- Linear regression
  - Hastie, Montanari, Rosset, Tibshirani, 2019
  - Belkin, Hsu, Xu, 2019
  - Muthukumar, Vodrahalli, Sahai, 2019
  - Bartlett, Long, Lugosi, Tsigler, 2019
  - Mei, Montanari, 2019
- Kernel regression
  - Liang, Rakhlin, 2018
  - Liang, Rakhlin, Zhai, 2019
- Local methods
  - Belkin, Hsu, Mitra, 2018
  - Belkin, Rakhlin, Tsybakov, 2018
- and many more ...

Recent theoretical developments

Understanding generalization of interpolators in simpler settings:

- Linear regression
  - Hastie, Montanari, Rosset, Tibshirani, 2019
  - Belkin, Hsu, Xu, 2019
  - Muthukumar, Vodrahalli, Sahai, 2019
  - Bartlett, Long, Lugosi, Tsigler, 2019
  - Mei, Montanari, 2019
- Kernel regression
  - Liang, Rakhlin, 2018
  - Liang, Rakhlin, Zhai, 2019
- Local methods
  - Belkin, Hsu, Mitra, 2018
  - Belkin, Rakhlin, Tsybakov, 2018
- and many more ...

Recent theoretical developments

Understanding generalization of interpolators in simpler settings:

- Linear regression
  - Hastie, Montanari, Rosset, Tibshirani, 2019
  - Belkin, Hsu, Xu, 2019
  - Muthukumar, Vodrahalli, Sahai, 2019
  - Bartlett, Long, Lugosi, Tsigler, 2019
  - Mei, Montanari, 2019
- Kernel regression
  - Liang, Rakhlin, 2018
  - Liang, Rakhlin, Zhai, 2019
- Local methods
  - Belkin, Hsu, Mitra, 2018
  - Belkin, Rakhlin, Tsybakov, 2018
- and many more ...

What do we currently understand?

- In nearly all applications, current practice suggests we should design models to be massively overparametrized
- Once trained (typically by SGD), these models interpolate the training data (achieve zero training error)
- Still they are capable of having (often do have) good test error

Current understanding of this? In full theoretical rigor, not great.

However, the story is fairly well-understood for linear models, kernel models, and random feature models. See, e.g., nice monographs:

- Bartlett, Montanari, and Rakhlin (2021), "Deep learning: a statistical viewpoint"
- Belkin (2021), "Fit without fear: remarkable mathematical phenomena of deep learning through the prism of interpolation"

What do we currently understand?

- In nearly all applications, current practice suggests we should design models to be massively overparametrized
- Once trained (typically by SGD), these models interpolate the training data (achieve zero training error)
- Still they are capable of having (often do have) good test error

Current understanding of this? In full theoretical rigor, not great.

However, the story is fairly well-understood for linear models, kernel models, and random feature models. See, e.g., nice monographs:

- Bartlett, Montanari, and Rakhlin (2021), "Deep learning: a statistical viewpoint"
- Belkin (2021), "Fit without fear: remarkable mathematical phenomena of deep learning through the prism of interpolation"

What do we currently understand?

- In nearly all applications, current practice suggests we should design models to be massively overparametrized
- Once trained (typically by SGD), these models interpolate the training data (achieve zero training error)
- Still they are capable of having (often do have) good test error

Current understanding of this? In full theoretical rigor, not great.

However, the story is fairly well-understood for linear models, kernel models, and random feature models. See, e.g., nice monographs:

- Bartlett, Montanari, and Rakhlin (2021), "Deep learning: a statistical viewpoint"
- Belkin (2021), "Fit without fear: remarkable mathematical phenomena of deep learning through the prism of interpolation"

What do we currently understand?

- In nearly all applications, current practice suggests we should design models to be massively overparametrized
- Once trained (typically by SGD), these models interpolate the training data (achieve zero training error)
- Still they are capable of having (often do have) good test error

Current understanding of this? In full theoretical rigor, not great.

However, the story is fairly well-understood for linear models, kernel models, and random feature models. See, e.g., nice monographs:

- Bartlett, Montanari, and Rakhlin (2021), "Deep learning: a statistical viewpoint"
- Belkin (2021), "Fit without fear: remarkable mathematical phenomena of deep learning through the prism of interpolation"

What do we currently understand?

- In nearly all applications, current practice suggests we should design models to be massively overparametrized
- Once trained (typically by SGD), these models interpolate the training data (achieve zero training error)
- Still they are capable of having (often do have) good test error

Current understanding of this? In full theoretical rigor, not great.

However, the story is fairly well-understood for linear models, kernel models, and random feature models. See, e.g., nice monographs:

- Bartlett, Montanari, and Rakhlin (2021), "Deep learning: a statistical viewpoint"
- Belkin (2021), "Fit without fear: remarkable mathematical phenomena of deep learning through the prism of interpolation"

What do we currently understand?

- In nearly all applications, current practice suggests we should design models to be massively overparametrized
- Once trained (typically by SGD), these models interpolate the training data (achieve zero training error)
- Still they are capable of having (often do have) good test error

Current understanding of this? In full theoretical rigor, not great.

However, the story is fairly well-understood for linear models, kernel models, and random feature models. See, e.g., nice monographs:

- Bartlett, Montanari, and Rakhlin (2021), "Deep learning: a statistical viewpoint"
- Belkin (2021), "Fit without fear: remarkable mathematical phenomena of deep learning through the prism of interpolation"

What do we currently understand?

- In nearly all applications, current practice suggests we should design models to be massively overparametrized
- Once trained (typically by SGD), these models interpolate the training data (achieve zero training error)
- Still they are capable of having (often do have) good test error

Current understanding of this? In full theoretical rigor, not great.

However, the story is fairly well-understood for linear models, kernel models, and random feature models. See, e.g., nice monographs:

- Bartlett, Montanari, and Rakhlin (2021), "Deep learning: a statistical viewpoint"
- Belkin (2021), "Fit without fear: remarkable mathematical phenomena of deep learning through the prism of interpolation"

Goals of this work

This work attempts to answer the following questions:

- Is there a good measure of model complexity for predictive models?
- How to compare model complexity of different (near) interpolators?

Takaway points

- Propose measures of model complexity that are:
  - algorithm-specific and applies for any prediction algorithm
  - produce a number between 0 and n (the number of observations)
- Two variants of model complexities are:
  - emergent model complexity that depends on the prediction algorithm as well as underlying the regression function
  - intrinsic model complexity that depends on the prediction algorithm only and its adaptability to pure noise
- Results when applied to some illustrative examples:
  - min ℓ_2/ℓ_1-norm interpolators: the complexity measures maximized when $n \approx p$ and typically decreases as p increases beyond n
  - we can reparameterize every overparameterized model into an equivalent underparameterized model in terms of risk behavior

Based on extension of ideas from optimism theory and degrees of freedom.

Takaway points

- Propose measures of model complexity that are:
  - algorithm-specific and applies for any prediction algorithm
  - produce a number between 0 and n (the number of observations)
- Two variants of model complexities are:
  - emergent model complexity that depends on the prediction algorithm as well as underlying the regression function
  - intrinsic model complexity that depends on the prediction algorithm only and its adaptability to pure noise
- Results when applied to some illustrative examples:
  - min ℓ_2/ℓ_1-norm interpolators: the complexity measures maximized when $n \approx p$ and typically decreases as p increases beyond n
  - we can reparameterize every overparameterized model into an equivalent underparameterized model in terms of risk behavior

Based on extension of ideas from optimism theory and degrees of freedom.

Takaway points

- Propose measures of model complexity that are:
  - algorithm-specific and applies for any prediction algorithm
  - produce a number between 0 and n (the number of observations)
- Two variants of model complexities are:
  - emergent model complexity that depends on the prediction algorithm as well as underlying the regression function
  - intrinsic model complexity that depends on the prediction algorithm only and its adaptability to pure noise
- Results when applied to some illustrative examples:
  - min ℓ_2/ℓ_1-norm interpolators: the complexity measures maximized when $n \approx p$ and typically decreases as p increases beyond n
  - we can reparameterize every overparameterized model into an equivalent underparameterized model in terms of risk behavior

Based on extension of ideas from optimism theory and degrees of freedom.

Takaway points

- Propose measures of model complexity that are:
  - algorithm-specific and applies for any prediction algorithm
  - produce a number between 0 and n (the number of observations)
- Two variants of model complexities are:
  - emergent model complexity that depends on the prediction algorithm as well as underlying the regression function
  - intrinsic model complexity that depends on the prediction algorithm only and its adaptability to pure noise
- Results when applied to some illustrative examples:
  - min ℓ_2/ℓ_1-norm interpolators: the complexity measures maximized when $n \approx p$ and typically decreases as p increases beyond n
  - we can reparameterize every overparameterized model into an equivalent underparameterized model in terms of risk behavior

Based on extension of ideas from optimism theory and degrees of freedom.

Takaway points

- Propose measures of model complexity that are:
  - algorithm-specific and applies for any prediction algorithm
  - produce a number between 0 and n (the number of observations)
- Two variants of model complexities are:
  - emergent model complexity that depends on the prediction algorithm as well as underlying the regression function
  - intrinsic model complexity that depends on the prediction algorithm only and its adaptability to pure noise
- Results when applied to some illustrative examples:
  - min ℓ_2/ℓ_1-norm interpolators: the complexity measures maximized when $n \approx p$ and typically decreases as p increases beyond n
  - we can reparameterize every overparameterized model into an equivalent underparameterized model in terms of risk behavior

Based on extension of ideas from optimism theory and degrees of freedom.

Takaway points

- Propose measures of model complexity that are:
  - algorithm-specific and applies for any prediction algorithm
  - produce a number between 0 and n (the number of observations)
- Two variants of model complexities are:
  - emergent model complexity that depends on the prediction algorithm as well as underlying the regression function
  - intrinsic model complexity that depends on the prediction algorithm only and its adaptability to pure noise
- Results when applied to some illustrative examples:
  - min ℓ_2/ℓ_1-norm interpolators: the complexity measures maximized when $n \approx p$ and typically decreases as p increases beyond n
  - we can reparameterize every overparameterized model into an equivalent underparameterized model in terms of risk behavior

Based on extension of ideas from optimism theory and degrees of freedom.

Takaway points

- Propose measures of model complexity that are:
  - algorithm-specific and applies for any prediction algorithm
  - produce a number between 0 and n (the number of observations)
- Two variants of model complexities are:
  - emergent model complexity that depends on the prediction algorithm as well as underlying the regression function
  - intrinsic model complexity that depends on the prediction algorithm only and its adaptability to pure noise
- Results when applied to some illustrative examples:
  - min ℓ_2/ℓ_1-norm interpolators: the complexity measures maximized when $n \approx p$ and typically decreases as p increases beyond n
  - we can reparameterize every overparameterized model into an equivalent underparameterized model in terms of risk behavior

Based on extension of ideas from optimism theory and degrees of freedom.

Outline

Degrees of freedom at large

Degrees of freedom means different things in different fields, but they're more or less similar. There is a core concept behind it.

It essentially has to do with the dimension or effective number of parameters of "something".

- In mechanics, that something = mechanical system.
- In physics and chemistry, something = physical system.
- In statistics, something = prediction procedure.

A nice commonality is that we can usually guess the degrees of freedom based on intuition, at least qualitatively.

Degrees of freedom at large

Degrees of freedom means different things in different fields, but they're more or less similar. There is a core concept behind it.

It essentially has to do with the dimension or effective number of parameters of "something".

- In mechanics, that something = mechanical system.
- In physics and chemistry, something = physical system.
- In statistics, something = prediction procedure.

A nice commonality is that we can usually guess the degrees of freedom based on intuition, at least qualitatively.

Degrees of freedom at large

Degrees of freedom means different things in different fields, but they're more or less similar. There is a core concept behind it.

It essentially has to do with the dimension or effective number of parameters of "something".

- In mechanics, that something = mechanical system.
- In physics and chemistry, something = physical system.
- In statistics, something = prediction procedure.

A nice commonality is that we can usually guess the degrees of freedom based on intuition, at least qualitatively.

Degrees of freedom at large

Degrees of freedom means different things in different fields, but they're more or less similar. There is a core concept behind it.

It essentially has to do with the dimension or effective number of parameters of "something".

- In mechanics, that something = mechanical system.
- In physics and chemistry, something = physical system.
- In statistics, something = prediction procedure.

A nice commonality is that we can usually guess the degrees of freedom based on intuition, at least qualitatively.

Degrees of freedom at large

Degrees of freedom means different things in different fields, but they're more or less similar. There is a core concept behind it.

It essentially has to do with the dimension or effective number of parameters of "something".

- In mechanics, that something = mechanical system.
- In physics and chemistry, something = physical system.
- In statistics, something = prediction procedure.

A nice commonality is that we can usually guess the degrees of freedom based on intuition, at least qualitatively.

Degrees of freedom at large

Degrees of freedom means different things in different fields, but they're more or less similar. There is a core concept behind it.

It essentially has to do with the dimension or effective number of parameters of "something".

- In mechanics, that something = mechanical system.
- In physics and chemistry, something = physical system.
- In statistics, something = prediction procedure.

A nice commonality is that we can usually guess the degrees of freedom based on intuition, at least qualitatively.

Degrees of freedom in statistics

Degrees of freedom (df) is a classical topic in statistics, dating back to Mallows (1973), Stein (1981), Efron (1986)

Given an estimator $\widehat{f}$ of the regression function (i.e., $\widehat{f}(x)$ estimates $\mathbb{E}[y|x]$), trained on (x_i, y_i), $i = 1, \ldots, n$, we define

$$\mathrm{df}(\widehat{f}) = \frac{1}{\sigma^2} \sum_{i=1}^{n} \mathrm{Cov}(y_i, \widehat{f}(x_i))$$

where $\sigma^2 = \mathrm{Var}(y|x)$, and each x_i is treated as fixed

Key fact. For $\widehat{f}$ the least squares estimator of response $Y \in \mathbb{R}^n$ on feature matrix $X \in \mathbb{R}^{n \times p}$, with p linearly independent columns,

$$\mathrm{df}(\widehat{f}) = \frac{1}{\sigma^2} \mathrm{tr}(\mathrm{Cov}(Y, P_X Y)) = \mathrm{tr}(P_X) = p$$

Degrees of freedom in statistics

Degrees of freedom (df) is a classical topic in statistics, dating back to Mallows (1973), Stein (1981), Efron (1986)

Given an estimator $\widehat{f}$ of the regression function (i.e., $\widehat{f}(x)$ estimates $\mathbb{E}[y|x]$), trained on (x_i, y_i), $i = 1, \ldots, n$, we define

$$\mathrm{df}(\widehat{f}) = \frac{1}{\sigma^2} \sum_{i=1}^{n} \mathrm{Cov}(y_i, \widehat{f}(x_i))$$

where $\sigma^2 = \mathrm{Var}(y|x)$, and each x_i is treated as fixed

Key fact. For $\widehat{f}$ the least squares estimator of response $Y \in \mathbb{R}^n$ on feature matrix $X \in \mathbb{R}^{n \times p}$, with p linearly independent columns,

$$\mathrm{df}(\widehat{f}) = \frac{1}{\sigma^2} \mathrm{tr}(\mathrm{Cov}(Y, P_X Y)) = \mathrm{tr}(P_X) = p$$

Degrees of freedom in statistics

Degrees of freedom (df) is a classical topic in statistics, dating back to Mallows (1973), Stein (1981), Efron (1986)

