Overparametrization in linear models, and uniform consistency of cross-validation for ridge regression

Pratik Patil, Yuting Wei, Alessandro Rinaldo, Ryan J. Tibshirani

Carnegie Mellon University

JSM 2021

Overparametrization in machine learning

Modern deep learning models typically fit a huge number of parameters. Such overparametrization 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 on training data
- Generalization: despite overfitting, models generalize well in practice

Overparametrization in machine learning

Modern deep learning models typically fit a huge number of parameters. Such overparametrization 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 on training data
- Generalization: despite overfitting, models generalize well in practice

Overparametrization in machine learning

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

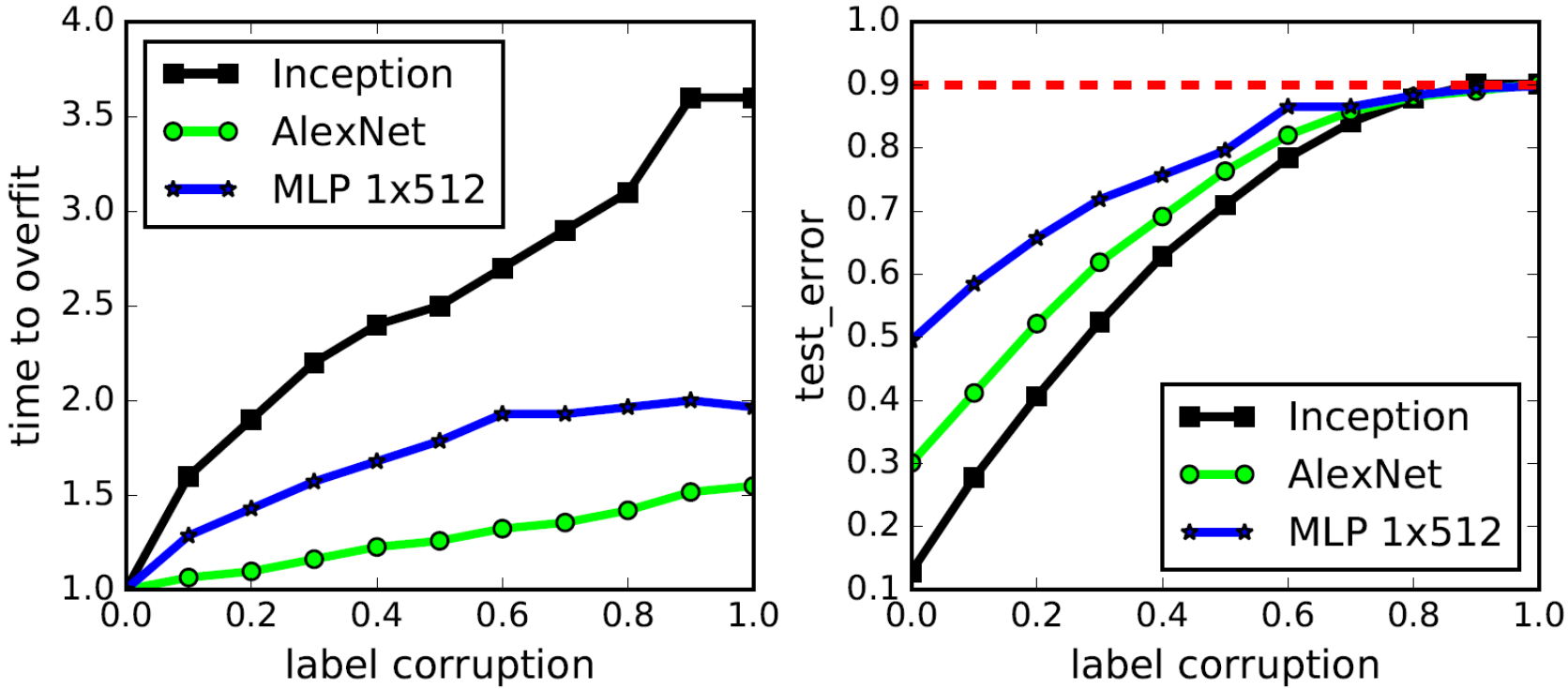
- Representation: allows rich, expressive models for diverse real data
- Optimization: simple, local optimization methods often find near-optimal solutions to empirical risk minimization on training data
- Generalization: despite overfitting, models generalize well in practice

Overparametrization in machine learning

Modern deep learning models typically fit a huge number of parameters. Such overparametrization 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 on training data
- Generalization: despite overfitting, models generalize well in practice

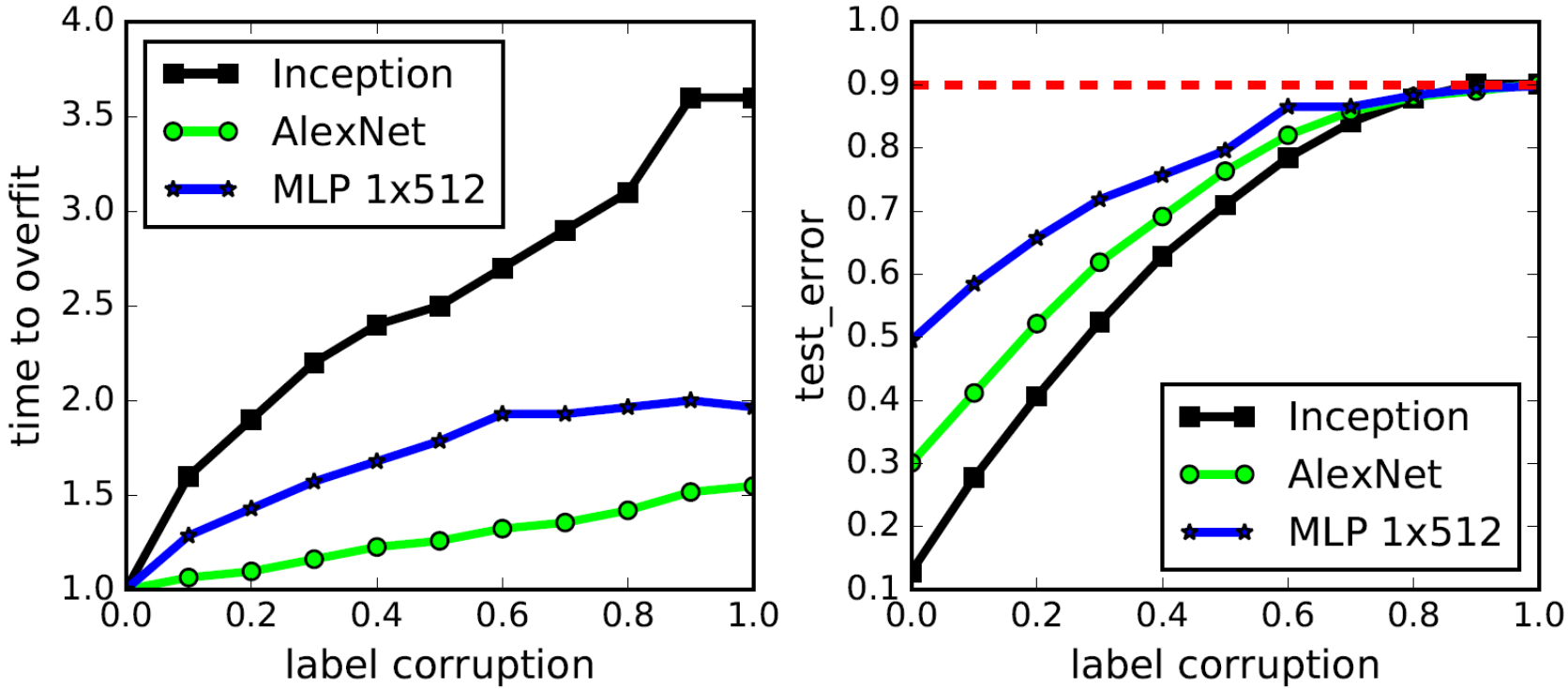
An influential experiment



"Understanding deep learning requires rethinking generalization"
Zhang, Bengio, Hardt, Recht, Vinyals, 2017

- CIFAR10 data (60,000 images [32 × 32]) with artificial label noise
- Three neural network architectures (with number of parameters): Inception (1,649,402), AlexNet (1,387,786), MLP 1x512 (1,209,866)

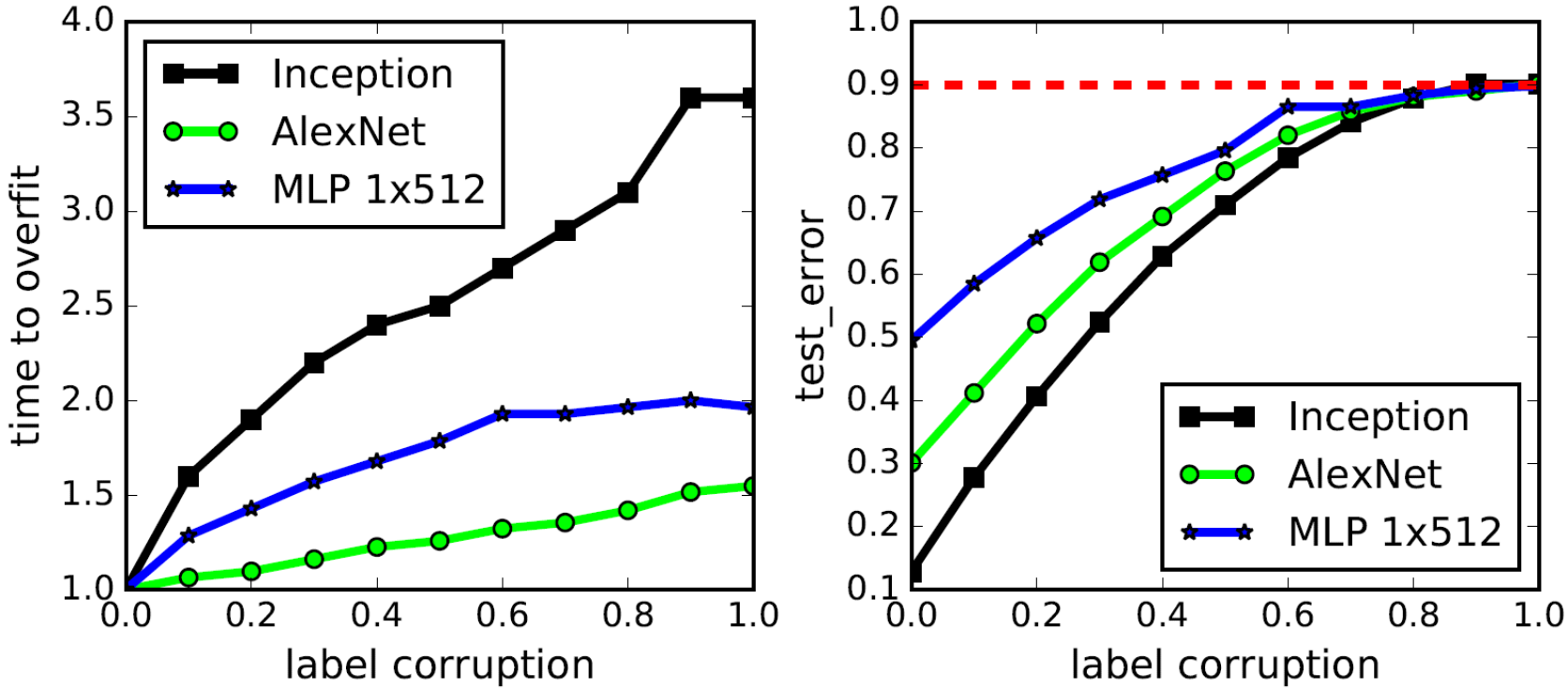
An influential experiment



"Understanding deep learning requires rethinking generalization"
Zhang, Bengio, Hardt, Recht, Vinyals, 2017

- CIFAR10 data (60,000 images [32 $\times$ 32]) with artificial label noise
- Three neural network architectures (with number of parameters): Inception (1,649,402), AlexNet (1,387,786), MLP 1x512 (1,209,866)

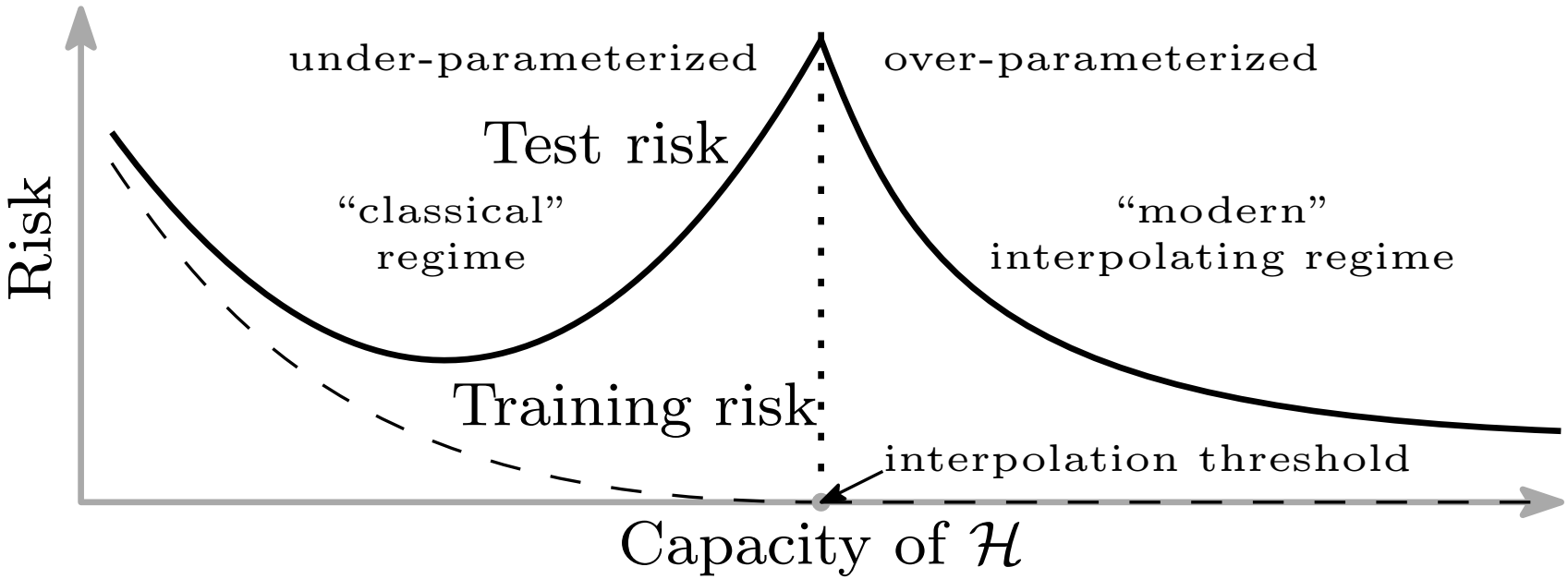
An influential experiment



"Understanding deep learning requires rethinking generalization"
Zhang, Bengio, Hardt, Recht, Vinyals, 2017

- CIFAR10 data (60,000 images [32 $\times$ 32]) with artificial label noise
- Three neural network architectures (with number of parameters): Inception (1,649,402), AlexNet (1,387,786), MLP 1x512 (1,209,866)

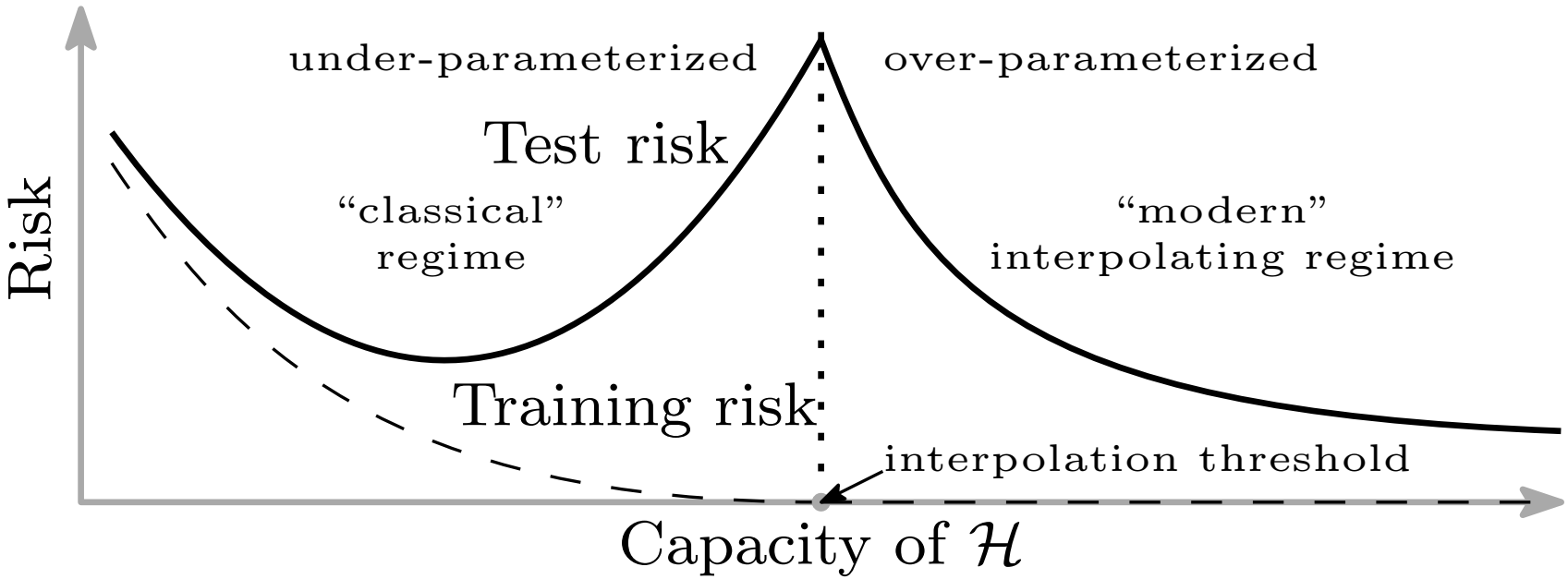
An influential follow-up



"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 not only with neural networks, but more generally for kernel machines, random forest, boosting, etc.