Given an estimator $\widehat{f}$ of the regression function (i.e., $\widehat{f}(x)$ estimates $\mathbb{E}[y|x]$), trained on (x_i, y_i), $i = 1, \ldots, n$, we define

$$\mathrm{df}(\widehat{f}) = \frac{1}{\sigma^2} \sum_{i=1}^{n} \mathrm{Cov}(y_i, \widehat{f}(x_i))$$

where $\sigma^2 = \mathrm{Var}(y|x)$, and each x_i is treated as fixed

Key fact. For $\widehat{f}$ the least squares estimator of response $Y \in \mathbb{R}^n$ on feature matrix $X \in \mathbb{R}^{n \times p}$, with p linearly independent columns,

$$\mathrm{df}(\widehat{f}) = \frac{1}{\sigma^2} \mathrm{tr}(\mathrm{Cov}(Y, P_X Y)) = \mathrm{tr}(P_X) = p$$

Degrees of freedom in statistics

Degrees of freedom (df) is a classical topic in statistics, dating back to Mallows (1973), Stein (1981), Efron (1986)

Given an estimator $\widehat{f}$ of the regression function (i.e., $\widehat{f}(x)$ estimates $\mathbb{E}[y|x]$), trained on (x_i, y_i), $i = 1, \ldots, n$, we define

$$\mathrm{df}(\widehat{f}) = \frac{1}{\sigma^2} \sum_{i=1}^{n} \mathrm{Cov}(y_i, \widehat{f}(x_i))$$

where $\sigma^2 = \mathrm{Var}(y|x)$, and each x_i is treated as fixed

Key fact. For $\widehat{f}$ the least squares estimator of response $Y \in \mathbb{R}^n$ on feature matrix $X \in \mathbb{R}^{n \times p}$, with p linearly independent columns,

$$\mathrm{df}(\widehat{f}) = \frac{1}{\sigma^2} \mathrm{tr}(\mathrm{Cov}(Y, P_X Y)) = \mathrm{tr}(P_X) = p$$

Degrees of freedom in statistics

Degrees of freedom (df) is a classical topic in statistics, dating back to Mallows (1973), Stein (1981), Efron (1986)

Given an estimator $\widehat{f}$ of the regression function (i.e., $\widehat{f}(x)$ estimates $\mathbb{E}[y|x]$), trained on (x_i, y_i), $i = 1, \ldots, n$, we define

$$\mathrm{df}(\widehat{f}) = \frac{1}{\sigma^2} \sum_{i=1}^{n} \mathrm{Cov}(y_i, \widehat{f}(x_i))$$

where $\sigma^2 = \mathrm{Var}(y|x)$, and each x_i is treated as fixed

Key fact. For $\widehat{f}$ the least squares estimator of response $Y \in \mathbb{R}^n$ on feature matrix $X \in \mathbb{R}^{n \times p}$, with p linearly independent columns,

$$\mathrm{df}(\widehat{f}) = \frac{1}{\sigma^2} \mathrm{tr}(\mathrm{Cov}(Y, P_X Y)) = \mathrm{tr}(P_X) = p$$

Degrees of freedom in statistics

Degrees of freedom (df) is a classical topic in statistics, dating back to Mallows (1973), Stein (1981), Efron (1986)

Given an estimator $\widehat{f}$ of the regression function (i.e., $\widehat{f}(x)$ estimates $\mathbb{E}[y|x]$), trained on (x_i, y_i), $i = 1, \ldots, n$, we define

$$\mathrm{df}(\widehat{f}) = \frac{1}{\sigma^2} \sum_{i=1}^{n} \mathrm{Cov}(y_i, \widehat{f}(x_i))$$

where $\sigma^2 = \mathrm{Var}(y|x)$, and each x_i is treated as fixed

Key fact. For $\widehat{f}$ the least squares estimator of response $Y \in \mathbb{R}^n$ on feature matrix $X \in \mathbb{R}^{n \times p}$, with p linearly independent columns,

$$\mathrm{df}(\widehat{f}) = \frac{1}{\sigma^2} \mathrm{tr}(\mathrm{Cov}(Y, P_X Y)) = \mathrm{tr}(P_X) = p$$

Degrees of freedom in statistics

Degrees of freedom (df) is a classical topic in statistics, dating back to Mallows (1973), Stein (1981), Efron (1986)

Given an estimator $\widehat{f}$ of the regression function (i.e., $\widehat{f}(x)$ estimates $\mathbb{E}[y|x]$), trained on (x_i, y_i), $i = 1, \ldots, n$, we define

$$\mathrm{df}(\widehat{f}) = \frac{1}{\sigma^2} \sum_{i=1}^{n} \mathrm{Cov}(y_i, \widehat{f}(x_i))$$

where $\sigma^2 = \mathrm{Var}(y|x)$, and each x_i is treated as fixed

Key fact. For $\widehat{f}$ the least squares estimator of response $Y \in \mathbb{R}^n$ on feature matrix $X \in \mathbb{R}^{n \times p}$, with p linearly independent columns,

$$\mathrm{df}(\widehat{f}) = \frac{1}{\sigma^2} \mathrm{tr}(\mathrm{Cov}(Y, P_X Y)) = \mathrm{tr}(P_X) = p$$

Connection to fixed-X error

This definition is intrinsically connected to fixed-X test error:

$$\underbrace{\mathbb{E}_{Y,Y^*}\left[\frac{1}{n}\sum_{i=1}^{n}(y_i^* - \widehat{f}(x_i))^2\right]}_{\mathrm{ErrF}(\widehat{f})} = \underbrace{\mathbb{E}_{Y}\left[\frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{f}(x_i))^2\right]}_{\mathrm{train_F}(\widehat{f})} + \frac{2\sigma^2}{n}\mathrm{df}(\widehat{f})$$

The difference between test and training error is called optimism:

$$\mathrm{OptF}(\widehat{f}) = \mathrm{ErrF}(\widehat{f}) - \mathrm{train_F}(\widehat{f})$$

This nicely motivates degrees of freedom and connects it to practice

Also points to a big shortcoming ... fixed-X error is not as relevant to modern stat/ML practice which is driven by random-X error

$$\mathrm{ErrR}(\widehat{f}) = \mathbb{E}_{X,Y,x_0,y_0}[(y_0 - \widehat{f}(x_0))^2]$$

Connection to fixed-X error

This definition is intrinsically connected to fixed-X test error:

$$\underbrace{\mathbb{E}_{Y,Y^*}\left[\frac{1}{n}\sum_{i=1}^{n}(y_i^* - \widehat{f}(x_i))^2\right]}_{\mathrm{ErrF}(\widehat{f})} = \underbrace{\mathbb{E}_{Y}\left[\frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{f}(x_i))^2\right]}_{\mathrm{train_F}(\widehat{f})} + \frac{2\sigma^2}{n}\mathrm{df}(\widehat{f})$$

The difference between test and training error is called optimism:

$$\mathrm{OptF}(\widehat{f}) = \mathrm{ErrF}(\widehat{f}) - \mathrm{train_F}(\widehat{f})$$

This nicely motivates degrees of freedom and connects it to practice

Also points to a big shortcoming ... fixed-X error is not as relevant to modern stat/ML practice which is driven by random-X error

$$\mathrm{ErrR}(\widehat{f}) = \mathbb{E}_{X,Y,x_0,y_0}[(y_0 - \widehat{f}(x_0))^2]$$

Connection to fixed-X error

This definition is intrinsically connected to fixed-X test error:

$$\underbrace{\mathbb{E}_{Y,Y^*}\left[\frac{1}{n}\sum_{i=1}^{n}(y_i^* - \widehat{f}(x_i))^2\right]}_{\mathrm{ErrF}(\widehat{f})} = \underbrace{\mathbb{E}_{Y}\left[\frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{f}(x_i))^2\right]}_{\mathrm{train_F}(\widehat{f})} + \frac{2\sigma^2}{n}\mathrm{df}(\widehat{f})$$

The difference between test and training error is called optimism:

$$\mathrm{OptF}(\widehat{f}) = \mathrm{ErrF}(\widehat{f}) - \mathrm{train_F}(\widehat{f})$$

This nicely motivates degrees of freedom and connects it to practice

Also points to a big shortcoming ... fixed-X error is not as relevant to modern stat/ML practice which is driven by random-X error

$$\mathrm{ErrR}(\widehat{f}) = \mathbb{E}_{X,Y,x_0,y_0}[(y_0 - \widehat{f}(x_0))^2]$$

Connection to fixed-X error

This definition is intrinsically connected to fixed-X test error:

$$\underbrace{\mathbb{E}_{Y,Y^*}\left[\frac{1}{n}\sum_{i=1}^{n}(y_i^* - \widehat{f}(x_i))^2\right]}_{\mathrm{ErrF}(\widehat{f})} = \underbrace{\mathbb{E}_{Y}\left[\frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{f}(x_i))^2\right]}_{\mathrm{train_F}(\widehat{f})} + \frac{2\sigma^2}{n}\mathrm{df}(\widehat{f})$$

The difference between test and training error is called optimism:

$$\mathrm{OptF}(\widehat{f}) = \mathrm{ErrF}(\widehat{f}) - \mathrm{train_F}(\widehat{f})$$

This nicely motivates degrees of freedom and connects it to practice

Also points to a big shortcoming ... fixed-X error is not as relevant to modern stat/ML practice which is driven by random-X error

$$\mathrm{ErrR}(\widehat{f}) = \mathbb{E}_{X,Y,x_0,y_0}[(y_0 - \widehat{f}(x_0))^2]$$

Connection to fixed-X error

This definition is intrinsically connected to fixed-X test error:

$$\underbrace{\mathbb{E}_{Y,Y^*}\left[\frac{1}{n}\sum_{i=1}^{n}(y_i^* - \widehat{f}(x_i))^2\right]}_{\mathrm{ErrF}(\widehat{f})} = \underbrace{\mathbb{E}_{Y}\left[\frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{f}(x_i))^2\right]}_{\mathrm{train_F}(\widehat{f})} + \frac{2\sigma^2}{n}\mathrm{df}(\widehat{f})$$

The difference between test and training error is called optimism:

$$\mathrm{OptF}(\widehat{f}) = \mathrm{ErrF}(\widehat{f}) - \mathrm{train_F}(\widehat{f})$$

This nicely motivates degrees of freedom and connects it to practice

Also points to a big shortcoming ... fixed-X error is not as relevant to modern stat/ML practice which is driven by random-X error

$$\mathrm{ErrR}(\widehat{f}) = \mathbb{E}_{X,Y,x_0,y_0}[(y_0 - \widehat{f}(x_0))^2]$$

Connection to fixed-X error

This definition is intrinsically connected to fixed-X test error:

$$\underbrace{\mathbb{E}_{Y,Y^*}\left[\frac{1}{n}\sum_{i=1}^{n}(y_i^* - \widehat{f}(x_i))^2\right]}_{\mathrm{ErrF}(\widehat{f})} = \underbrace{\mathbb{E}_{Y}\left[\frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{f}(x_i))^2\right]}_{\mathrm{train_F}(\widehat{f})} + \frac{2\sigma^2}{n}\mathrm{df}(\widehat{f})$$

The difference between test and training error is called optimism:

$$\mathrm{OptF}(\widehat{f}) = \mathrm{ErrF}(\widehat{f}) - \mathrm{train_F}(\widehat{f})$$

This nicely motivates degrees of freedom and connects it to practice

Also points to a big shortcoming ... fixed-X error is not as relevant to modern stat/ML practice which is driven by random-X error

$$\mathrm{ErrR}(\widehat{f}) = \mathbb{E}_{X,Y,x_0,y_0}[(y_0 - \widehat{f}(x_0))^2]$$

Fixed-X $\neq$ random-X, especially for interpolators

Classically—for smooth functions, in low dimensions—we can tie fixed-X and random-X errors together (e.g., by empirical process theory)

But beyond this—nonsmooth functions, or high dimensions—they can be very different.
Epitomized by generalizing interpolators:

- $\mathrm{ErrR}(\widehat{f}) \to 0$, but
- $\mathrm{ErrF}(\widehat{f}) = \mathbb{E}[n^{-1} \sum_{i=1}^{n} (y_i^* - y_i)^2] = 2\sigma^2$

Similarly, for any interpolator $\widehat{f}$:

$$\mathrm{df}(\widehat{f}) = \frac{1}{\sigma^2} \sum_{i=1}^{n} \mathrm{Cov}(y_i, y_i) = n$$

which is useless!

Fixed-X $\neq$ random-X, especially for interpolators

Classically—for smooth functions, in low dimensions—we can tie fixed-X and random-X errors together (e.g., by empirical process theory)

But beyond this—nonsmooth functions, or high dimensions—they can be very different.

Epitomized by generalizing interpolators:

- $\mathrm{ErrR}(\widehat{f}) \to 0$, but
- $\mathrm{ErrF}(\widehat{f}) = \mathbb{E}[n^{-1} \sum_{i=1}^{n} (y_i^* - y_i)^2] = 2\sigma^2$

Similarly, for any interpolator $\widehat{f}$:

$$\mathrm{df}(\widehat{f}) = \frac{1}{\sigma^2} \sum_{i=1}^{n} \mathrm{Cov}(y_i, y_i) = n$$

which is useless!

Fixed-X $\neq$ random-X, especially for interpolators

Classically—for smooth functions, in low dimensions—we can tie fixed-X and random-X errors together (e.g., by empirical process theory)

But beyond this—nonsmooth functions, or high dimensions—they can be very different.
Epitomized by generalizing interpolators:

- $\mathrm{ErrR}(\widehat{f}) \to 0$, but
- $\mathrm{ErrF}(\widehat{f}) = \mathbb{E}[n^{-1} \sum_{i=1}^{n} (y_i^* - y_i)^2] = 2\sigma^2$

Similarly, for any interpolator $\widehat{f}$:

$$\mathrm{df}(\widehat{f}) = \frac{1}{\sigma^2} \sum_{i=1}^{n} \mathrm{Cov}(y_i, y_i) = n$$

which is useless!

Fixed-X $\neq$ random-X, especially for interpolators

Classically—for smooth functions, in low dimensions—we can tie fixed-X and random-X errors together (e.g., by empirical process theory)

But beyond this—nonsmooth functions, or high dimensions—they can be very different.
Epitomized by generalizing interpolators:

- $\mathrm{ErrR}(\widehat{f}) \to 0$, but
- $\mathrm{ErrF}(\widehat{f}) = \mathbb{E}[n^{-1}\sum_{i=1}^{n}(y_i^* - y_i)^2] = 2\sigma^2$

Similarly, for any interpolator $\widehat{f}$:

$$\mathrm{df}(\widehat{f}) = \frac{1}{\sigma^2}\sum_{i=1}^{n}\mathrm{Cov}(y_i, y_i) = n$$

which is useless!

Fixed-X $\neq$ random-X, especially for interpolators

Classically—for smooth functions, in low dimensions—we can tie fixed-X and random-X errors together (e.g., by empirical process theory)

But beyond this—nonsmooth functions, or high dimensions—they can be very different.
Epitomized by generalizing interpolators:

- $\mathrm{ErrR}(\widehat{f}) \to 0$, but
- $\mathrm{ErrF}(\widehat{f}) = \mathbb{E}[n^{-1} \sum_{i=1}^{n} (y_i^* - y_i)^2] = 2\sigma^2$

Similarly, for any interpolator $\widehat{f}$:

$$\mathrm{df}(\widehat{f}) = \frac{1}{\sigma^2} \sum_{i=1}^{n} \mathrm{Cov}(y_i, y_i) = n$$

which is useless!