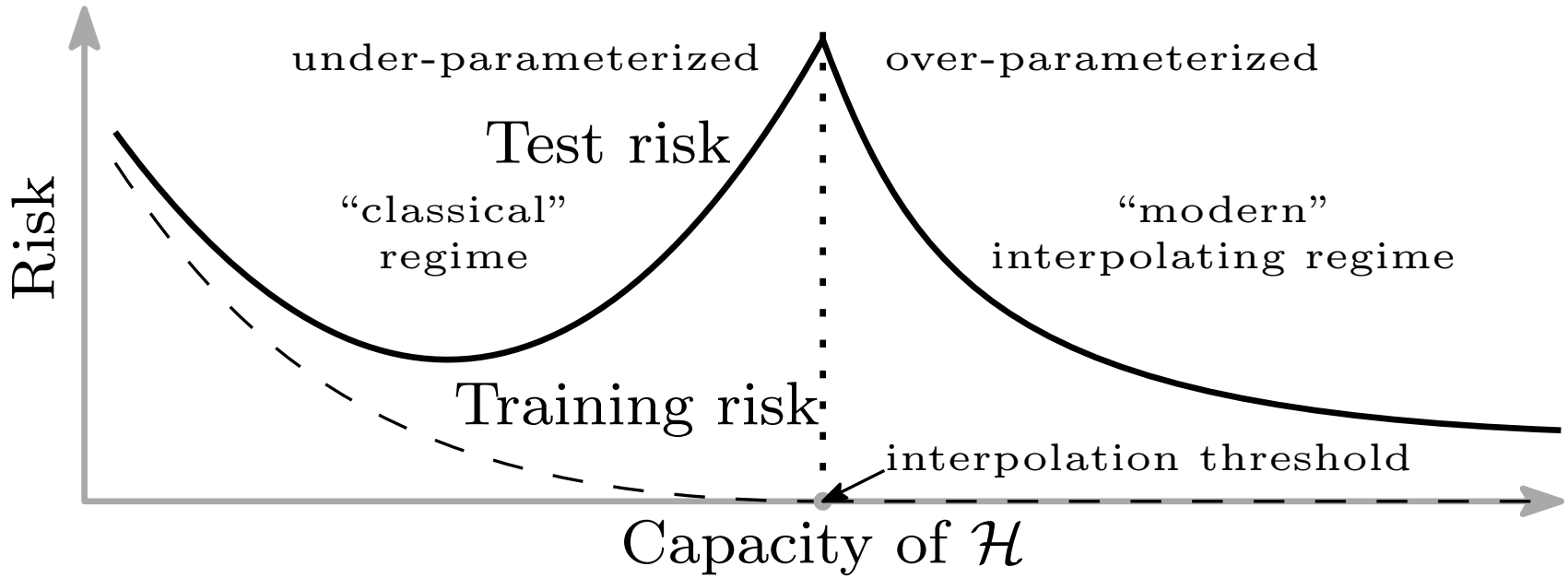
An influential follow-up



"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 not only with neural networks, but more generally for kernel machines, random forest, boosting, etc.

An influential follow-up



"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 not only with neural networks, but more generally for kernel machines, random forest, boosting, etc.

Conceptual and theoretical challenges

In summary, current deep learning practice suggests:

- we should design neural networks to be highly overparametrized
- train until nearly zero training error (i.e., interpolate training data)
- the trained models still can generalize well and have good test error

Main statistical challenges for us:

- benign overfitting conceptually breaks classical bias-variance tradeoff
- deep models fall outside realm of uniform convergence
- core question is how to correctly measure model complexity

Conceptual and theoretical challenges

In summary, current deep learning practice suggests:

- we should design neural networks to be highly overparametrized
- train until nearly zero training error (i.e., interpolate training data)
- the trained models still can generalize well and have good test error

Main statistical challenges for us:

- benign overfitting conceptually breaks classical bias-variance tradeoff
- deep models fall outside realm of uniform convergence
- core question is how to correctly measure model complexity

Conceptual and theoretical challenges

In summary, current deep learning practice suggests:

- we should design neural networks to be highly overparametrized
- train until nearly zero training error (i.e., interpolate training data)
- the trained models still can generalize well and have good test error

Main statistical challenges for us:

- benign overfitting conceptually breaks classical bias-variance tradeoff
- deep models fall outside realm of uniform convergence
- core question is how to correctly measure model complexity

Conceptual and theoretical challenges

In summary, current deep learning practice suggests:

- we should design neural networks to be highly overparametrized
- train until nearly zero training error (i.e., interpolate training data)
- the trained models still can generalize well and have good test error

Main statistical challenges for us:

- benign overfitting conceptually breaks classical bias-variance tradeoff
- deep models fall outside realm of uniform convergence
- core question is how to correctly measure model complexity

Conceptual and theoretical challenges

In summary, current deep learning practice suggests:

- we should design neural networks to be highly overparametrized
- train until nearly zero training error (i.e., interpolate training data)
- the trained models still can generalize well and have good test error

Main statistical challenges for us:

- benign overfitting conceptually breaks classical bias-variance tradeoff
- deep models fall outside realm of uniform convergence
- core question is how to correctly measure model complexity

Conceptual and theoretical challenges

In summary, current deep learning practice suggests:

- we should design neural networks to be highly overparametrized
- train until nearly zero training error (i.e., interpolate training data)
- the trained models still can generalize well and have good test error

Main statistical challenges for us:

- benign overfitting conceptually breaks classical bias-variance tradeoff
- deep models fall outside realm of uniform convergence
- core question is how to correctly measure model complexity

Conceptual and theoretical challenges

In summary, current deep learning practice suggests:

- we should design neural networks to be highly overparametrized
- train until nearly zero training error (i.e., interpolate training data)
- the trained models still can generalize well and have good test error

Main statistical challenges for us:

- benign overfitting conceptually breaks classical bias-variance tradeoff
- deep models fall outside realm of uniform convergence
- core question is how to correctly measure model complexity

Conceptual and theoretical challenges

In summary, current deep learning practice suggests:

- we should design neural networks to be highly overparametrized
- train until nearly zero training error (i.e., interpolate training data)
- the trained models still can generalize well and have good test error

Main statistical challenges for us:

- benign overfitting conceptually breaks classical bias-variance tradeoff
- deep models fall outside realm of uniform convergence
- core question is how to correctly measure model complexity

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 ...

This work: cross-validation for ridge regression in overparametrized regime that includes the min-norm least squares interpolator (zero regularization)

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 ...

This work: cross-validation for ridge regression in overparametrized regime that includes the min-norm least squares interpolator (zero regularization)

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 ...

This work: cross-validation for ridge regression in overparametrized regime that includes the min-norm least squares interpolator (zero regularization)

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 ...

This work: cross-validation for ridge regression in overparametrized regime that includes the min-norm least squares interpolator (zero regularization)

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 ...

This work: cross-validation for ridge regression in overparametrized regime that includes the min-norm least squares interpolator (zero regularization)

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 ...

This work: cross-validation for ridge regression in overparametrized regime that includes the min-norm least squares interpolator (zero regularization)

Linearized overparametrized setting

Standard regression setting with n i.i.d. data $(x_i, y_i) \in \mathbb{R}^p \times \mathbb{R}$.

- features:
  - linear: $x_i = \Sigma^{1/2} z_i$ where $z_i \in \mathbb{R}^p$ has i.i.d. components with mean 0 and variance 1 and finite 4^+ moments, Σ with bounded spectrum
  - non-linear: $x_i = \sigma(W z_i)$ where $z_i \in \mathbb{R}^d$ has i.i.d. components from $\mathcal{N}(0,1)$, $W \in \mathbb{R}^{p \times d}$ has i.i.d. entries from $\mathcal{N}(0, 1/d)$, and σ is an activation function of sub-exponential growth applied componentwise (non-linear features capture linearized two-layer neural network)
- response:
  - $y = X\beta_0 + \varepsilon$
  - $\beta_0 \in \mathbb{R}^p$ is an unknown parameter with bounded effective energy
  - ε independent of X with mean 0, covariance $\sigma^2 I_n$, finite 4^+ moments
- proportional asymptotics:
  - linear: $n, p \to \infty$ such that $p/n \to \gamma \in (0, \infty)$
  - non-linear: $n, d, p \to \infty$ such that $p/n \to \gamma \in (0, \infty)$ and $p/d \to \psi \in (1, \infty)$

Linearized overparametrized setting

Standard regression setting with n i.i.d. data $(x_i, y_i) \in \mathbb{R}^p \times \mathbb{R}$.

- features:
  - linear: $x_i = \Sigma^{1/2} z_i$ where $z_i \in \mathbb{R}^p$ has i.i.d. components with mean 0 and variance 1 and finite 4^+ moments, Σ with bounded spectrum
  - non-linear: $x_i = \sigma(W z_i)$ where $z_i \in \mathbb{R}^d$ has i.i.d. components from $\mathcal{N}(0,1)$, $W \in \mathbb{R}^{p \times d}$ has i.i.d. entries from $\mathcal{N}(0, 1/d)$, and σ is an activation function of sub-exponential growth applied componentwise (non-linear features capture linearized two-layer neural network)
- response:
  - $y = X\beta_0 + \varepsilon$
  - $\beta_0 \in \mathbb{R}^p$ is an unknown parameter with bounded effective energy
  - ε independent of X with mean 0, covariance $\sigma^2 I_n$, finite 4^+ moments
- proportional asymptotics:
  - linear: $n, p \to \infty$ such that $p/n \to \gamma \in (0, \infty)$
  - non-linear: $n, d, p \to \infty$ such that $p/n \to \gamma \in (0, \infty)$ and $p/d \to \psi \in (1, \infty)$

Linearized overparametrized setting

Standard regression setting with n i.i.d. data $(x_i, y_i) \in \mathbb{R}^p \times \mathbb{R}$.

- features:
  - linear: $x_i = \Sigma^{1/2} z_i$ where $z_i \in \mathbb{R}^p$ has i.i.d. components with mean 0 and variance 1 and finite 4^+ moments, Σ with bounded spectrum
  - non-linear: $x_i = \sigma(W z_i)$ where $z_i \in \mathbb{R}^d$ has i.i.d. components from $\mathcal{N}(0,1)$, $W \in \mathbb{R}^{p \times d}$ has i.i.d. entries from $\mathcal{N}(0, 1/d)$, and σ is an activation function of sub-exponential growth applied componentwise (non-linear features capture linearized two-layer neural network)
- response:
  - $y = X\beta_0 + \varepsilon$
  - $\beta_0 \in \mathbb{R}^p$ is an unknown parameter with bounded effective energy
  - ε independent of X with mean 0, covariance $\sigma^2 I_n$, finite 4^+ moments
- proportional asymptotics:
  - linear: $n, p \to \infty$ such that $p/n \to \gamma \in (0, \infty)$
  - non-linear: $n, d, p \to \infty$ such that $p/n \to \gamma \in (0, \infty)$ and $p/d \to \psi \in (1, \infty)$

Linearized overparametrized setting

Standard regression setting with n i.i.d. data $(x_i, y_i) \in \mathbb{R}^p \times \mathbb{R}$.