Fixed-X $\neq$ random-X, especially for interpolators

Classically—for smooth functions, in low dimensions—we can tie fixed-X and random-X errors together (e.g., by empirical process theory)

But beyond this—nonsmooth functions, or high dimensions—they can be very different.
Epitomized by generalizing interpolators:

- $\mathrm{ErrR}(\widehat{f}) \to 0$, but
- $\mathrm{ErrF}(\widehat{f}) = \mathbb{E}[n^{-1} \sum_{i=1}^{n} (y_i^* - y_i)^2] = 2\sigma^2$

Similarly, for any interpolator $\widehat{f}$:

$$\mathrm{df}(\widehat{f}) = \frac{1}{\sigma^2} \sum_{i=1}^{n} \mathrm{Cov}(y_i, y_i) = n$$

which is useless!

Fixed-X $\neq$ random-X, especially for interpolators

Classically—for smooth functions, in low dimensions—we can tie fixed-X and random-X errors together (e.g., by empirical process theory)

But beyond this—nonsmooth functions, or high dimensions—they can be very different.
Epitomized by generalizing interpolators:

- $\mathrm{ErrR}(\widehat{f}) \to 0$, but
- $\mathrm{ErrF}(\widehat{f}) = \mathbb{E}[n^{-1} \sum_{i=1}^{n} (y_i^* - y_i)^2] = 2\sigma^2$

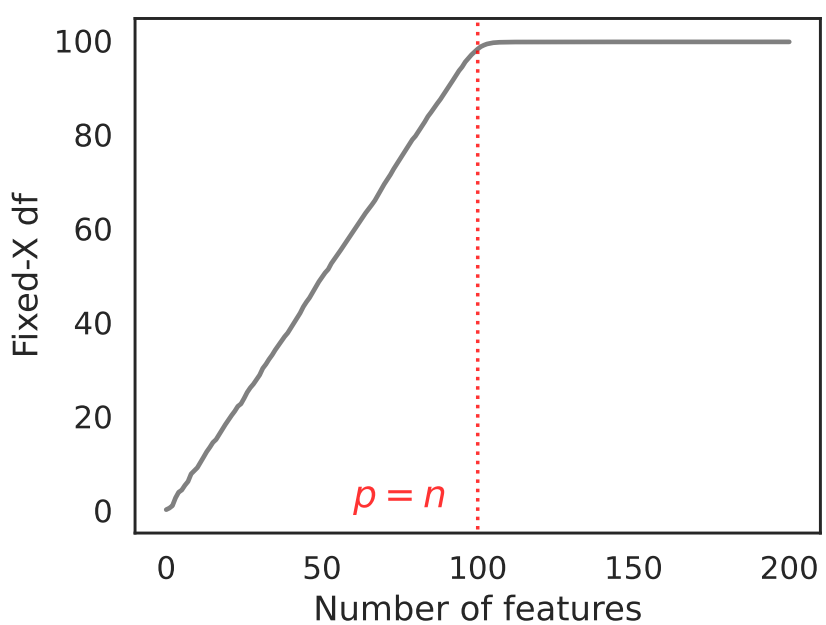
Similarly, for any interpolator $\widehat{f}$:

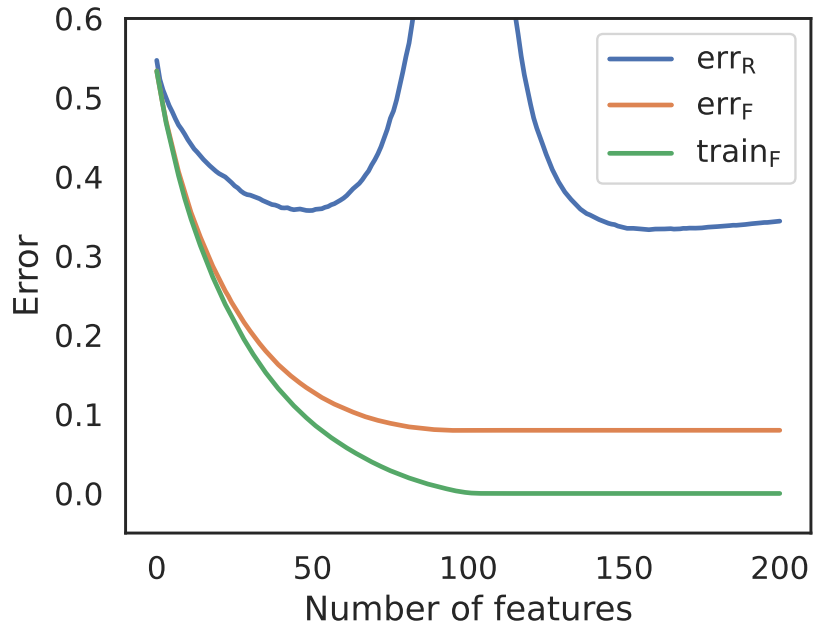
$$\mathrm{df}(\widehat{f}) = \frac{1}{\sigma^2} \sum_{i=1}^{n} \mathrm{Cov}(y_i, y_i) = n$$

which is useless!

Example: ridgeless least squares df

Classical, fixed-X df for ridgeless least squares, with $n = 100$:





Can we generalize df to random-X, and beyond?

Goals:

- Generalize df so that it connects to random-X error, and gives meaningful answers for any estimator (even interpolators)
- Allow for decomposition of df according to some user-specified components

Disclaimer: *what follows is embarassingly simple ...*

Can we generalize df to random-X, and beyond?

Goals:

- Generalize df so that it connects to random-X error, and gives meaningful answers for any estimator (even interpolators)
- Allow for decomposition of df according to some user-specified components

Disclaimer: *what follows is embarassingly simple ...*

Can we generalize df to random-X, and beyond?

Goals:

- Generalize df so that it connects to random-X error, and gives meaningful answers for any estimator (even interpolators)
- Allow for decomposition of df according to some user-specified components

Disclaimer: *what follows is embarassingly simple ...*

Can we generalize df to random-X, and beyond?

Goals:

- Generalize df so that it connects to random-X error, and gives meaningful answers for any estimator (even interpolators)
- Allow for decomposition of df according to some user-specified components

Disclaimer: *what follows is embarassingly simple ...*

Outline

What is complexity?

What makes up a measure of complexity? Two things:

- a metric
- a reference model

That is, the metric assigns a number to the given model $\widehat{f}$, and the reference model $\widehat{f}^{\text{ref}}$ provides units, so we can interpret the metric

A bit more detail:

- Metric should be "negatively oriented" for complexity—smaller values mean less complex
- Reference model should be something whose parameters we are "happy to count"

What is complexity?

What makes up a measure of complexity? Two things:

- a metric
- a reference model

That is, the metric assigns a number to the given model $\widehat{f}$, and the reference model $\widehat{f}^{\text{ref}}$ provides units, so we can interpret the metric

A bit more detail:

- Metric should be "negatively oriented" for complexity—smaller values mean less complex
- Reference model should be something whose parameters we are "happy to count"

What is complexity?

What makes up a measure of complexity? Two things:

- a metric
- a reference model

That is, the metric assigns a number to the given model $\widehat{f}$, and the reference model $\widehat{f}^{\text{ref}}$ provides units, so we can interpret the metric

A bit more detail:

- Metric should be "negatively oriented" for complexity—smaller values mean less complex
- Reference model should be something whose parameters we are "happy to count"

What is complexity?

What makes up a measure of complexity? Two things:

- a metric
- a reference model

That is, the metric assigns a number to the given model $\widehat{f}$, and the reference model $\widehat{f}^{\text{ref}}$ provides units, so we can interpret the metric

A bit more detail:

- Metric should be "negatively oriented" for complexity—smaller values mean less complex
- Reference model should be something whose parameters we are "happy to count"

What is complexity?

What makes up a measure of complexity? Two things:

- a metric
- a reference model

That is, the metric assigns a number to the given model $\widehat{f}$, and the reference model $\widehat{f}^{\text{ref}}$ provides units, so we can interpret the metric

A bit more detail:

- Metric should be "negatively oriented" for complexity—smaller values mean less complex
- Reference model should be something whose parameters we are "happy to count"

What is complexity?

What makes up a measure of complexity? Two things:

- a metric
- a reference model

That is, the metric assigns a number to the given model $\widehat{f}$, and the reference model $\widehat{f}^{\text{ref}}$ provides units, so we can interpret the metric

A bit more detail:

- Metric should be "negatively oriented" for complexity—smaller values mean less complex
- Reference model should be something whose parameters we are "happy to count"

What is complexity?

What makes up a measure of complexity? Two things:

- a metric
- a reference model

That is, the metric assigns a number to the given model $\widehat{f}$, and the reference model $\widehat{f}^{\text{ref}}$ provides units, so we can interpret the metric

A bit more detail:

- Metric should be "negatively oriented" for complexity—smaller values mean less complex
- Reference model should be something whose parameters we are "happy to count"

What is complexity?

What makes up a measure of complexity? Two things:

- a metric
- a reference model

That is, the metric assigns a number to the given model $\widehat{f}$, and the reference model $\widehat{f}^{\text{ref}}$ provides units, so we can interpret the metric

A bit more detail:

- Metric should be "negatively oriented" for complexity—smaller values mean less complex
- Reference model should be something whose parameters we are "happy to count"

What is complexity?

What makes up a measure of complexity? Two things:

- a metric
- a reference model

That is, the metric assigns a number to the given model $\widehat{f}$, and the reference model $\widehat{f}^{\text{ref}}$ provides units, so we can interpret the metric

A bit more detail:

- Metric should be "negatively oriented" for complexity—smaller values mean less complex
- Reference model should be something whose parameters we are "happy to count"

Re-interpreting degrees of freedom

We can actually re-interpret the classical definition of df in this light, with: metric $= \mathrm{OptF}$, and $\widehat{f}^{\mathrm{ref}} =$ least squares

That is, *we will define* $\mathrm{df}(\widehat{f}) = k$, for the number $k \geq 0$ such that

$$\mathrm{OptF}(\widehat{f}) = \mathrm{OptF}(\widehat{f}_k^{\mathrm{ls}})$$

where $\widehat{f}_k^{\mathrm{ls}}$ is the least squares estimator of the given $Y \in \mathbb{R}^n$ on a feature matrix $X \in \mathbb{R}^{n \times k}$ with k linearly independent columns

Think about it this way: $\mathrm{df}(\widehat{f}) = 5.3$ means $\widehat{f}$ has the same fixed-X optimism as least squares with 5.3 parameters

Re-interpreting degrees of freedom

We can actually re-interpret the classical definition of df in this light, with: metric $= \mathrm{OptF}$, and $\widehat{f}^{\mathrm{ref}} =$ least squares

That is, *we will define* $\mathrm{df}(\widehat{f}) = k$, for the number $k \geq 0$ such that

$$\mathrm{OptF}(\widehat{f}) = \mathrm{OptF}(\widehat{f}_k^{\mathrm{ls}})$$

where $\widehat{f}_k^{\mathrm{ls}}$ is the least squares estimator of the given $Y \in \mathbb{R}^n$ on a feature matrix $X \in \mathbb{R}^{n \times k}$ with k linearly independent columns

Think about it this way: $\mathrm{df}(\widehat{f}) = 5.3$ means $\widehat{f}$ has the same fixed-X optimism as least squares with 5.3 parameters

Re-interpreting degrees of freedom

We can actually re-interpret the classical definition of df in this light, with: metric $= \mathrm{OptF}$, and $\widehat{f}^{\mathrm{ref}} =$ least squares

That is, *we will define* $\mathrm{df}(\widehat{f}) = k$, for the number $k \geq 0$ such that

$$\mathrm{OptF}(\widehat{f}) = \mathrm{OptF}(\widehat{f}_k^{\mathrm{ls}})$$

where $\widehat{f}_k^{\mathrm{ls}}$ is the least squares estimator of the given $Y \in \mathbb{R}^n$ on a feature matrix $X \in \mathbb{R}^{n \times k}$ with k linearly independent columns

Think about it this way: $\mathrm{df}(\widehat{f}) = 5.3$ means $\widehat{f}$ has the same fixed-X optimism as least squares with 5.3 parameters

Re-interpreting degrees of freedom

We can actually re-interpret the classical definition of df in this light, with: metric $= \mathrm{OptF}$, and $\widehat{f}^{\mathrm{ref}} =$ least squares

That is, *we will define* $\mathrm{df}(\widehat{f}) = k$, for the number $k \geq 0$ such that

$$\mathrm{OptF}(\widehat{f}) = \mathrm{OptF}(\widehat{f}_k^{\mathrm{ls}})$$

where $\widehat{f}_k^{\mathrm{ls}}$ is the least squares estimator of the given $Y \in \mathbb{R}^n$ on a feature matrix $X \in \mathbb{R}^{n \times k}$ with k linearly independent columns

Think about it this way: $\mathrm{df}(\widehat{f}) = 5.3$ means $\widehat{f}$ has the same fixed-X optimism as least squares with 5.3 parameters

Re-interpreting degrees of freedom

We can actually re-interpret the classical definition of df in this light, with: metric $= \mathrm{OptF}$, and $\widehat{f}^{\mathrm{ref}} =$ least squares

That is, *we will define* $\mathrm{df}(\widehat{f}) = k$, for the number $k \geq 0$ such that

$$\mathrm{OptF}(\widehat{f}) = \mathrm{OptF}(\widehat{f}_k^{\mathrm{ls}})$$

where $\widehat{f}_k^{\mathrm{ls}}$ is the least squares estimator of the given $Y \in \mathbb{R}^n$ on a feature matrix $X \in \mathbb{R}^{n \times k}$ with k linearly independent columns

Think about it this way: $\mathrm{df}(\widehat{f}) = 5.3$ means $\widehat{f}$ has the same fixed-X optimism as least squares with 5.3 parameters

Re-interpreting degrees of freedom

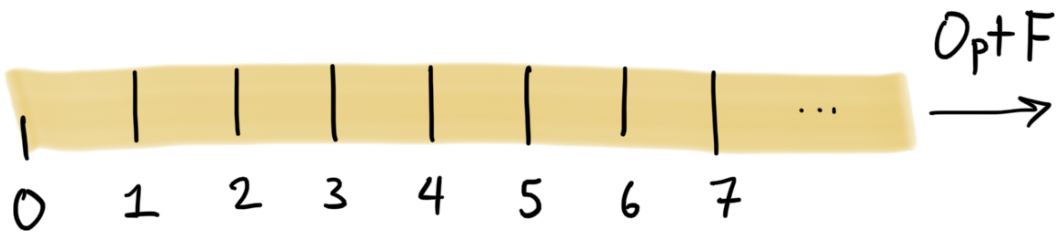
We can actually re-interpret the classical definition of df in this light, with: metric $= \mathrm{OptF}$, and $\widehat{f}^{\mathrm{ref}} =$ least squares

That is, *we will define* $\mathrm{df}(\widehat{f}) = k$, for the number $k \geq 0$ such that

$$\mathrm{OptF}(\widehat{f}) = \mathrm{OptF}(\widehat{f}_k^{\mathrm{ls}})$$

where $\widehat{f}_k^{\mathrm{ls}}$ is the least squares estimator of the given $Y \in \mathbb{R}^n$ on a feature matrix $X \in \mathbb{R}^{n \times k}$ with k linearly independent columns

Think about it this way: $\mathrm{df}(\widehat{f}) = 5.3$ means $\widehat{f}$ has the same fixed-X optimism as least squares with 5.3 parameters



Lifting this idea to random-X

Naturally we can lift this to the random-X setting. We define

$$\mathrm{OptR}(\widehat{f}) = \underbrace{\mathbb{E}_{X,Y,x_0,y_0}[(y_0 - \widehat{f}(x_0))^2]}_{\mathrm{ErrR}(\widehat{f})} - \underbrace{\mathbb{E}_{X,Y}\left[\frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{f}(x_i))^2\right]}_{\mathrm{train_R}(\widehat{f})}$$

We will again use $\widehat{f}_k^{\mathrm{ls}}$ for reference class: least squares on k features. However, in the random-X setting, it matters:

- what these features are (law of x_i)
- how they relate to the response (law of $y_i|x_i$)

(Recall, none of this mattered in the fixed-X setting ...)