- features:
  - linear: $x_i = \Sigma^{1/2} z_i$ where $z_i \in \mathbb{R}^p$ has i.i.d. components with mean 0 and variance 1 and finite 4^+ moments, Σ with bounded spectrum
  - non-linear: $x_i = \sigma(W z_i)$ where $z_i \in \mathbb{R}^d$ has i.i.d. components from $\mathcal{N}(0,1)$, $W \in \mathbb{R}^{p \times d}$ has i.i.d. entries from $\mathcal{N}(0, 1/d)$, and σ is an activation function of sub-exponential growth applied componentwise (non-linear features capture linearized two-layer neural network)
- response:
  - $y = X\beta_0 + \varepsilon$
  - $\beta_0 \in \mathbb{R}^p$ is an unknown parameter with bounded effective energy
  - ε independent of X with mean 0, covariance $\sigma^2 I_n$, finite 4^+ moments
- proportional asymptotics:
  - linear: $n, p \to \infty$ such that $p/n \to \gamma \in (0, \infty)$
  - non-linear: $n, d, p \to \infty$ such that $p/n \to \gamma \in (0, \infty)$ and $p/d \to \psi \in (1, \infty)$

Main motivation and punchline of this work

- Given a tuning parameter λ, recall that ridge regression solves

$$\underset{\beta \in \mathbb{R}^p}{\text{minimize}} \sum_{i=1}^{n} (y_i - x_i^T \beta)^2 / n + \lambda \|\beta\|_2^2$$

- Choice λ crucially affects the performance of the fitted model
- Depending on the geometry of data features and underlying signal, optimal λ in the overparametrized setting can be 0 or even negative

Key question: how to select λ based on observed data in high dimensions

We show that under proportional asymptotics, almost surely, the leave-one-out and generalized cross-validation estimators

1. converge to out-of-sample prediction error uniformly in λ;
2. pick optimal λ for prediction error, including when $\lambda = 0$ or negative

Main motivation and punchline of this work

- Given a tuning parameter λ, recall that ridge regression solves

$$\underset{\beta \in \mathbb{R}^p}{\text{minimize}} \sum_{i=1}^{n} (y_i - x_i^T \beta)^2 / n + \lambda \|\beta\|_2^2$$

- Choice λ crucially affects the performance of the fitted model
- Depending on the geometry of data features and underlying signal, optimal λ in the overparametrized setting can be 0 or even negative

Key question: how to select λ based on observed data in high dimensions

We show that under proportional asymptotics, almost surely, the leave-one-out and generalized cross-validation estimators

1. converge to out-of-sample prediction error uniformly in λ;
2. pick optimal λ for prediction error, including when $\lambda = 0$ or negative

Main motivation and punchline of this work

- Given a tuning parameter λ, recall that ridge regression solves

$$\underset{\beta \in \mathbb{R}^p}{\text{minimize}} \sum_{i=1}^{n} (y_i - x_i^T \beta)^2 / n + \lambda \|\beta\|_2^2$$

- Choice λ crucially affects the performance of the fitted model
- Depending on the geometry of data features and underlying signal, optimal λ in the overparametrized setting can be 0 or even negative

Key question: how to select λ based on observed data in high dimensions

We show that under proportional asymptotics, almost surely, the leave-one-out and generalized cross-validation estimators

1. converge to out-of-sample prediction error uniformly in λ;
2. pick optimal λ for prediction error, including when $\lambda = 0$ or negative

Main motivation and punchline of this work

- Given a tuning parameter λ, recall that ridge regression solves

$$\underset{\beta \in \mathbb{R}^p}{\text{minimize}} \sum_{i=1}^{n} (y_i - x_i^T \beta)^2 / n + \lambda \|\beta\|_2^2$$

- Choice λ crucially affects the performance of the fitted model
- Depending on the geometry of data features and underlying signal, optimal λ in the overparametrized setting can be 0 or even negative

Key question: how to select λ based on observed data in high dimensions

We show that under proportional asymptotics, almost surely, the leave-one-out and generalized cross-validation estimators

1. converge to out-of-sample prediction error uniformly in λ;
2. pick optimal λ for prediction error, including when $\lambda = 0$ or negative

Main motivation and punchline of this work

- Given a tuning parameter λ, recall that ridge regression solves

$$\underset{\beta \in \mathbb{R}^p}{\text{minimize}} \sum_{i=1}^{n} (y_i - x_i^T \beta)^2 / n + \lambda \|\beta\|_2^2$$

- Choice λ crucially affects the performance of the fitted model
- Depending on the geometry of data features and underlying signal, optimal λ in the overparametrized setting can be 0 or even negative

Key question: how to select λ based on observed data in high dimensions

We show that under proportional asymptotics, almost surely, the leave-one-out and generalized cross-validation estimators

1. converge to out-of-sample prediction error uniformly in λ;
2. pick optimal λ for prediction error, including when $\lambda = 0$ or negative

Main motivation and punchline of this work

- Given a tuning parameter λ, recall that ridge regression solves

$$\underset{\beta \in \mathbb{R}^p}{\text{minimize}} \sum_{i=1}^{n} (y_i - x_i^T \beta)^2 / n + \lambda \|\beta\|_2^2$$

- Choice λ crucially affects the performance of the fitted model
- Depending on the geometry of data features and underlying signal, optimal λ in the overparametrized setting can be 0 or even negative

Key question: how to select λ based on observed data in high dimensions

We show that under proportional asymptotics, almost surely, the leave-one-out and generalized cross-validation estimators

1. converge to out-of-sample prediction error uniformly in λ;
2. pick optimal λ for prediction error, including when $\lambda = 0$ or negative

Outline

High-dimensional ridge regression

- Let $X \in \mathbb{R}^{n \times p}$ denote feature matrix, $y \in \mathbb{R}^n$ denote response vector
- Let $\widehat{\beta}_\lambda := \arg\min_{\beta \in \mathbb{R}^p} \|y - X\beta\|_2^2/n + \lambda\|\beta\|_2^2$ denote ridge solution
  - if $\lambda > 0$, problem convex in β and has an explicit solution:

$$\widehat{\beta}_\lambda = (X^T X/n + \lambda I_p)^{-1} X^T y/n$$

  - for any $\lambda \in \mathbb{R}$, extend using Moore-Penrose inverse:

$$\widehat{\beta}_\lambda = (X^T X/n + \lambda I_p)^{+} X^T y/n$$

  - when $\lambda = 0$, this reduces to least squares sol with minimum ℓ_2 norm; in particular, when $\text{rank}(X) = n \leq p$, the solution interpolates data, i.e. $X\widehat{\beta} = y$, and has minimum ℓ_2 norm among all interpolators

High-dimensional ridge regression

- Let $X \in \mathbb{R}^{n \times p}$ denote feature matrix, $y \in \mathbb{R}^n$ denote response vector
- Let $\widehat{\beta}_\lambda := \arg\min_{\beta \in \mathbb{R}^p} \|y - X\beta\|_2^2/n + \lambda\|\beta\|_2^2$ denote ridge solution
  - if $\lambda > 0$, problem convex in β and has an explicit solution:

$$\widehat{\beta}_\lambda = (X^T X/n + \lambda I_p)^{-1} X^T y/n$$

  - for any $\lambda \in \mathbb{R}$, extend using Moore-Penrose inverse:

$$\widehat{\beta}_\lambda = (X^T X/n + \lambda I_p)^{+} X^T y/n$$

  - when $\lambda = 0$, this reduces to least squares sol with minimum ℓ_2 norm; in particular, when $\text{rank}(X) = n \leq p$, the solution interpolates data, i.e. $X\widehat{\beta} = y$, and has minimum ℓ_2 norm among all interpolators

High-dimensional ridge regression

- Let $X \in \mathbb{R}^{n \times p}$ denote feature matrix, $y \in \mathbb{R}^n$ denote response vector
- Let $\widehat{\beta}_\lambda := \arg\min_{\beta \in \mathbb{R}^p} \|y - X\beta\|_2^2/n + \lambda\|\beta\|_2^2$ denote ridge solution
  - if $\lambda > 0$, problem convex in β and has an explicit solution:

$$\widehat{\beta}_\lambda = (X^T X/n + \lambda I_p)^{-1} X^T y/n$$

  - for any $\lambda \in \mathbb{R}$, extend using Moore-Penrose inverse:

$$\widehat{\beta}_\lambda = (X^T X/n + \lambda I_p)^{+} X^T y/n$$

  - when $\lambda = 0$, this reduces to least squares sol with minimum ℓ_2 norm; in particular, when $\text{rank}(X) = n \leq p$, the solution interpolates data, i.e. $X\widehat{\beta} = y$, and has minimum ℓ_2 norm among all interpolators

High-dimensional ridge regression

- Let $X \in \mathbb{R}^{n \times p}$ denote feature matrix, $y \in \mathbb{R}^n$ denote response vector
- Let $\widehat{\beta}_\lambda := \arg\min_{\beta \in \mathbb{R}^p} \|y - X\beta\|_2^2/n + \lambda\|\beta\|_2^2$ denote ridge solution
  - if $\lambda > 0$, problem convex in β and has an explicit solution:

$$\widehat{\beta}_\lambda = (X^T X/n + \lambda I_p)^{-1} X^T y/n$$

  - for any $\lambda \in \mathbb{R}$, extend using Moore-Penrose inverse:

$$\widehat{\beta}_\lambda = (X^T X/n + \lambda I_p)^{+} X^T y/n$$

  - when $\lambda = 0$, this reduces to least squares sol with minimum ℓ_2 norm; in particular, when $\text{rank}(X) = n \leq p$, the solution interpolates data, i.e. $X\widehat{\beta} = y$, and has minimum ℓ_2 norm among all interpolators

High-dimensional ridge regression

- Let $X \in \mathbb{R}^{n \times p}$ denote feature matrix, $y \in \mathbb{R}^n$ denote response vector
- Let $\widehat{\beta}_\lambda := \arg\min_{\beta \in \mathbb{R}^p} \|y - X\beta\|_2^2/n + \lambda\|\beta\|_2^2$ denote ridge solution
  - if $\lambda > 0$, problem convex in β and has an explicit solution:

$$\widehat{\beta}_\lambda = (X^T X/n + \lambda I_p)^{-1} X^T y/n$$

  - for any $\lambda \in \mathbb{R}$, extend using Moore-Penrose inverse:

$$\widehat{\beta}_\lambda = (X^T X/n + \lambda I_p)^{+} X^T y/n$$

  - when $\lambda = 0$, this reduces to least squares sol with minimum ℓ_2 norm; in particular, when $\text{rank}(X) = n \leq p$, the solution interpolates data, i.e. $X\widehat{\beta} = y$, and has minimum ℓ_2 norm among all interpolators

Prediction error and cross validation

- We measure the performance of fitted models $\widehat{\beta}_\lambda$ by their expected squared out-of-sample prediction error defined as

$$\mathrm{err}(\lambda) := \mathbb{E}_{x_0, y_0}\left[(y_0 - x_0^T \widehat{\beta}_\lambda)^2 \mid X, y\right],$$

  where (x_0, y_0) is test pair sampled from same training distribution
  - random (conditional on observed data X and y)
  - unknown (depends on characteristics of data generating distribution)
- Several estimators of prediction error:
  - k-fold cross validation (large bias when $k = 5$ or even when $k = 10$)
  - Generalized cross validation
  - Stein unbiased error estimate (in-sample prediction error)

We study the case when $k = n$ also called leave-one-out cross-validation, and generalized cross-validation

Prediction error and cross validation

- We measure the performance of fitted models $\widehat{\beta}_\lambda$ by their expected squared out-of-sample prediction error defined as

$$\mathrm{err}(\lambda) := \mathbb{E}_{x_0,y_0}\left[(y_0 - x_0^T\widehat{\beta}_\lambda)^2 \mid X, y\right],$$

  where (x_0, y_0) is test pair sampled from same training distribution
  - random (conditional on observed data X and y)
  - unknown (depends on characteristics of data generating distribution)
- Several estimators of prediction error:
  - k-fold cross validation (large bias when $k = 5$ or even when $k = 10$)
  - Generalized cross validation
  - Stein unbiased error estimate (in-sample prediction error)

We study the case when $k = n$ also called leave-one-out cross-validation, and generalized cross-validation

Prediction error and cross validation

- We measure the performance of fitted models $\widehat{\beta}_\lambda$ by their expected squared out-of-sample prediction error defined as

$$\mathrm{err}(\lambda) := \mathbb{E}_{x_0,y_0}\left[(y_0 - x_0^T\widehat{\beta}_\lambda)^2 \mid X, y\right],$$

  where (x_0, y_0) is test pair sampled from same training distribution
  - random (conditional on observed data X and y)
  - unknown (depends on characteristics of data generating distribution)
- Several estimators of prediction error:
  - k-fold cross validation (large bias when $k = 5$ or even when $k = 10$)
  - Generalized cross validation
  - Stein unbiased error estimate (in-sample prediction error)

We study the case when $k = n$ also called leave-one-out cross-validation, and generalized cross-validation

Prediction error and cross validation

- We measure the performance of fitted models $\widehat{\beta}_\lambda$ by their expected squared out-of-sample prediction error defined as

$$\mathrm{err}(\lambda) := \mathbb{E}_{x_0,y_0}\left[(y_0 - x_0^T\widehat{\beta}_\lambda)^2 \mid X, y\right],$$

  where (x_0, y_0) is test pair sampled from same training distribution
  - random (conditional on observed data X and y)
  - unknown (depends on characteristics of data generating distribution)
- Several estimators of prediction error:
  - k-fold cross validation (large bias when $k = 5$ or even when $k = 10$)
  - Generalized cross validation
  - Stein unbiased error estimate (in-sample prediction error)

We study the case when $k = n$ also called leave-one-out cross-validation, and generalized cross-validation

Leave-one-out and generalized cross-validation

- Leave-one-out cross-validation (LOOCV):
  - for every i, train on all data except (x_i, y_i), call the estimate $\widehat{\beta}_\lambda^{-i}$
  - compute test error on the i^{th} point and take average

$$\text{loo}(\lambda) = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - x_i^T\widehat{\beta}_\lambda^{-i}\right)^2$$

$$\overset{\text{(shortcut)}}{=} \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - x_i^T\widehat{\beta}_\lambda}{1 - [L_\lambda]_{ii}}\right)^2$$

where $L_\lambda = X(X^TX/n + \lambda I_p)^+X^T/n$ is the ridge smoothing matrix

- Generalized cross-validation (GCV)
  - same as leave-one-out shortcut but a single re-weighting

$$\text{gcv}(\lambda) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - x_i^T\widehat{\beta}_\lambda}{1 - \text{tr}[L_\lambda]/n}\right)^2$$

- When $\widehat{\beta}_\lambda$ is an interpolator, i.e. $L_\lambda = I_n$, both estimates are "0/0"; we then define the estimates as their respective limits as $\lambda \to 0$

Leave-one-out and generalized cross-validation

- Leave-one-out cross-validation (LOOCV):
  - for every i, train on all data except (x_i, y_i), call the estimate $\widehat{\beta}_\lambda^{-i}$
  - compute test error on the i^{th} point and take average

$$\text{loo}(\lambda) = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - x_i^T\widehat{\beta}_\lambda^{-i}\right)^2$$

$$\overset{\text{(shortcut)}}{=} \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - x_i^T\widehat{\beta}_\lambda}{1 - [L_\lambda]_{ii}}\right)^2$$

  where $L_\lambda = X(X^TX/n + \lambda I_p)^+X^T/n$ is the ridge smoothing matrix

- Generalized cross-validation (GCV)
  - same as leave-one-out shortcut but a single re-weighting

$$\text{gcv}(\lambda) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - x_i^T\widehat{\beta}_\lambda}{1 - \text{tr}[L_\lambda]/n}\right)^2$$

- When $\widehat{\beta}_\lambda$ is an interpolator, i.e. $L_\lambda = I_n$, both estimates are "0/0"; we then define the estimates as their respective limits as $\lambda \to 0$

Leave-one-out and generalized cross-validation

- Leave-one-out cross-validation (LOOCV):
  - for every i, train on all data except (x_i, y_i), call the estimate $\widehat{\beta}_\lambda^{-i}$
  - compute test error on the i^{th} point and take average

$$\text{loo}(\lambda) = \frac{1}{n}\sum_{i=1}^n \left(y_i - x_i^T \widehat{\beta}_\lambda^{-i}\right)^2$$

$$\overset{\text{(shortcut)}}{=} \frac{1}{n}\sum_{i=1}^n \left(\frac{y_i - x_i^T \widehat{\beta}_\lambda}{1 - [L_\lambda]_{ii}}\right)^2$$

  where $L_\lambda = X(X^TX/n + \lambda I_p)^+ X^T/n$ is the ridge smoothing matrix

- Generalized cross-validation (GCV)
  - same as leave-one-out shortcut but a single re-weighting

$$\text{gcv}(\lambda) = \frac{1}{n}\sum_{i=1}^n \left(\frac{y_i - x_i^T \widehat{\beta}_\lambda}{1 - \text{tr}[L_\lambda]/n}\right)^2$$

- When $\widehat{\beta}_\lambda$ is an interpolator, i.e. $L_\lambda = I_n$, both estimates are "0/0"; we then define the estimates as their respective limits as $\lambda \to 0$

Leave-one-out and generalized cross-validation

- Leave-one-out cross-validation (LOOCV):
  - for every i, train on all data except (x_i, y_i), call the estimate $\widehat{\beta}_\lambda^{-i}$
  - compute test error on the i^{th} point and take average

$$\text{loo}(\lambda) = \frac{1}{n}\sum_{i=1}^n \left(y_i - x_i^T \widehat{\beta}_\lambda^{-i}\right)^2$$

$$\overset{\text{(shortcut)}}{=} \frac{1}{n}\sum_{i=1}^n \left(\frac{y_i - x_i^T \widehat{\beta}_\lambda}{1 - [L_\lambda]_{ii}}\right)^2$$

  where $L_\lambda = X(X^TX/n + \lambda I_p)^+ X^T/n$ is the ridge smoothing matrix

- Generalized cross-validation (GCV)
  - same as leave-one-out shortcut but a single re-weighting

$$\text{gcv}(\lambda) = \frac{1}{n}\sum_{i=1}^n \left(\frac{y_i - x_i^T \widehat{\beta}_\lambda}{1 - \text{tr}[L_\lambda]/n}\right)^2$$

- When $\widehat{\beta}_\lambda$ is an interpolator, i.e. $L_\lambda = I_n$, both estimates are "0/0"; we then define the estimates as their respective limits as $\lambda \to 0$

Goals of the work

There are two main questions that we answer in this work:

1. How do $\mathrm{gcv}(\lambda)$ and $\mathrm{loo}(\lambda)$ compare to $\mathrm{err}(\lambda)$ as functions of λ?
2. How do $\mathrm{err}(\widehat{\lambda}_I^{\mathrm{gcv}})$ and $\mathrm{err}(\widehat{\lambda}_I^{\mathrm{loo}})$ compare to $\mathrm{err}(\lambda_I^\star)$ where $\lambda_I^\star$ denotes the optimal oracle ride tuning parameter

$$\lambda_I^\star = \underset{\lambda \in I \subseteq \mathbb{R}}{\arg\min}\ \mathrm{err}(\lambda),$$

and $\widehat{\lambda}_I^{\mathrm{gcv}}$ and $\widehat{\lambda}_I^{\mathrm{loo}}$ denote the corresponding tuning parameters that minimize GCV and LOOCV over an interval I?