So what data generating distribution should we use for our reference model in the random-X setting?

Lifting this idea to random-X

Naturally we can lift this to the random-X setting. We define

$$\mathrm{OptR}(\widehat{f}) = \underbrace{\mathbb{E}_{X,Y,x_0,y_0}[(y_0 - \widehat{f}(x_0))^2]}_{\mathrm{ErrR}(\widehat{f})} - \underbrace{\mathbb{E}_{X,Y}\left[\frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{f}(x_i))^2\right]}_{\mathrm{train}_{\mathrm{R}}(\widehat{f})}$$

We will again use $\widehat{f}_k^{\mathrm{ls}}$ for reference class: least squares on k features. However, in the random-X setting, it matters:

- what these features are (law of x_i)
- how they relate to the response (law of $y_i|x_i$)

(Recall, none of this mattered in the fixed-X setting ...)

So what data generating distribution should we use for our reference model in the random-X setting?

Lifting this idea to random-X

Naturally we can lift this to the random-X setting. We define

$$\mathrm{OptR}(\widehat{f}) = \underbrace{\mathbb{E}_{X,Y,x_0,y_0}[(y_0 - \widehat{f}(x_0))^2]}_{\mathrm{ErrR}(\widehat{f})} - \underbrace{\mathbb{E}_{X,Y}\left[\frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{f}(x_i))^2\right]}_{\mathrm{train}_{\mathrm{R}}(\widehat{f})}$$

We will again use $\widehat{f}_k^{\mathrm{ls}}$ for reference class: least squares on k features.

However, in the random-X setting, it matters:

- what these features are (law of x_i)
- how they relate to the response (law of $y_i|x_i$)

(Recall, none of this mattered in the fixed-X setting ...)

So what data generating distribution should we use for our reference model in the random-X setting?

Lifting this idea to random-X

Naturally we can lift this to the random-X setting. We define

$$\mathrm{OptR}(\widehat{f}) = \underbrace{\mathbb{E}_{X,Y,x_0,y_0}[(y_0 - \widehat{f}(x_0))^2]}_{\mathrm{ErrR}(\widehat{f})} - \underbrace{\mathbb{E}_{X,Y}\left[\frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{f}(x_i))^2\right]}_{\mathrm{train}_{\mathrm{R}}(\widehat{f})}$$

We will again use $\widehat{f}_k^{\mathrm{ls}}$ for reference class: least squares on k features. However, in the random-X setting, it matters:

- what these features are (law of x_i)
- how they relate to the response (law of $y_i|x_i$)

(Recall, none of this mattered in the fixed-X setting ...)

So what data generating distribution should we use for our reference model in the random-X setting?

Lifting this idea to random-X

Naturally we can lift this to the random-X setting. We define

$$\mathrm{OptR}(\widehat{f}) = \underbrace{\mathbb{E}_{X,Y,x_0,y_0}[(y_0 - \widehat{f}(x_0))^2]}_{\mathrm{ErrR}(\widehat{f})} - \underbrace{\mathbb{E}_{X,Y}\left[\frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{f}(x_i))^2\right]}_{\mathrm{train_R}(\widehat{f})}$$

We will again use $\widehat{f}_k^{\mathrm{ls}}$ for reference class: least squares on k features. However, in the random-X setting, it matters:

- what these features are (law of x_i)
- how they relate to the response (law of $y_i|x_i$)

(Recall, none of this mattered in the fixed-X setting ...)

So what data generating distribution should we use for our reference model in the random-X setting?

Lifting this idea to random-X

Naturally we can lift this to the random-X setting. We define

$$\mathrm{OptR}(\widehat{f}) = \underbrace{\mathbb{E}_{X,Y,x_0,y_0}[(y_0 - \widehat{f}(x_0))^2]}_{\mathrm{ErrR}(\widehat{f})} - \underbrace{\mathbb{E}_{X,Y}\left[\frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{f}(x_i))^2\right]}_{\mathrm{train}_{\mathrm{R}}(\widehat{f})}$$

We will again use $\widehat{f}_k^{\mathrm{ls}}$ for reference class: least squares on k features. However, in the random-X setting, it matters:

- what these features are (law of x_i)
- how they relate to the response (law of $y_i|x_i$)

(Recall, none of this mattered in the fixed-X setting ...)

So what data generating distribution should we use for our reference model in the random-X setting?

Lifting this idea to random-X

Naturally we can lift this to the random-X setting. We define

$$\mathrm{OptR}(\widehat{f}) = \underbrace{\mathbb{E}_{X,Y,x_0,y_0}[(y_0 - \widehat{f}(x_0))^2]}_{\mathrm{ErrR}(\widehat{f})} - \underbrace{\mathbb{E}_{X,Y}\left[\frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{f}(x_i))^2\right]}_{\mathrm{train}_{\mathrm{R}}(\widehat{f})}$$

We will again use $\widehat{f}_k^{\mathrm{ls}}$ for reference class: least squares on k features. However, in the random-X setting, it matters:

- what these features are (law of x_i)
- how they relate to the response (law of $y_i|x_i$)

(Recall, none of this mattered in the fixed-X setting ...)

So what data generating distribution should we use for our reference model in the random-X setting?

Lifting this idea to random-X

Naturally we can lift this to the random-X setting. We define

$$\mathrm{OptR}(\widehat{f}) = \underbrace{\mathbb{E}_{X,Y,x_0,y_0}[(y_0 - \widehat{f}(x_0))^2]}_{\mathrm{ErrR}(\widehat{f})} - \underbrace{\mathbb{E}_{X,Y}\left[\frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{f}(x_i))^2\right]}_{\mathrm{train_R}(\widehat{f})}$$

We will again use $\widehat{f}_k^{\mathrm{ls}}$ for reference class: least squares on k features. However, in the random-X setting, it matters:

- what these features are (law of x_i)
- how they relate to the response (law of $y_i|x_i$)

(Recall, none of this mattered in the fixed-X setting ...)

So what data generating distribution should we use for our reference model in the random-X setting?

Universal asymptotics

Consider i.i.d. samples (x_i, y_i), $i = 0, \ldots, n+1$, from the following standard random matrix theory (RMT) model:

- $x_i = \Sigma^{1/2} z_i$, where $\Sigma \in \mathbb{R}^{p \times p}$ is deterministic with eigenvalues bounded away from 0 and ∞; and $z_i \in \mathbb{R}^p$ has i.i.d. coordinates with zero mean, unit variance, and finite 4th moment
- $y_i = x_i^\top \beta + \epsilon_i$, where ϵ_i has zero mean, unit variance, and is independent of x_i

Theorem. Under the standard RMT model above, as $n, p \to \infty$ with $p/n \to \gamma < 1$, the random-X optimism least squares satisfies:[1]

$$\mathrm{OptR}(\widehat{f}_p^{\mathrm{ls}}) - \sigma^2 \left(\frac{n}{n-p} - \frac{n-p}{n} \right) \to 0$$

[1] Convergence is actually almost sure wrt training features $X \in \mathbb{R}^{n \times p}$; need uniform integrability to get convergence in expectation.

Universal asymptotics

Consider i.i.d. samples (x_i, y_i), $i = 0, \ldots, n+1$, from the following standard random matrix theory (RMT) model:

- $x_i = \Sigma^{1/2} z_i$, where $\Sigma \in \mathbb{R}^{p \times p}$ is deterministic with eigenvalues bounded away from 0 and ∞; and $z_i \in \mathbb{R}^p$ has i.i.d. coordinates with zero mean, unit variance, and finite 4th moment
- $y_i = x_i^\top \beta + \epsilon_i$, where ϵ_i has zero mean, unit variance, and is independent of x_i

Theorem. Under the standard RMT model above, as $n, p \to \infty$ with $p/n \to \gamma < 1$, the random-X optimism least squares satisfies:[1]

$$\mathrm{OptR}(\widehat{f}_p^{\mathrm{ls}}) - \sigma^2 \left(\frac{n}{n-p} - \frac{n-p}{n} \right) \to 0$$

[1] Convergence is actually almost sure wrt training features $X \in \mathbb{R}^{n \times p}$; need uniform integrability to get convergence in expectation.

Universal asymptotics

Consider i.i.d. samples (x_i, y_i), $i = 0, \ldots, n+1$, from the following standard random matrix theory (RMT) model:

- $x_i = \Sigma^{1/2} z_i$, where $\Sigma \in \mathbb{R}^{p \times p}$ is deterministic with eigenvalues bounded away from 0 and ∞; and $z_i \in \mathbb{R}^p$ has i.i.d. coordinates with zero mean, unit variance, and finite 4th moment
- $y_i = x_i^\top \beta + \epsilon_i$, where ϵ_i has zero mean, unit variance, and is independent of x_i

Theorem. Under the standard RMT model above, as $n, p \to \infty$ with $p/n \to \gamma < 1$, the random-X optimism least squares satisfies:[1]

$$\mathrm{OptR}(\widehat{f}_p^{\mathrm{ls}}) - \sigma^2 \left(\frac{n}{n-p} - \frac{n-p}{n} \right) \to 0$$

[1] Convergence is actually almost sure wrt training features $X \in \mathbb{R}^{n \times p}$; need uniform integrability to get convergence in expectation.

Universal asymptotics

Consider i.i.d. samples (x_i, y_i), $i = 0, \ldots, n+1$, from the following standard random matrix theory (RMT) model:

- $x_i = \Sigma^{1/2} z_i$, where $\Sigma \in \mathbb{R}^{p \times p}$ is deterministic with eigenvalues bounded away from 0 and ∞; and $z_i \in \mathbb{R}^p$ has i.i.d. coordinates with zero mean, unit variance, and finite 4th moment
- $y_i = x_i^\top \beta + \epsilon_i$, where ϵ_i has zero mean, unit variance, and is independent of x_i

Theorem. Under the standard RMT model above, as $n, p \to \infty$ with $p/n \to \gamma < 1$, the random-X optimism least squares satisfies:[1]

$$\mathrm{OptR}(\widehat{f}_p^{\mathrm{ls}}) - \sigma^2 \left(\frac{n}{n-p} - \frac{n-p}{n} \right) \to 0$$

[1] Convergence is actually almost sure wrt training features $X \in \mathbb{R}^{n \times p}$; need uniform integrability to get convergence in expectation.

Universal asymptotics

Consider i.i.d. samples (x_i, y_i), $i = 0, \ldots, n+1$, from the following standard random matrix theory (RMT) model:

- $x_i = \Sigma^{1/2} z_i$, where $\Sigma \in \mathbb{R}^{p \times p}$ is deterministic with eigenvalues bounded away from 0 and ∞; and $z_i \in \mathbb{R}^p$ has i.i.d. coordinates with zero mean, unit variance, and finite 4th moment
- $y_i = x_i^\top \beta + \epsilon_i$, where ϵ_i has zero mean, unit variance, and is independent of x_i

Theorem. Under the standard RMT model above, as $n, p \to \infty$ with $p/n \to \gamma < 1$, the random-X optimism least squares satisfies:[1]

$$\mathrm{OptR}(\widehat{f}_p^{\mathrm{ls}}) - \sigma^2 \left(\frac{n}{n-p} - \frac{n-p}{n} \right) \to 0$$

[1] Convergence is actually almost sure wrt training features $X \in \mathbb{R}^{n \times p}$; need uniform integrability to get convergence in expectation.

Emergent random-X df

Using this we can define a quantity we call emergent random-X df: $\mathrm{df}^{\mathrm{e}}(\widehat{f}) = k$, for the number $k \geq 0$ such that

$$\mathrm{OptR}(\widehat{f}; \{\text{original data}\}) = \mathrm{OptR}(\widehat{f}_k^{\mathrm{ls}}; \{\text{RMT data}\})$$

- LHS: random-X opt of the given model $\widehat{f}$ on the original data
- RHS: random-X opt of least squares $\widehat{f}_k^{\mathrm{ls}}$ on "RMT data", i.e., $x_i = \Sigma^{1/2} z_i$, $y_i = x_i^\top \beta + \epsilon_i$, as before
- RHS admits simple asymptotic approximation: $\sigma^2(\frac{n}{n-k} - \frac{n-k}{n})$

We call this "emergent" random-X df as it refers to the complexity that "emerges" from the optimism of $\widehat{f}$ on data at hand. The LHS generally has both bias and variance components, whereas the RHS is pure variance

Emergent random-X df

Using this we can define a quantity we call emergent random-X df: $\mathrm{df}^{\mathrm{e}}(\widehat{f}) = k$, for the number $k \geq 0$ such that

$$\mathrm{OptR}(\widehat{f}; \{\text{original data}\}) = \mathrm{OptR}(\widehat{f}_k^{\mathrm{ls}}; \{\text{RMT data}\})$$

- LHS: random-X opt of the given model $\widehat{f}$ on the original data
- RHS: random-X opt of least squares $\widehat{f}_k^{\mathrm{ls}}$ on "RMT data", i.e., $x_i = \Sigma^{1/2} z_i$, $y_i = x_i^\top \beta + \epsilon_i$, as before
- RHS admits simple asymptotic approximation: $\sigma^2(\frac{n}{n-k} - \frac{n-k}{n})$

We call this "emergent" random-X df as it refers to the complexity that "emerges" from the optimism of $\widehat{f}$ on data at hand. The LHS generally has both bias and variance components, whereas the RHS is pure variance

Emergent random-X df

Using this we can define a quantity we call emergent random-X df: $\mathrm{df}^{\mathrm{e}}(\widehat{f}) = k$, for the number $k \geq 0$ such that

$$\mathrm{OptR}(\widehat{f}; \{\text{original data}\}) = \mathrm{OptR}(\widehat{f}_k^{\mathrm{ls}}; \{\text{RMT data}\})$$

- LHS: random-X opt of the given model $\widehat{f}$ on the original data
- RHS: random-X opt of least squares $\widehat{f}_k^{\mathrm{ls}}$ on "RMT data", i.e., $x_i = \Sigma^{1/2} z_i$, $y_i = x_i^\top \beta + \epsilon_i$, as before
- RHS admits simple asymptotic approximation: $\sigma^2(\frac{n}{n-k} - \frac{n-k}{n})$

We call this "emergent" random-X df as it refers to the complexity that "emerges" from the optimism of $\widehat{f}$ on data at hand. The LHS generally has both bias and variance components, whereas the RHS is pure variance

Emergent random-X df

Using this we can define a quantity we call emergent random-X df: $\mathrm{df}^{\mathrm{e}}(\widehat{f}) = k$, for the number $k \geq 0$ such that

$$\mathrm{OptR}(\widehat{f}; \{\text{original data}\}) = \mathrm{OptR}(\widehat{f}_k^{\mathrm{ls}}; \{\text{RMT data}\})$$

- LHS: random-X opt of the given model $\widehat{f}$ on the original data
- RHS: random-X opt of least squares $\widehat{f}_k^{\mathrm{ls}}$ on "RMT data", i.e., $x_i = \Sigma^{1/2} z_i$, $y_i = x_i^\top \beta + \epsilon_i$, as before
- RHS admits simple asymptotic approximation: $\sigma^2(\frac{n}{n-k} - \frac{n-k}{n})$

We call this "emergent" random-X df as it refers to the complexity that "emerges" from the optimism of $\widehat{f}$ on data at hand. The LHS generally has both bias and variance components, whereas the RHS is pure variance