Goals of the work

There are two main questions that we answer in this work:

1. How do $\mathrm{gcv}(\lambda)$ and $\mathrm{loo}(\lambda)$ compare to $\mathrm{err}(\lambda)$ as functions of λ?
2. How do $\mathrm{err}(\widehat{\lambda}_I^{\mathrm{gcv}})$ and $\mathrm{err}(\widehat{\lambda}_I^{\mathrm{loo}})$ compare to $\mathrm{err}(\lambda_I^\star)$ where $\lambda_I^\star$ denotes the optimal oracle ride tuning parameter

$$\lambda_I^\star = \underset{\lambda \in I \subseteq \mathbb{R}}{\arg\min}\ \mathrm{err}(\lambda),$$

and $\widehat{\lambda}_I^{\mathrm{gcv}}$ and $\widehat{\lambda}_I^{\mathrm{loo}}$ denote the corresponding tuning parameters that minimize GCV and LOOCV over an interval I?

Goals of the work

There are two main questions that we answer in this work:

1. How do $\mathrm{gcv}(\lambda)$ and $\mathrm{loo}(\lambda)$ compare to $\mathrm{err}(\lambda)$ as functions of λ?
2. How do $\mathrm{err}(\widehat{\lambda}_I^{\mathrm{gcv}})$ and $\mathrm{err}(\widehat{\lambda}_I^{\mathrm{loo}})$ compare to $\mathrm{err}(\lambda_I^\star)$ where $\lambda_I^\star$ denotes the optimal oracle ride tuning parameter

$$\lambda_I^\star = \underset{\lambda \in I \subseteq \mathbb{R}}{\arg\min}\ \mathrm{err}(\lambda),$$

and $\widehat{\lambda}_I^{\mathrm{gcv}}$ and $\widehat{\lambda}_I^{\mathrm{loo}}$ denote the corresponding tuning parameters that minimize GCV and LOOCV over an interval I?

Outline

Summary of main results

Under proportional asymptotics, and certain bounded norm conditions on β_0 and Σ, we show

1. GCV pointwise convergence
   - $\mathrm{gcv}(\lambda)$ converges to $\mathrm{err}(\lambda)$ pointwise in λ
2. GCV uniform convergences
   - convergence holds uniformly over compact intervals of λ including 0
3. LOOCV convergences
   - the analogous results hold for $\mathrm{loo}(\lambda)$ by relating it to $\mathrm{gcv}(\lambda)$
4. Optimal tuned prediction errors
   - both $\mathrm{err}(\widehat{\lambda}_l^{\mathrm{gcv}})$ and $\mathrm{err}(\widehat{\lambda}_l^{\mathrm{loo}})$ converge to $\mathrm{err}(\lambda_l^\star)$

Summary of main results

Under proportional asymptotics, and certain bounded norm conditions on β_0 and Σ, we show

1. GCV pointwise convergence
   - $\mathrm{gcv}(\lambda)$ converges to $\mathrm{err}(\lambda)$ pointwise in λ
2. GCV uniform convergences
   - convergence holds uniformly over compact intervals of λ including 0
3. LOOCV convergences
   - the analogous results hold for $\mathrm{loo}(\lambda)$ by relating it to $\mathrm{gcv}(\lambda)$
4. Optimal tuned prediction errors
   - both $\mathrm{err}(\widehat{\lambda}_l^{\mathrm{gcv}})$ and $\mathrm{err}(\widehat{\lambda}_l^{\mathrm{loo}})$ converge to $\mathrm{err}(\lambda_l^\star)$

Summary of main results

Under proportional asymptotics, and certain bounded norm conditions on β_0 and Σ, we show

1. GCV pointwise convergence
   - $\mathrm{gcv}(\lambda)$ converges to $\mathrm{err}(\lambda)$ pointwise in λ
2. GCV uniform convergences
   - convergence holds uniformly over compact intervals of λ including 0
3. LOOCV convergences
   - the analogous results hold for $\mathrm{loo}(\lambda)$ by relating it to $\mathrm{gcv}(\lambda)$
4. Optimal tuned prediction errors
   - both $\mathrm{err}(\widehat{\lambda}_l^{\mathrm{gcv}})$ and $\mathrm{err}(\widehat{\lambda}_l^{\mathrm{loo}})$ converge to $\mathrm{err}(\lambda_l^\star)$

Summary of main results

Under proportional asymptotics, and certain bounded norm conditions on β_0 and Σ, we show

1. GCV pointwise convergence
   - $\mathrm{gcv}(\lambda)$ converges to $\mathrm{err}(\lambda)$ pointwise in λ
2. GCV uniform convergences
   - convergence holds uniformly over compact intervals of λ including 0
3. LOOCV convergences
   - the analogous results hold for $\mathrm{loo}(\lambda)$ by relating it to $\mathrm{gcv}(\lambda)$
4. Optimal tuned prediction errors
   - both $\mathrm{err}(\widehat{\lambda}_l^{\mathrm{gcv}})$ and $\mathrm{err}(\widehat{\lambda}_l^{\mathrm{loo}})$ converge to $\mathrm{err}(\lambda_l^\star)$

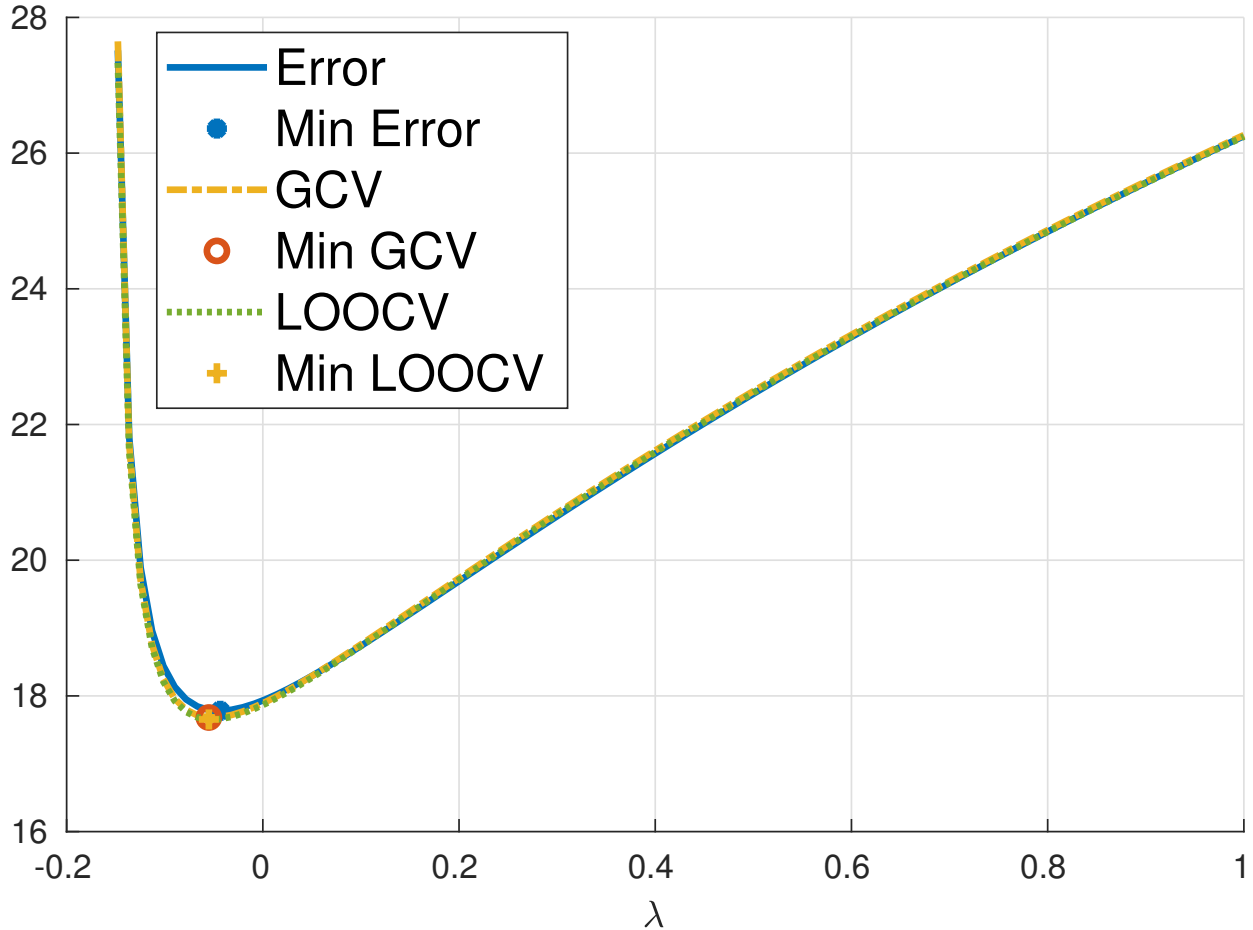
Summary of main results

Under proportional asymptotics, and certain bounded norm conditions on β_0 and Σ, we show

1. GCV pointwise convergence
   - $\mathrm{gcv}(\lambda)$ converges to $\mathrm{err}(\lambda)$ pointwise in λ
2. GCV uniform convergences
   - convergence holds uniformly over compact intervals of λ including 0
3. LOOCV convergences
   - the analogous results hold for $\mathrm{loo}(\lambda)$ by relating it to $\mathrm{gcv}(\lambda)$
4. Optimal tuned prediction errors
   - both $\mathrm{err}(\widehat{\lambda}_l^{\mathrm{gcv}})$ and $\mathrm{err}(\widehat{\lambda}_l^{\mathrm{loo}})$ converge to $\mathrm{err}(\lambda_l^\star)$

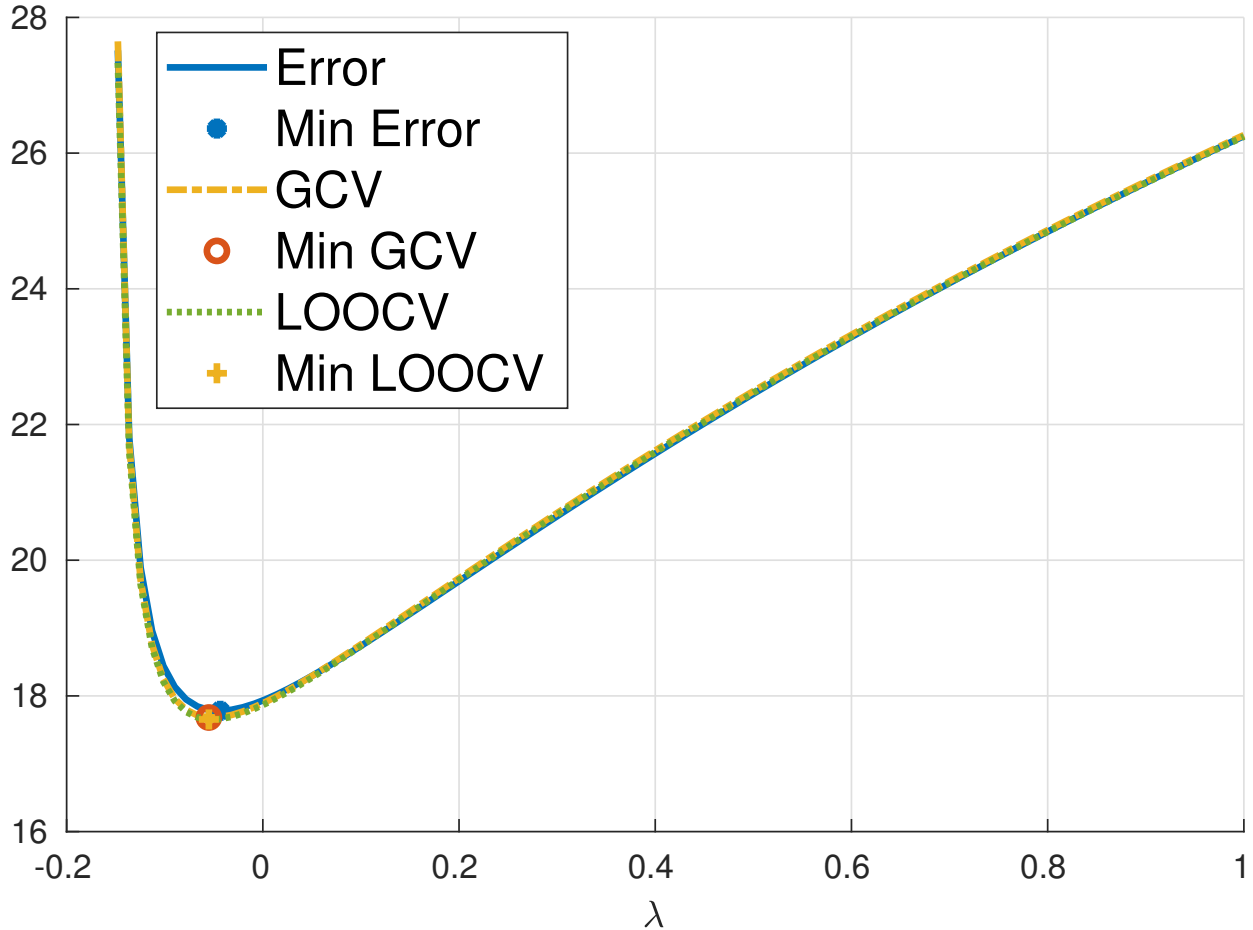
Outline

Linear features (negative optimal regularization)



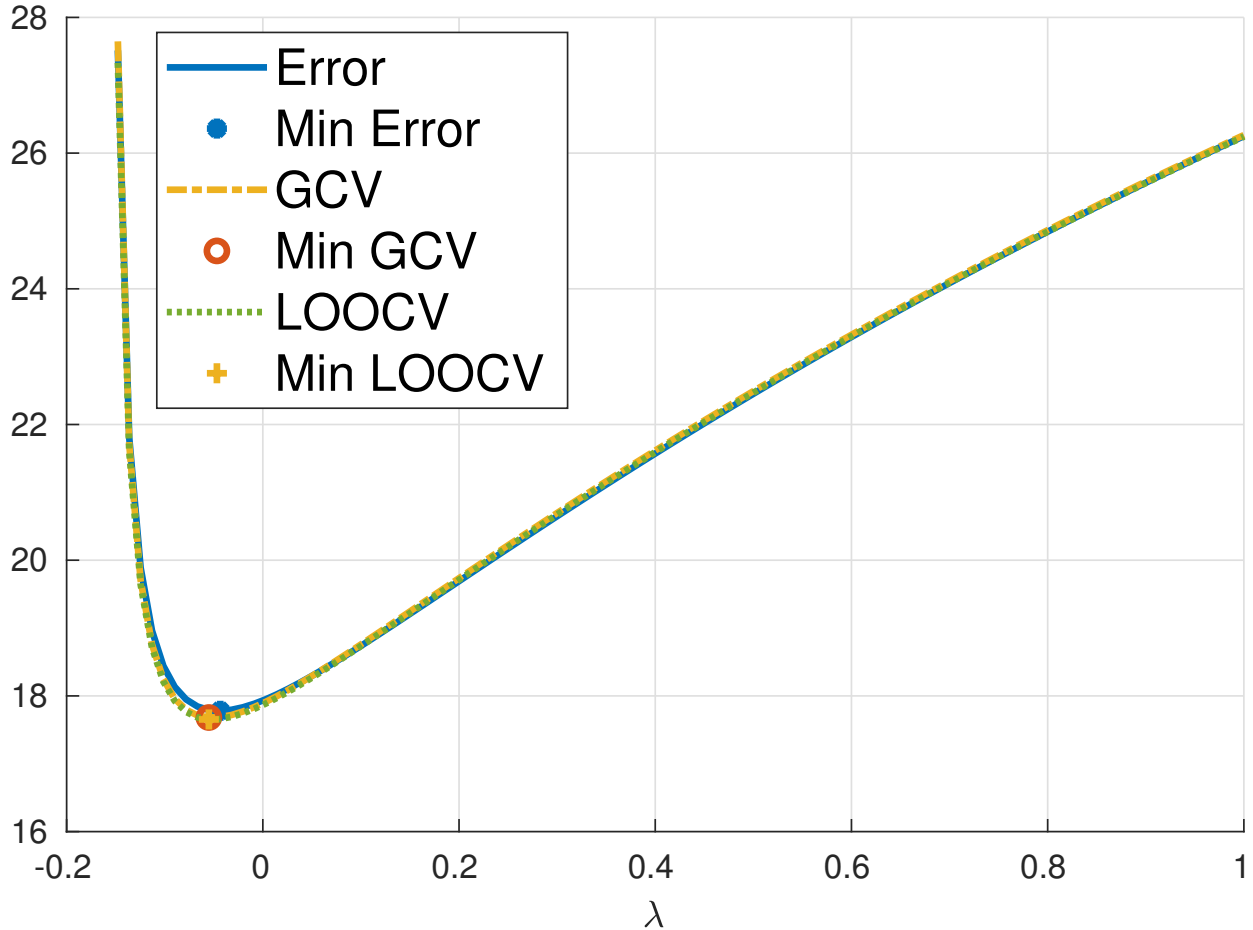
- Overparametrized regime ($p = 12000$, $n = 6000$)
- Autoregressive Σ
- β_0 aligned with the top eigendirection of Σ

Linear features (negative optimal regularization)



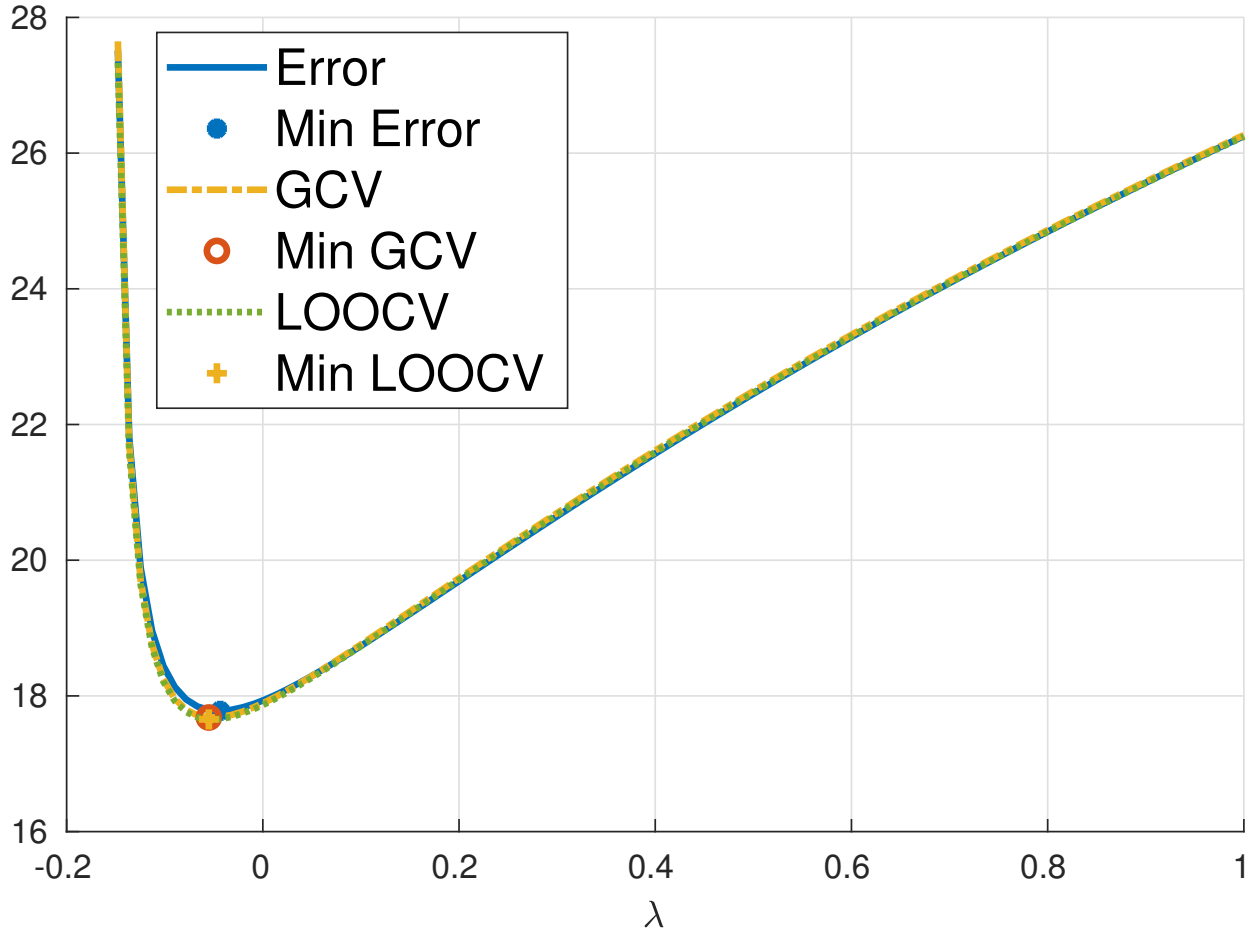
- Overparametrized regime ($p = 12000$, $n = 6000$)
- Autoregressive Σ
- β_0 aligned with the top eigendirection of Σ