Emergent random-X df

Using this we can define a quantity we call emergent random-X df: $\mathrm{df}^{\mathrm{e}}(\widehat{f}) = k$, for the number $k \geq 0$ such that

$$\mathrm{OptR}(\widehat{f}; \{\text{original data}\}) = \mathrm{OptR}(\widehat{f}_k^{\mathrm{ls}}; \{\text{RMT data}\})$$

- LHS: random-X opt of the given model $\widehat{f}$ on the original data
- RHS: random-X opt of least squares $\widehat{f}_k^{\mathrm{ls}}$ on "RMT data", i.e., $x_i = \Sigma^{1/2} z_i$, $y_i = x_i^\top \beta + \epsilon_i$, as before
- RHS admits simple asymptotic approximation: $\sigma^2(\frac{n}{n-k} - \frac{n-k}{n})$

We call this "emergent" random-X df as it refers to the complexity that "emerges" from the optimism of $\widehat{f}$ on data at hand. The LHS generally has both bias and variance components, whereas the RHS is pure variance

Emergent random-X df

Using this we can define a quantity we call emergent random-X df: $\mathrm{df}^{\mathrm{e}}(\widehat{f}) = k$, for the number $k \geq 0$ such that

$$\mathrm{OptR}(\widehat{f}; \{\text{original data}\}) = \mathrm{OptR}(\widehat{f}_k^{\mathrm{ls}}; \{\text{RMT data}\})$$

- LHS: random-X opt of the given model $\widehat{f}$ on the original data
- RHS: random-X opt of least squares $\widehat{f}_k^{\mathrm{ls}}$ on "RMT data", i.e., $x_i = \Sigma^{1/2} z_i$, $y_i = x_i^\top \beta + \epsilon_i$, as before
- RHS admits simple asymptotic approximation: $\sigma^2(\frac{n}{n-k} - \frac{n-k}{n})$

We call this "emergent" random-X df as it refers to the complexity that "emerges" from the optimism of $\widehat{f}$ on data at hand. The LHS generally has both bias and variance components, whereas the RHS is pure variance

Emergent random-X df

Using this we can define a quantity we call emergent random-X df: $\mathrm{df}^{\mathrm{e}}(\widehat{f}) = k$, for the number $k \geq 0$ such that

$$\mathrm{OptR}(\widehat{f}; \{\text{original data}\}) = \mathrm{OptR}(\widehat{f}_k^{\mathrm{ls}}; \{\text{RMT data}\})$$

- LHS: random-X opt of the given model $\widehat{f}$ on the original data
- RHS: random-X opt of least squares $\widehat{f}_k^{\mathrm{ls}}$ on "RMT data", i.e., $x_i = \Sigma^{1/2} z_i$, $y_i = x_i^\top \beta + \epsilon_i$, as before
- RHS admits simple asymptotic approximation: $\sigma^2(\frac{n}{n-k} - \frac{n-k}{n})$

We call this "emergent" random-X df as it refers to the complexity that "emerges" from the optimism of $\widehat{f}$ on data at hand. The LHS generally has both bias and variance components, whereas the RHS is pure variance

Emergent random-X df

Using this we can define a quantity we call emergent random-X df: $\mathrm{df}^{\mathrm{e}}(\widehat{f}) = k$, for the number $k \geq 0$ such that

$$\mathrm{OptR}(\widehat{f}; \{\text{original data}\}) = \mathrm{OptR}(\widehat{f}_k^{\mathrm{ls}}; \{\text{RMT data}\})$$

- LHS: random-X opt of the given model $\widehat{f}$ on the original data
- RHS: random-X opt of least squares $\widehat{f}_k^{\mathrm{ls}}$ on "RMT data", i.e., $x_i = \Sigma^{1/2} z_i$, $y_i = x_i^\top \beta + \epsilon_i$, as before
- RHS admits simple asymptotic approximation: $\sigma^2(\frac{n}{n-k} - \frac{n-k}{n})$

We call this "emergent" random-X df as it refers to the complexity that "emerges" from the optimism of $\widehat{f}$ on data at hand. The LHS generally has both bias and variance components, whereas the RHS is pure variance

Emergent random-X df

Using this we can define a quantity we call emergent random-X df: $\mathrm{df}^{\mathrm{e}}(\widehat{f}) = k$, for the number $k \geq 0$ such that

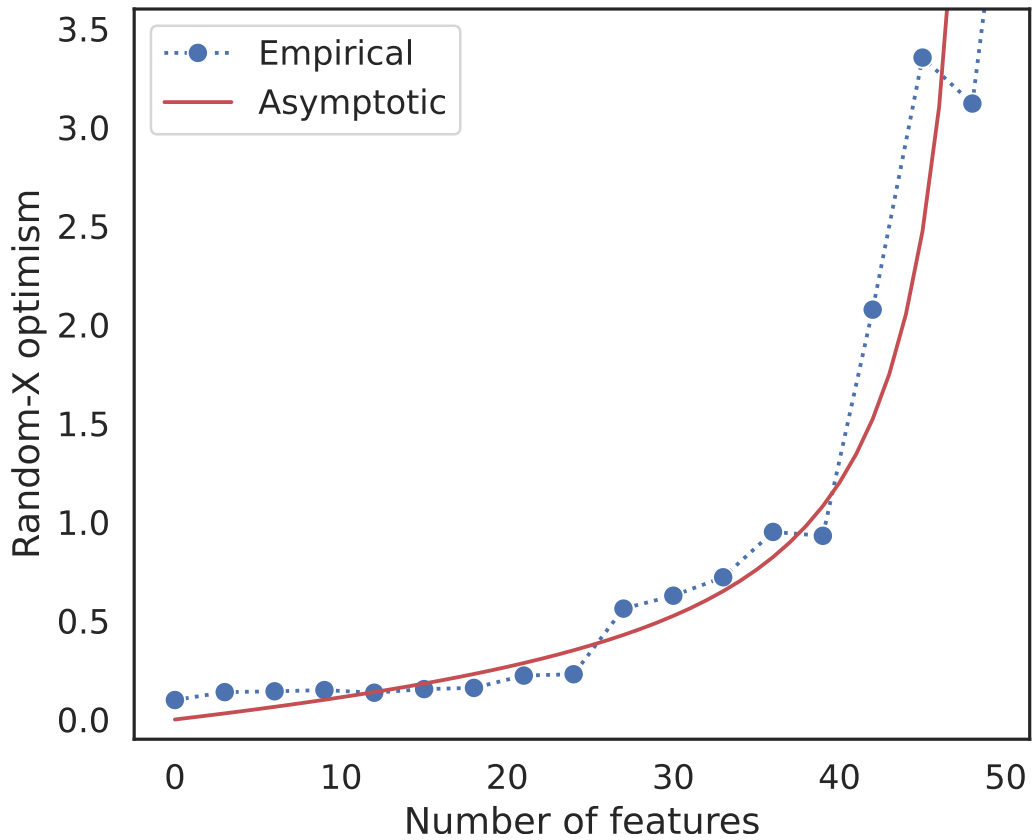
$$\mathrm{OptR}(\widehat{f}; \{\text{original data}\}) = \mathrm{OptR}(\widehat{f}_k^{\mathrm{ls}}; \{\text{RMT data}\})$$

- LHS: random-X opt of the given model $\widehat{f}$ on the original data
- RHS: random-X opt of least squares $\widehat{f}_k^{\mathrm{ls}}$ on "RMT data", i.e., $x_i = \Sigma^{1/2} z_i$, $y_i = x_i^\top \beta + \epsilon_i$, as before
- RHS admits simple asymptotic approximation: $\sigma^2(\frac{n}{n-k} - \frac{n-k}{n})$

We call this "emergent" random-X df as it refers to the complexity that "emerges" from the optimism of $\widehat{f}$ on data at hand. The LHS generally has both bias and variance components, whereas the RHS is pure variance

Illustration: least squares as reference

Random-X optimism for least squares, with $n = 50$:



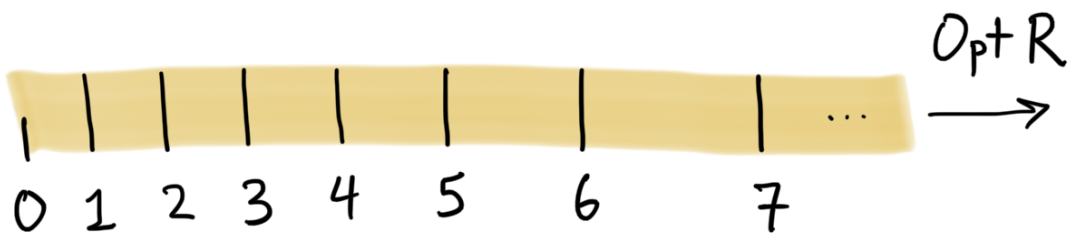
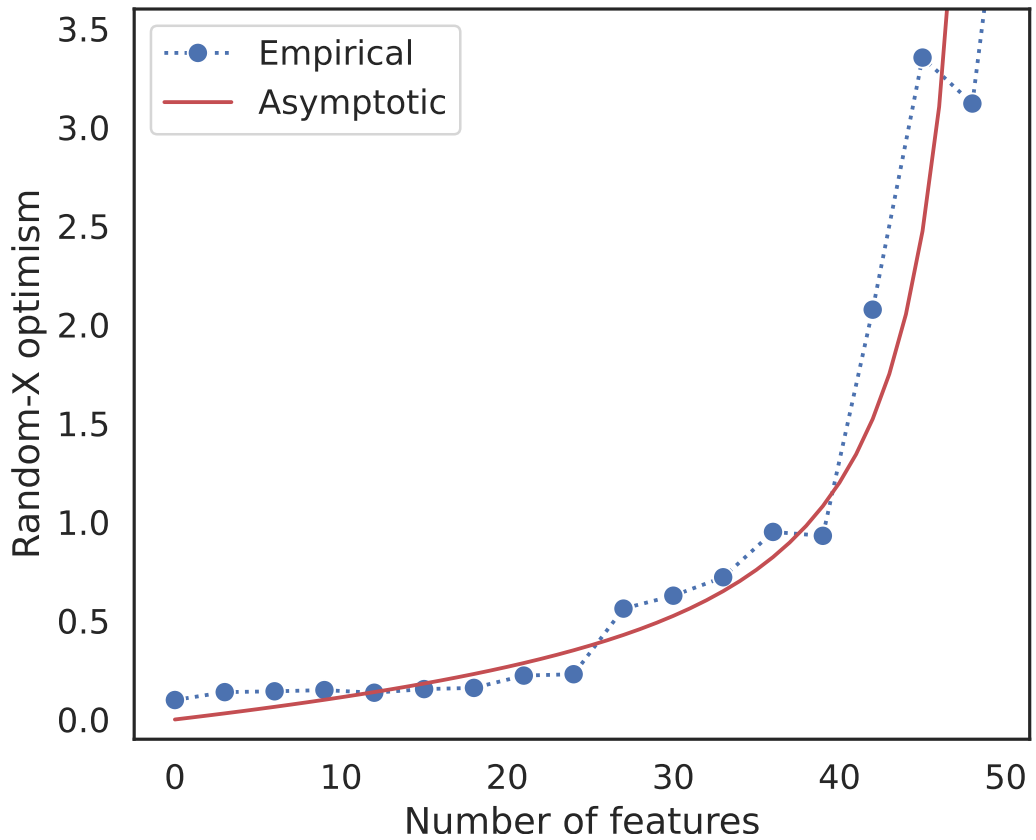
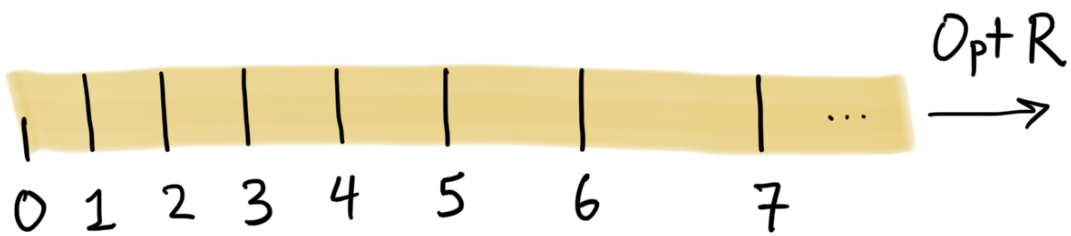


Illustration: least squares as reference

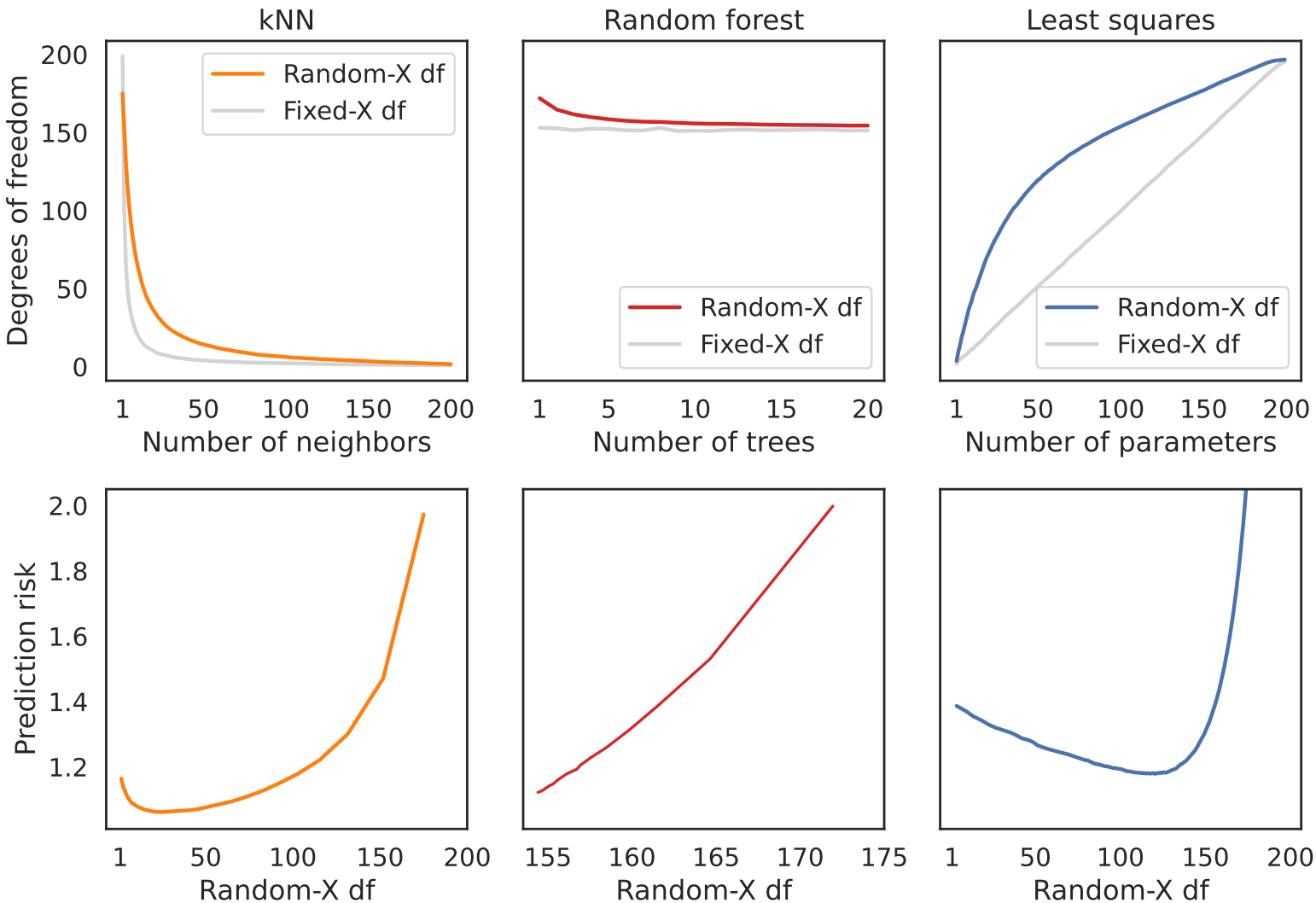
Random-X optimism for least squares, with $n = 50$:





Example: emergent random-X df

Ex with $n = 200$ samples, $d = 300$ features, $\mathbb{E}[y|x]$ nonlinear in x:



Intrinsic random-X df

We can also define a different quantity called intrinsic random-X df: $\mathrm{df}^{\mathrm{i}}(\widehat{f}) = k$, for the number $k \geq 0$ such that

$$\mathrm{OptR}(\widehat{f}; \{\text{original } x, \text{ noisy } y\}) = \mathrm{OptR}(\widehat{f}_k^{\mathrm{ls}}; \{\text{RMT data}\})$$

- LHS: random-X opt of the given model $\widehat{f}$ on original features x_i, but with pure noise for the response $y_i \sim N(0, \sigma^2)$
- RHS: random-X opt of least squares $\widehat{f}_k^{\mathrm{ls}}$ on "RMT data", i.e., $x_i = \Sigma^{1/2} z_i$, $y_i = x_i^\top \beta + \epsilon_i$, as before
- RHS admits simple asymptotic approximation: $\sigma^2(\frac{n}{n-k} - \frac{n-k}{n})$

We call this "intrinsic" random-X df as it refers to the complexity internal to the model $\widehat{f}$, regardless of the data. Note that both the LHS and RHS reflect pure variance

Intrinsic random-X df

We can also define a different quantity called intrinsic random-X df: $\mathrm{df}^{\mathrm{i}}(\widehat{f}) = k$, for the number $k \geq 0$ such that

$$\mathrm{OptR}(\widehat{f}; \{\text{original } x, \text{ noisy } y\}) = \mathrm{OptR}(\widehat{f}_k^{\mathrm{ls}}; \{\text{RMT data}\})$$

- LHS: random-X opt of the given model $\widehat{f}$ on original features x_i, but with pure noise for the response $y_i \sim N(0, \sigma^2)$
- RHS: random-X opt of least squares $\widehat{f}_k^{\mathrm{ls}}$ on "RMT data", i.e., $x_i = \Sigma^{1/2} z_i$, $y_i = x_i^\top \beta + \epsilon_i$, as before
- RHS admits simple asymptotic approximation: $\sigma^2 (\frac{n}{n-k} - \frac{n-k}{n})$

We call this "intrinsic" random-X df as it refers to the complexity internal to the model $\widehat{f}$, regardless of the data. Note that both the LHS and RHS reflect pure variance

Intrinsic random-X df

We can also define a different quantity called intrinsic random-X df: $\mathrm{df}^{\mathrm{i}}(\widehat{f}) = k$, for the number $k \geq 0$ such that

$$\mathrm{OptR}(\widehat{f}; \{\text{original } x, \text{ noisy } y\}) = \mathrm{OptR}(\widehat{f}_k^{\mathrm{ls}}; \{\text{RMT data}\})$$

- LHS: random-X opt of the given model $\widehat{f}$ on original features x_i, but with pure noise for the response $y_i \sim N(0, \sigma^2)$
- RHS: random-X opt of least squares $\widehat{f}_k^{\mathrm{ls}}$ on "RMT data", i.e., $x_i = \Sigma^{1/2} z_i$, $y_i = x_i^\top \beta + \epsilon_i$, as before
- RHS admits simple asymptotic approximation: $\sigma^2 (\frac{n}{n-k} - \frac{n-k}{n})$

We call this "intrinsic" random-X df as it refers to the complexity internal to the model $\widehat{f}$, regardless of the data. Note that both the LHS and RHS reflect pure variance

Intrinsic random-X df

We can also define a different quantity called intrinsic random-X df: $\mathrm{df}^{\mathrm{i}}(\widehat{f}) = k$, for the number $k \geq 0$ such that

$$\mathrm{OptR}(\widehat{f}; \{\text{original } x, \text{ noisy } y\}) = \mathrm{OptR}(\widehat{f}_k^{\mathrm{ls}}; \{\text{RMT data}\})$$

- LHS: random-X opt of the given model $\widehat{f}$ on original features x_i, but with pure noise for the response $y_i \sim N(0, \sigma^2)$
- RHS: random-X opt of least squares $\widehat{f}_k^{\mathrm{ls}}$ on "RMT data", i.e., $x_i = \Sigma^{1/2} z_i$, $y_i = x_i^\top \beta + \epsilon_i$, as before
- RHS admits simple asymptotic approximation: $\sigma^2 \big(\frac{n}{n-k} - \frac{n-k}{n} \big)$

We call this "intrinsic" random-X df as it refers to the complexity internal to the model $\widehat{f}$, regardless of the data. Note that both the LHS and RHS reflect pure variance

Intrinsic random-X df

We can also define a different quantity called intrinsic random-X df: $\mathrm{df}^{\mathrm{i}}(\widehat{f}) = k$, for the number $k \geq 0$ such that

$$\mathrm{OptR}(\widehat{f}; \{\text{original } x, \text{ noisy } y\}) = \mathrm{OptR}(\widehat{f}_k^{\mathrm{ls}}; \{\text{RMT data}\})$$

- LHS: random-X opt of the given model $\widehat{f}$ on original features x_i, but with pure noise for the response $y_i \sim N(0, \sigma^2)$
- RHS: random-X opt of least squares $\widehat{f}_k^{\mathrm{ls}}$ on "RMT data", i.e., $x_i = \Sigma^{1/2} z_i$, $y_i = x_i^\top \beta + \epsilon_i$, as before
- RHS admits simple asymptotic approximation: $\sigma^2 \big(\frac{n}{n-k} - \frac{n-k}{n} \big)$

We call this "intrinsic" random-X df as it refers to the complexity internal to the model $\widehat{f}$, regardless of the data. Note that both the LHS and RHS reflect pure variance

Intrinsic random-X df

We can also define a different quantity called intrinsic random-X df: $\mathrm{df}^{\mathrm{i}}(\widehat{f}) = k$, for the number $k \geq 0$ such that

$$\mathrm{OptR}(\widehat{f}; \{\text{original } x, \text{ noisy } y\}) = \mathrm{OptR}(\widehat{f}_k^{\mathrm{ls}}; \{\text{RMT data}\})$$

- LHS: random-X opt of the given model $\widehat{f}$ on original features x_i, but with pure noise for the response $y_i \sim N(0, \sigma^2)$
- RHS: random-X opt of least squares $\widehat{f}_k^{\mathrm{ls}}$ on "RMT data", i.e., $x_i = \Sigma^{1/2} z_i$, $y_i = x_i^\top \beta + \epsilon_i$, as before
- RHS admits simple asymptotic approximation: $\sigma^2(\frac{n}{n-k} - \frac{n-k}{n})$

We call this "intrinsic" random-X df as it refers to the complexity internal to the model $\widehat{f}$, regardless of the data. Note that both the LHS and RHS reflect pure variance

Intrinsic random-X df

We can also define a different quantity called intrinsic random-X df: $\mathrm{df}^{\mathrm{i}}(\widehat{f}) = k$, for the number $k \geq 0$ such that

$$\mathrm{OptR}(\widehat{f}; \{\text{original } x, \text{ noisy } y\}) = \mathrm{OptR}(\widehat{f}_k^{\mathrm{ls}}; \{\text{RMT data}\})$$

- LHS: random-X opt of the given model $\widehat{f}$ on original features x_i, but with pure noise for the response $y_i \sim N(0, \sigma^2)$
- RHS: random-X opt of least squares $\widehat{f}_k^{\mathrm{ls}}$ on "RMT data", i.e., $x_i = \Sigma^{1/2} z_i$, $y_i = x_i^\top \beta + \epsilon_i$, as before
- RHS admits simple asymptotic approximation: $\sigma^2(\frac{n}{n-k} - \frac{n-k}{n})$

We call this "intrinsic" random-X df as it refers to the complexity internal to the model $\widehat{f}$, regardless of the data. Note that both the LHS and RHS reflect pure variance

Intrinsic random-X df

We can also define a different quantity called intrinsic random-X df: $\mathrm{df}^{\mathrm{i}}(\widehat{f}) = k$, for the number $k \geq 0$ such that

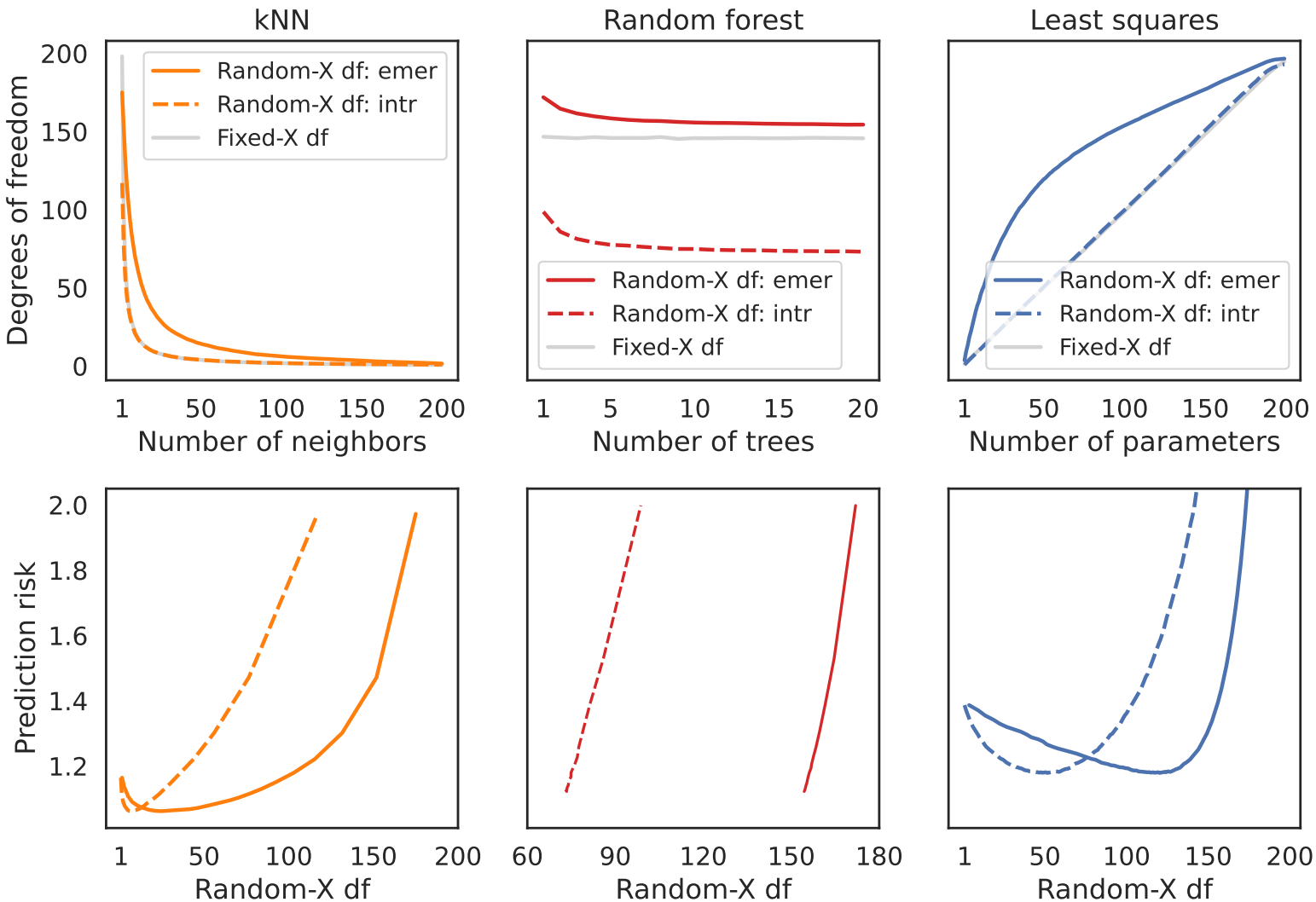
$$\mathrm{OptR}(\widehat{f}; \{\text{original } x, \text{ noisy } y\}) = \mathrm{OptR}(\widehat{f}_k^{\mathrm{ls}}; \{\text{RMT data}\})$$

- LHS: random-X opt of the given model $\widehat{f}$ on original features x_i, but with pure noise for the response $y_i \sim N(0, \sigma^2)$
- RHS: random-X opt of least squares $\widehat{f}_k^{\mathrm{ls}}$ on "RMT data", i.e., $x_i = \Sigma^{1/2} z_i$, $y_i = x_i^\top \beta + \epsilon_i$, as before
- RHS admits simple asymptotic approximation: $\sigma^2(\frac{n}{n-k} - \frac{n-k}{n})$

We call this "intrinsic" random-X df as it refers to the complexity internal to the model $\widehat{f}$, regardless of the data. Note that both the LHS and RHS reflect pure variance

Example: intrinsic random-X df

Ex with $n = 200$ samples, $d = 300$ features, $\mathbb{E}[y|x]$ nonlinear in x:

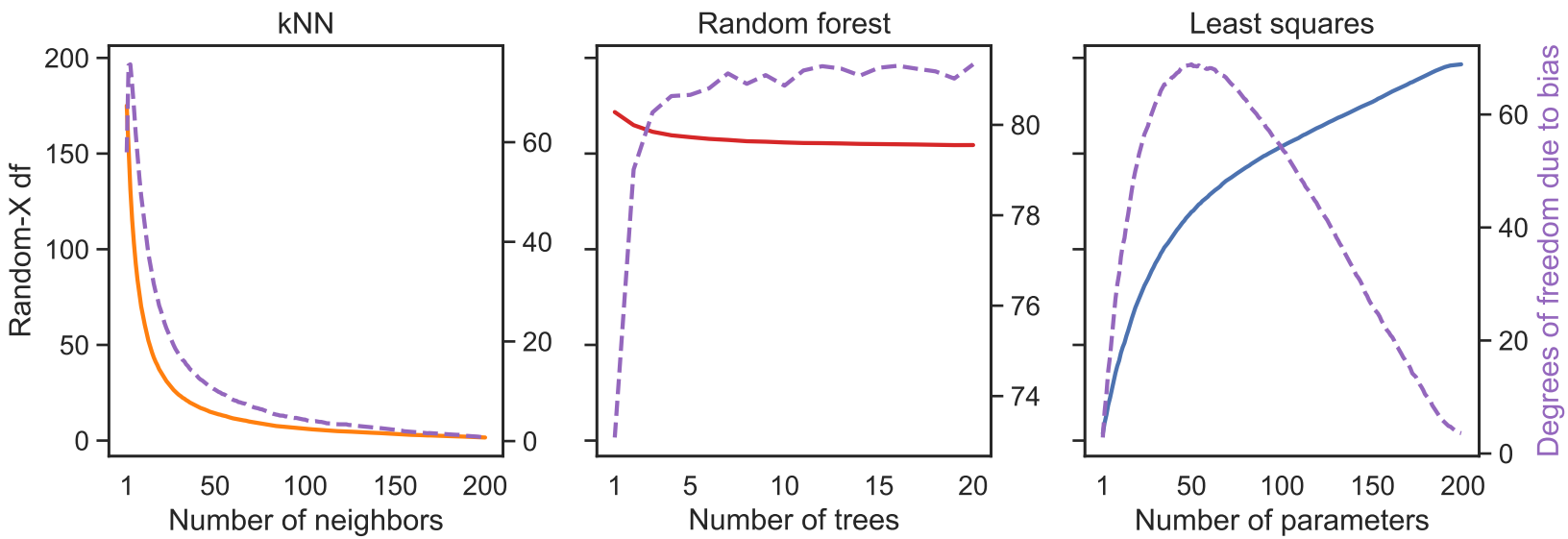


Bias attribution

By subtracting emergent and intrinsic degrees of freedom, we are left with the df due to bias:

$$\mathrm{df}^{\mathsf{bias}}(\widehat{f}) = \mathrm{df}^{\mathsf{e}}(\widehat{f}) - \mathrm{df}^{\mathsf{i}}(\widehat{f})$$

Back to our example:

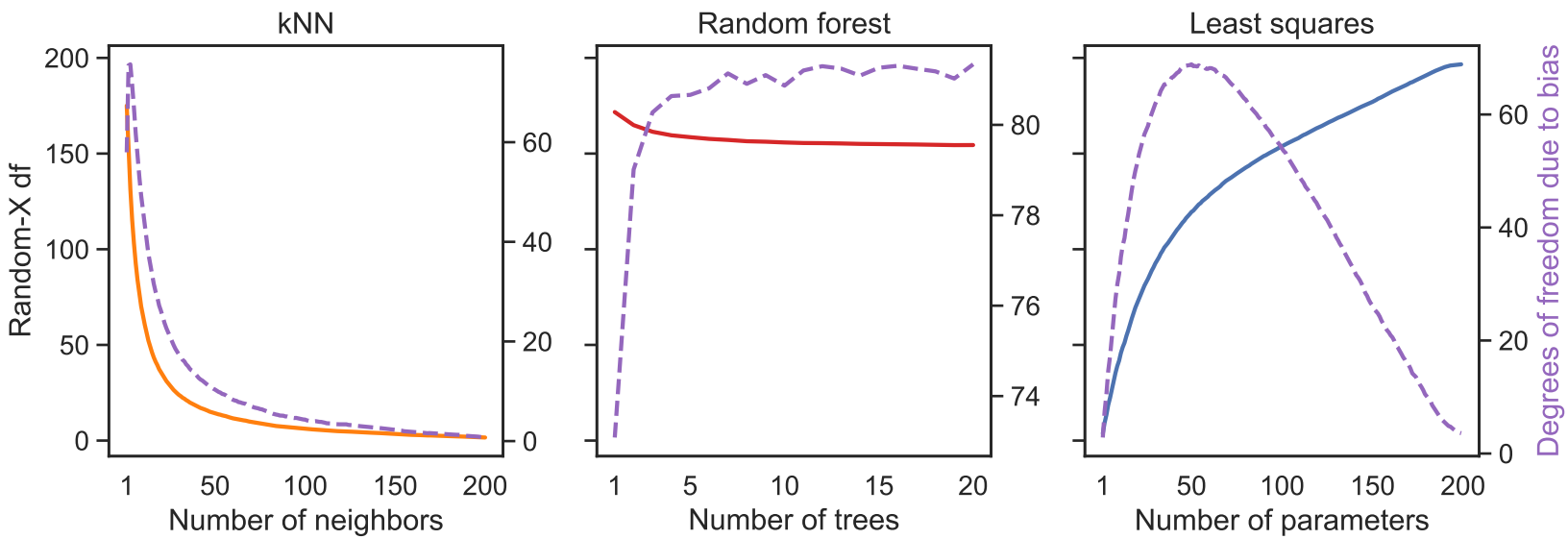


Bias attribution

By subtracting emergent and intrinsic degrees of freedom, we are left with the df due to bias:

$$\mathrm{df}^{\mathsf{bias}}(\widehat{f}) = \mathrm{df}^{\mathsf{e}}(\widehat{f}) - \mathrm{df}^{\mathsf{i}}(\widehat{f})$$

Back to our example:



General decomposition

Generic decompositions are possible. Let $e_1, \ldots, e_m$ be any list of user-chosen error components. E.g., $e_1 =$ bias, $e_2 =$ covariate shift, and so on

For any subset $S \subseteq \{e_1, \ldots, e_m\}$, let $\mathrm{df}^S(\widehat{f}) = k$, for the number $k \geq 0$ such that

$$\mathrm{OptR}(\widehat{f}; \{\text{data subject to } S\}) = \mathrm{OptR}(\widehat{f}_k^{\mathrm{ls}}; \{\text{RMT data}\})$$

We can then attribute d_i df to each error component e_i, as follows:

$$d_i = \sum_{S \subseteq \{e_1, \ldots, e_m\} \setminus \{e_i\}} \frac{|S|!(m - |S| - 1)!}{m!} (\mathrm{df}^{S \cup \{e_i\}}(\widehat{f}) - \mathrm{df}^S(\widehat{f}))$$

This is an instance of a Shapley value. Therefore it obeys all of the Shapley axioms; in particular, efficiency: $\sum_{i=1}^m d_i = \mathrm{df}^{\mathrm{e}}(\widehat{f})$

General decomposition

Generic decompositions are possible. Let $e_1, \ldots, e_m$ be any list of user-chosen error components. E.g., $e_1 =$ bias, $e_2 =$ covariate shift, and so on

For any subset $S \subseteq \{e_1, \ldots, e_m\}$, let $\mathrm{df}^S(\widehat{f}) = k$, for the number $k \geq 0$ such that

$$\mathrm{OptR}(\widehat{f}; \{\text{data subject to } S\}) = \mathrm{OptR}(\widehat{f}_k^{\mathrm{ls}}; \{\text{RMT data}\})$$

We can then attribute d_i df to each error component e_i, as follows:

$$d_i = \sum_{S \subseteq \{e_1, \ldots, e_m\} \setminus \{e_i\}} \frac{|S|!(m - |S| - 1)!}{m!} (\mathrm{df}^{S \cup \{e_i\}}(\widehat{f}) - \mathrm{df}^S(\widehat{f}))$$

This is an instance of a Shapley value. Therefore it obeys all of the Shapley axioms; in particular, efficiency: $\sum_{i=1}^m d_i = \mathrm{df}^{\mathrm{e}}(\widehat{f})$

General decomposition

Generic decompositions are possible. Let $e_1, \ldots, e_m$ be any list of user-chosen error components. E.g., $e_1 =$ bias, $e_2 =$ covariate shift, and so on

For any subset $S \subseteq \{e_1, \ldots, e_m\}$, let $\mathrm{df}^S(\widehat{f}) = k$, for the number $k \geq 0$ such that

$$\mathrm{OptR}(\widehat{f}; \{\text{data subject to } S\}) = \mathrm{OptR}(\widehat{f}_k^{\mathrm{ls}}; \{\text{RMT data}\})$$

We can then attribute d_i df to each error component e_i, as follows:

$$d_i = \sum_{S \subseteq \{e_1, \ldots, e_m\} \setminus \{e_i\}} \frac{|S|!(m - |S| - 1)!}{m!} (\mathrm{df}^{S \cup \{e_i\}}(\widehat{f}) - \mathrm{df}^S(\widehat{f}))$$

This is an instance of a Shapley value. Therefore it obeys all of the Shapley axioms; in particular, efficiency: $\sum_{i=1}^m d_i = \mathrm{df}^{\mathrm{e}}(\widehat{f})$

General decomposition

Generic decompositions are possible. Let $e_1, \ldots, e_m$ be any list of user-chosen error components. E.g., $e_1 =$ bias, $e_2 =$ covariate shift, and so on

For any subset $S \subseteq \{e_1, \ldots, e_m\}$, let $\mathrm{df}^S(\widehat{f}) = k$, for the number $k \geq 0$ such that

$$\mathrm{OptR}(\widehat{f}; \{\text{data subject to } S\}) = \mathrm{OptR}(\widehat{f}_k^{\mathrm{ls}}; \{\text{RMT data}\})$$

We can then attribute d_i df to each error component e_i, as follows:

$$d_i = \sum_{S \subseteq \{e_1, \ldots, e_m\} \setminus \{e_i\}} \frac{|S|!(m - |S| - 1)!}{m!} (\mathrm{df}^{S \cup \{e_i\}}(\widehat{f}) - \mathrm{df}^S(\widehat{f}))$$

This is an instance of a Shapley value. Therefore it obeys all of the Shapley axioms; in particular, efficiency: $\sum_{i=1}^m d_i = \mathrm{df}^{\mathrm{e}}(\widehat{f})$

General decomposition

Generic decompositions are possible. Let $e_1, \ldots, e_m$ be any list of user-chosen error components. E.g., $e_1 =$ bias, $e_2 =$ covariate shift, and so on

For any subset $S \subseteq \{e_1, \ldots, e_m\}$, let $\mathrm{df}^S(\widehat{f}) = k$, for the number $k \geq 0$ such that

$$\mathrm{OptR}(\widehat{f}; \{\text{data subject to } S\}) = \mathrm{OptR}(\widehat{f}_k^{\mathrm{ls}}; \{\text{RMT data}\})$$

We can then attribute d_i df to each error component e_i, as follows:

$$d_i = \sum_{S \subseteq \{e_1, \ldots, e_m\} \setminus \{e_i\}} \frac{|S|!(m - |S| - 1)!}{m!} (\mathrm{df}^{S \cup \{e_i\}}(\widehat{f}) - \mathrm{df}^S(\widehat{f}))$$

This is an instance of a Shapley value. Therefore it obeys all of the Shapley axioms; in particular, efficiency: $\sum_{i=1}^m d_i = \mathrm{df}^{\mathrm{e}}(\widehat{f})$

General decomposition

Generic decompositions are possible. Let $e_1, \ldots, e_m$ be any list of user-chosen error components. E.g., $e_1 =$ bias, $e_2 =$ covariate shift, and so on

For any subset $S \subseteq \{e_1, \ldots, e_m\}$, let $\mathrm{df}^S(\widehat{f}) = k$, for the number $k \geq 0$ such that

$$\mathrm{OptR}(\widehat{f}; \{\text{data subject to } S\}) = \mathrm{OptR}(\widehat{f}_k^{\mathrm{ls}}; \{\text{RMT data}\})$$

We can then attribute d_i df to each error component e_i, as follows:

$$d_i = \sum_{S \subseteq \{e_1, \ldots, e_m\} \setminus \{e_i\}} \frac{|S|!(m - |S| - 1)!}{m!} (\mathrm{df}^{S \cup \{e_i\}}(\widehat{f}) - \mathrm{df}^S(\widehat{f}))$$

This is an instance of a Shapley value. Therefore it obeys all of the Shapley axioms; in particular, efficiency: $\sum_{i=1}^m d_i = \mathrm{df}^{\mathrm{e}}(\widehat{f})$

General decomposition

Generic decompositions are possible. Let $e_1, \ldots, e_m$ be any list of user-chosen error components. E.g., $e_1 =$ bias, $e_2 =$ covariate shift, and so on

For any subset $S \subseteq \{e_1, \ldots, e_m\}$, let $\mathrm{df}^S(\widehat{f}) = k$, for the number $k \geq 0$ such that

$$\mathrm{OptR}(\widehat{f}; \{\text{data subject to } S\}) = \mathrm{OptR}(\widehat{f}_k^{\mathrm{ls}}; \{\text{RMT data}\})$$

We can then attribute d_i df to each error component e_i, as follows:

$$d_i = \sum_{S \subseteq \{e_1, \ldots, e_m\} \setminus \{e_i\}} \frac{|S|!(m - |S| - 1)!}{m!} (\mathrm{df}^{S \cup \{e_i\}}(\widehat{f}) - \mathrm{df}^S(\widehat{f}))$$

This is an instance of a Shapley value. Therefore it obeys all of the Shapley axioms; in particular, efficiency: $\sum_{i=1}^m d_i = \mathrm{df}^{\mathrm{e}}(\widehat{f})$

General decomposition

Generic decompositions are possible. Let $e_1, \ldots, e_m$ be any list of user-chosen error components. E.g., $e_1 =$ bias, $e_2 =$ covariate shift, and so on

For any subset $S \subseteq \{e_1, \ldots, e_m\}$, let $\mathrm{df}^S(\widehat{f}) = k$, for the number $k \geq 0$ such that

$$\mathrm{OptR}(\widehat{f}; \{\text{data subject to } S\}) = \mathrm{OptR}(\widehat{f}_k^{\mathrm{ls}}; \{\text{RMT data}\})$$

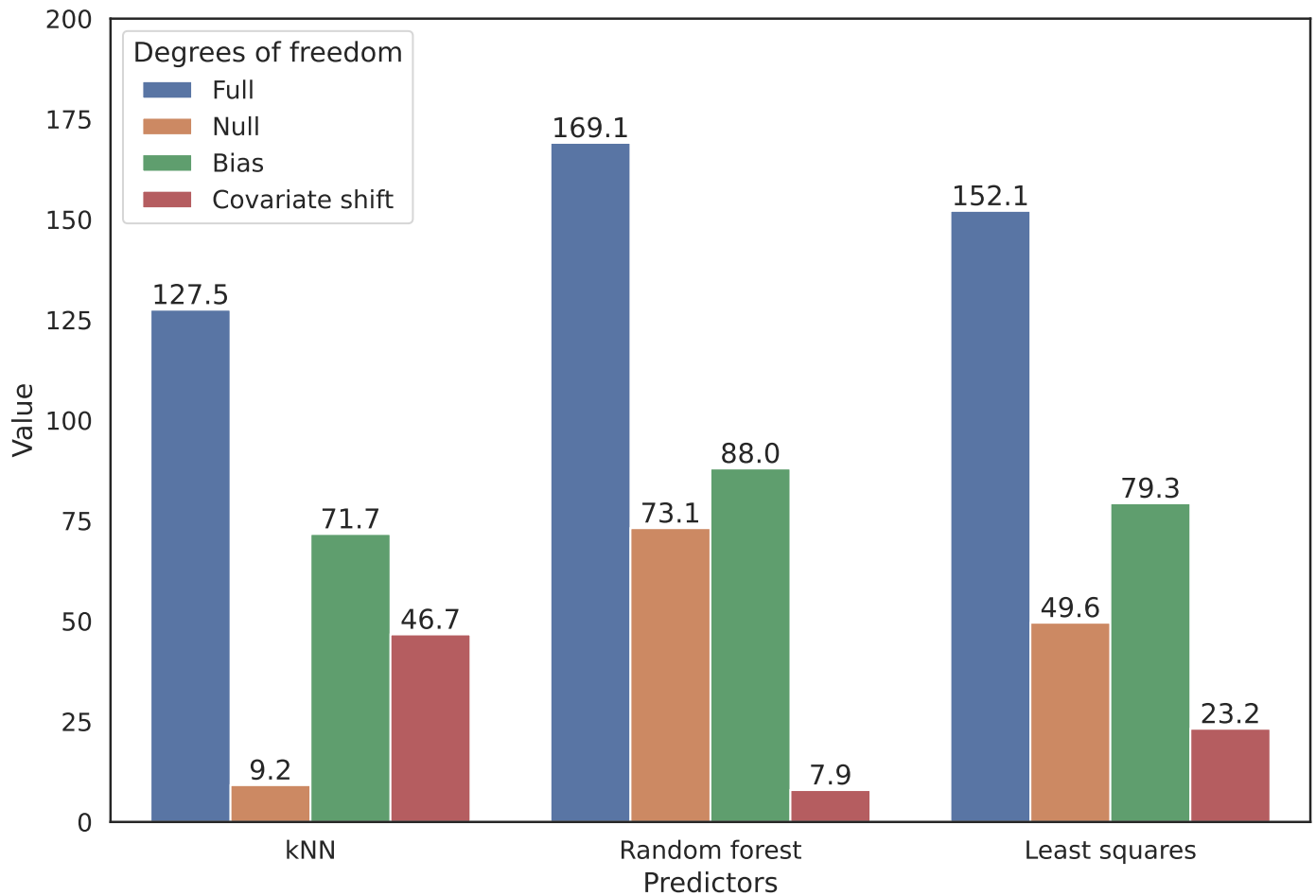
We can then attribute d_i df to each error component e_i, as follows:

$$d_i = \sum_{S \subseteq \{e_1, \ldots, e_m\} \setminus \{e_i\}} \frac{|S|!(m - |S| - 1)!}{m!} (\mathrm{df}^{S \cup \{e_i\}}(\widehat{f}) - \mathrm{df}^S(\widehat{f}))$$

This is an instance of a Shapley value. Therefore it obeys all of the Shapley axioms; in particular, efficiency: $\sum_{i=1}^m d_i = \mathrm{df}^{\mathrm{e}}(\widehat{f})$

Example: bias and covariate shift

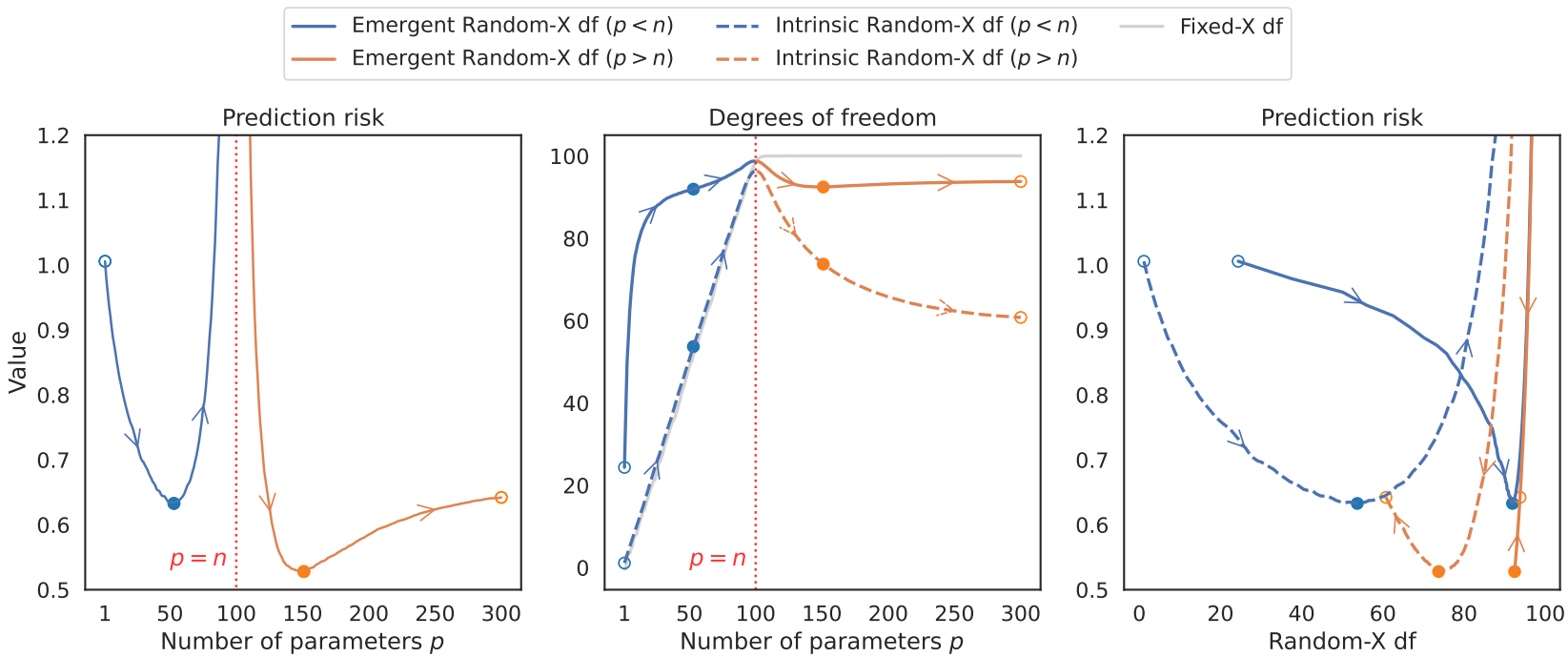
Ex with $n = 200$ samples, $d = 300$ features, $\mathbb{E}[y|x]$ nonlinear in x, and covariate shift ($\Sigma \to \tilde{\Sigma}$):



Double descent, revisited

We can use random-X df (any flavor) to reparametrize error curve for models with double descent. The df map is not monotone, but it shows that in the overparametrized regime, the effective number of parameters can actually be small

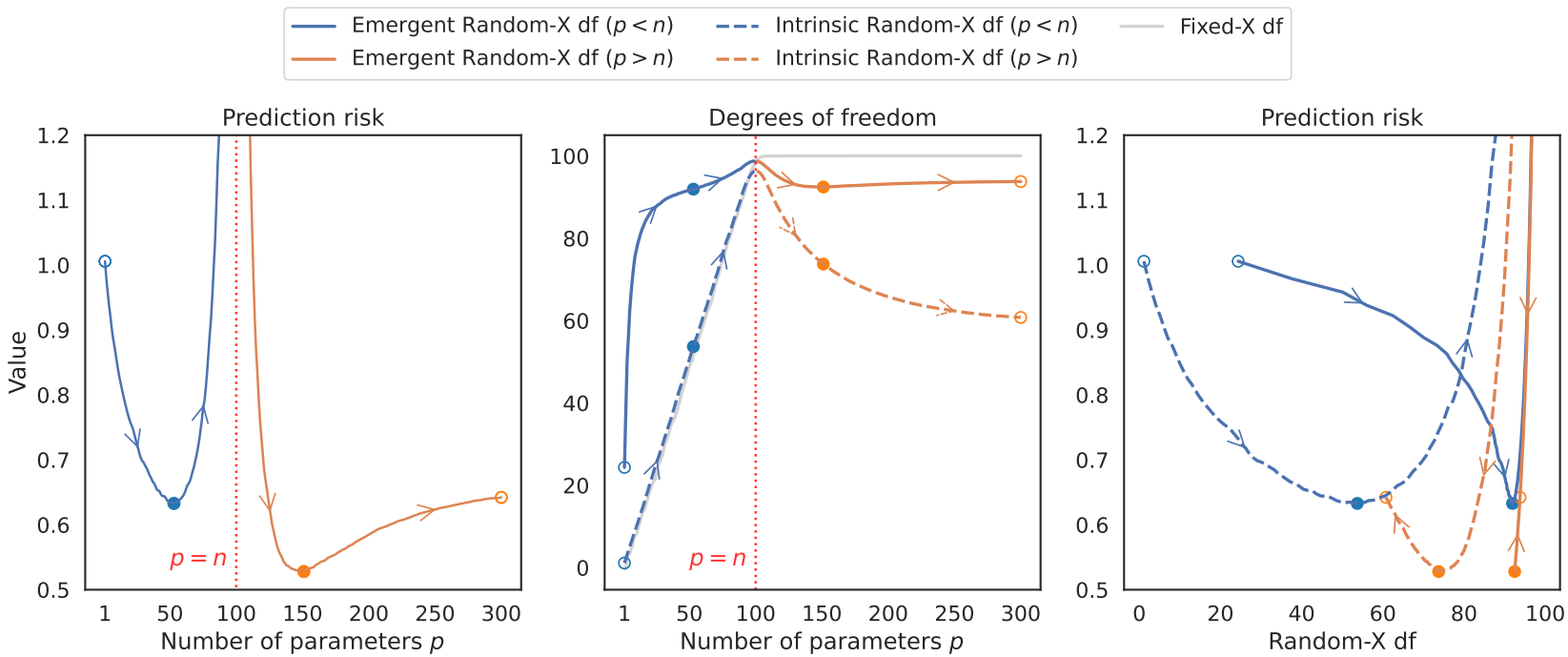
Back to our running example:



Double descent, revisited

We can use random-X df (any flavor) to reparametrize error curve for models with double descent. The df map is not monotone, but it shows that in the overparametrized regime, the effective number of parameters can actually be small

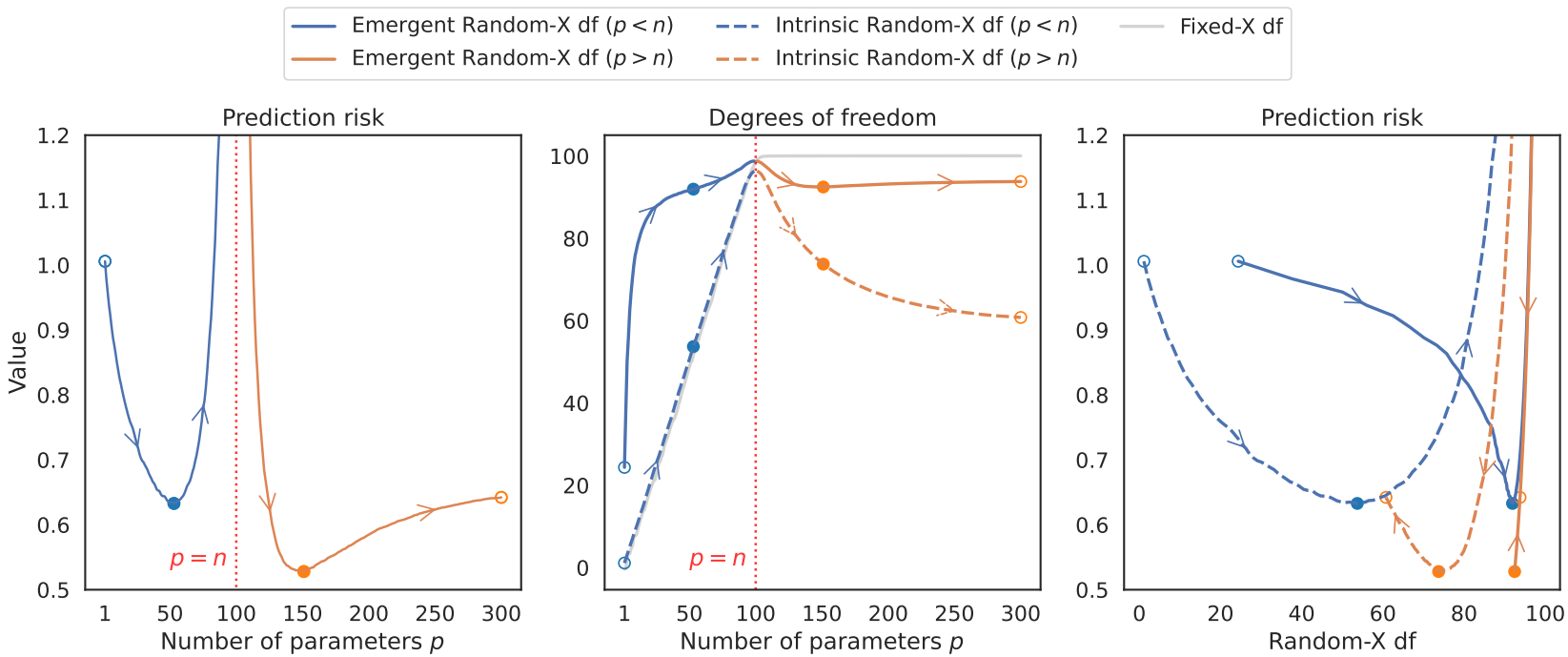
Back to our running example:



Double descent, revisited

We can use random-X df (any flavor) to reparametrize error curve for models with double descent. The df map is not monotone, but it shows that in the overparametrized regime, the effective number of parameters can actually be small

Back to our running example:



Outline

Discussion and future directions

A high-level view of the work:

- Suppose we are given a family of models for which we want a complexity measure under a specific error metric.
- Construct a family of "reference" models spanning same optimisms.
- Find the model in the reference family that is closest to the observed optimism. Declare complexity as complexity of that reference model.

Key relation:

$$\mathrm{OptR}(\widehat{f}) = \mathrm{OptR}(\widehat{f}^{\mathrm{ref}})$$

Going beyond . . .

- Other error metrics beyond squared error
- Unsupervised setting?

Discussion and future directions

A high-level view of the work:

- Suppose we are given a family of models for which we want a complexity measure under a specific error metric.
- Construct a family of "reference" models spanning same optimisms.
- Find the model in the reference family that is closest to the observed optimism. Declare complexity as complexity of that reference model.

Key relation:

$$\mathrm{OptR}(\widehat{f}) = \mathrm{OptR}(\widehat{f}^{\mathrm{ref}})$$

Going beyond . . .

- Other error metrics beyond squared error
- Unsupervised setting?

Discussion and future directions

A high-level view of the work:

- Suppose we are given a family of models for which we want a complexity measure under a specific error metric.
- Construct a family of "reference" models spanning same optimisms.
- Find the model in the reference family that is closest to the observed optimism. Declare complexity as complexity of that reference model.

Key relation:

$$\mathrm{OptR}(\widehat{f}) = \mathrm{OptR}(\widehat{f}^{\mathrm{ref}})$$

Going beyond . . .

- Other error metrics beyond squared error
- Unsupervised setting?

Discussion and future directions

A high-level view of the work:

- Suppose we are given a family of models for which we want a complexity measure under a specific error metric.
- Construct a family of "reference" models spanning same optimisms.
- Find the model in the reference family that is closest to the observed optimism. Declare complexity as complexity of that reference model.

Key relation:

$$\mathrm{OptR}(\widehat{f}) = \mathrm{OptR}(\widehat{f}^{\mathrm{ref}})$$

Going beyond . . .

- Other error metrics beyond squared error
- Unsupervised setting?

Discussion and future directions

A high-level view of the work:

- Suppose we are given a family of models for which we want a complexity measure under a specific error metric.
- Construct a family of "reference" models spanning same optimisms.
- Find the model in the reference family that is closest to the observed optimism. Declare complexity as complexity of that reference model.

Key relation:

$$\mathrm{OptR}(\widehat{f}) = \mathrm{OptR}(\widehat{f}^{\mathrm{ref}})$$

Going beyond . . .

- Other error metrics beyond squared error
- Unsupervised setting?

Discussion and future directions

A high-level view of the work:

- Suppose we are given a family of models for which we want a complexity measure under a specific error metric.
- Construct a family of "reference" models spanning same optimisms.
- Find the model in the reference family that is closest to the observed optimism. Declare complexity as complexity of that reference model.

Key relation:

$$\mathrm{OptR}(\widehat{f}) = \mathrm{OptR}(\widehat{f}^{\mathrm{ref}})$$

Going beyond ...

- Other error metrics beyond squared error
- Unsupervised setting?

Discussion and future directions

A high-level view of the work:

- Suppose we are given a family of models for which we want a complexity measure under a specific error metric.
- Construct a family of "reference" models spanning same optimisms.
- Find the model in the reference family that is closest to the observed optimism. Declare complexity as complexity of that reference model.

Key relation:

$$\mathrm{OptR}(\widehat{f}) = \mathrm{OptR}(\widehat{f}^{\mathrm{ref}})$$

Going beyond ...

- Other error metrics beyond squared error
- Unsupervised setting?

Discussion and future directions

A high-level view of the work:

- Suppose we are given a family of models for which we want a complexity measure under a specific error metric.
- Construct a family of "reference" models spanning same optimisms.
- Find the model in the reference family that is closest to the observed optimism. Declare complexity as complexity of that reference model.

Key relation:

$$\mathrm{OptR}(\widehat{f}) = \mathrm{OptR}(\widehat{f}^{\mathrm{ref}})$$

Going beyond ...

- Other error metrics beyond squared error
- Unsupervised setting?

Thanks for listening!

Questions/comments/thoughts?