Linear features (negative optimal regularization)



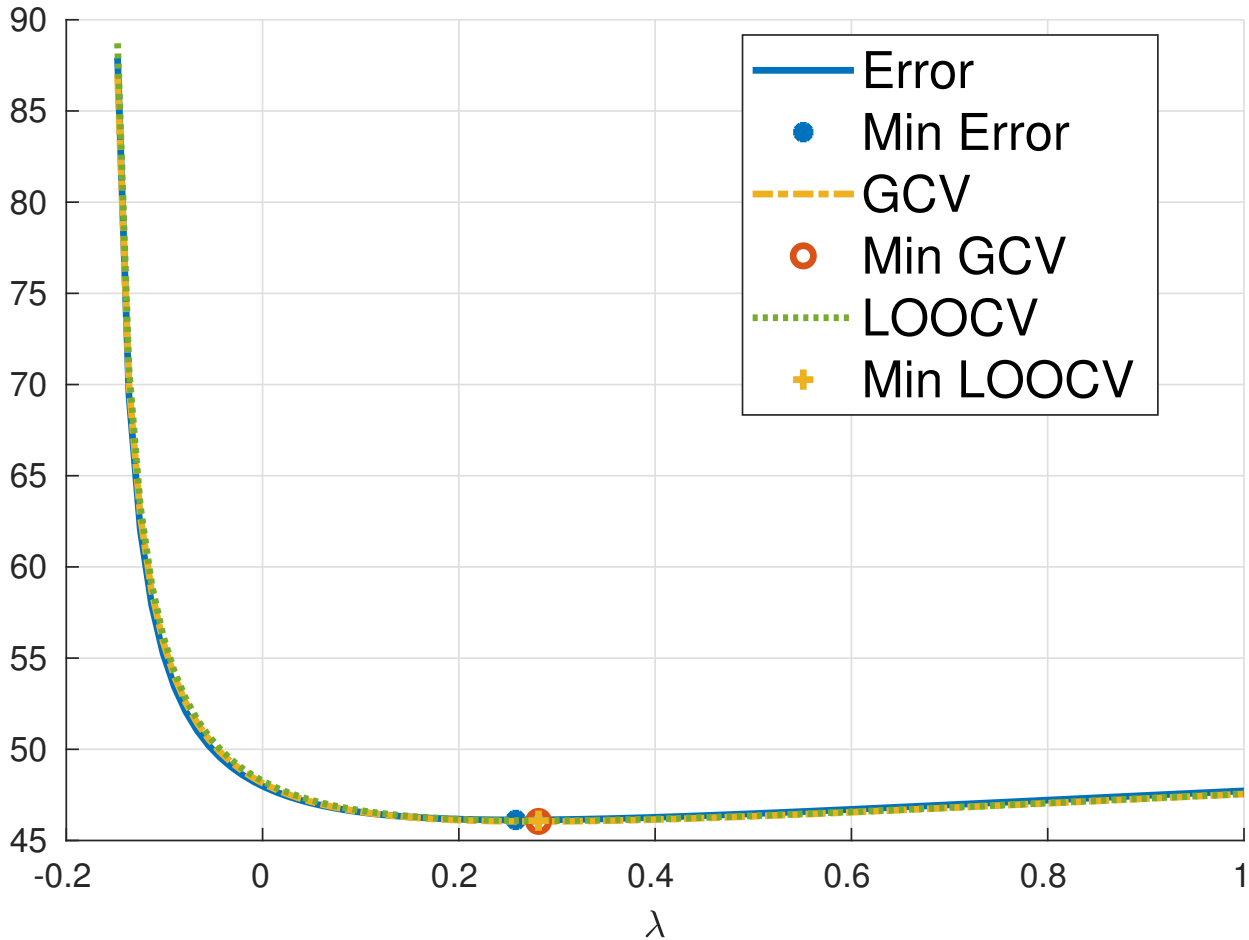
- Overparametrized regime ($p = 12000$, $n = 6000$)
- Autoregressive Σ
- β_0 aligned with the top eigendirection of Σ

Linear features (negative optimal regularization)



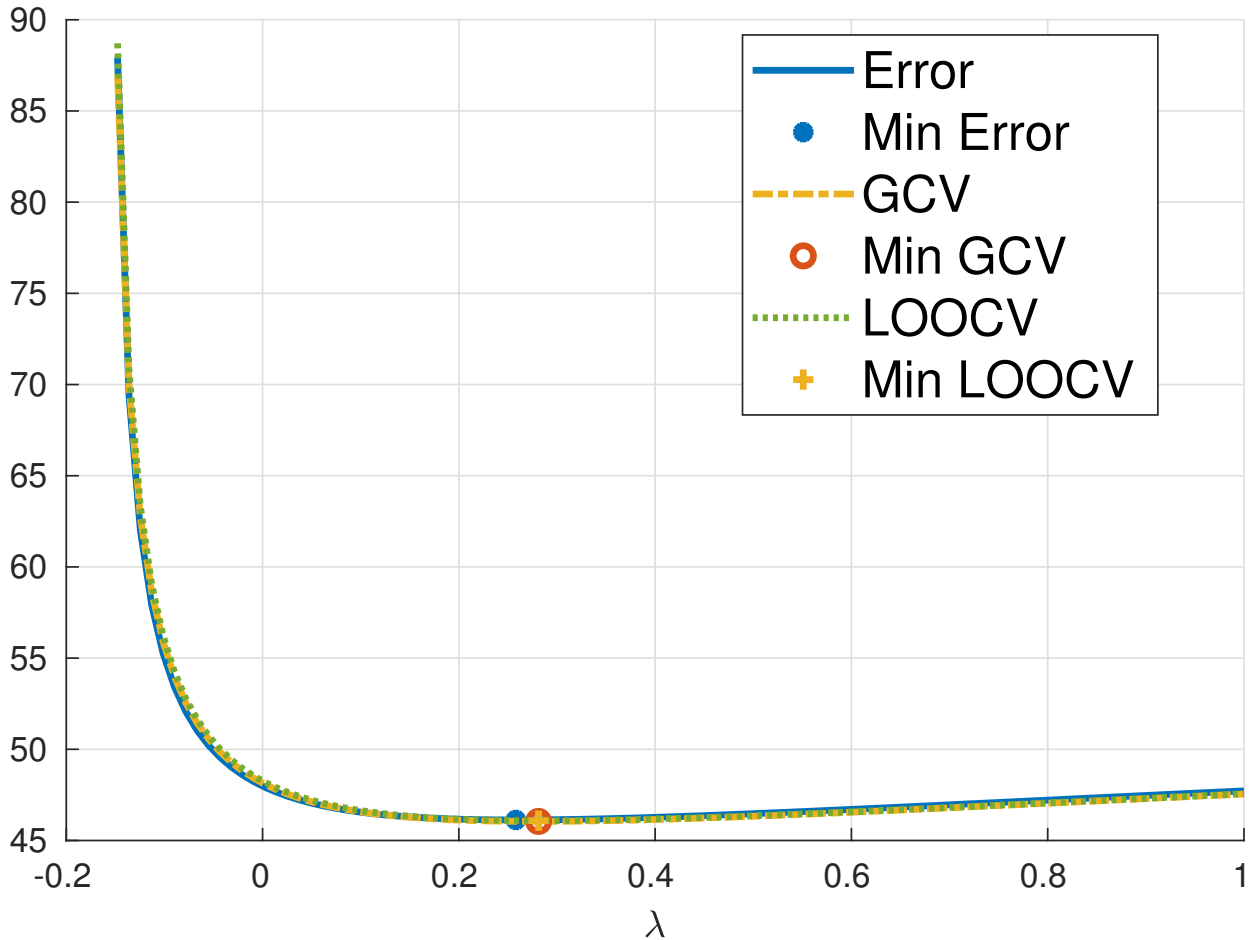
- Overparametrized regime ($p = 12000$, $n = 6000$)
- Autoregressive Σ
- β_0 aligned with the top eigendirection of Σ

Linear features (positive optimal regularization)



- Overparametrized regime ($p = 12000$, $n = 6000$)
- Autoregressive Σ
- β_0 aligned with the bottom eigendirection of Σ

Linear features (positive optimal regularization)



- Overparametrized regime ($p = 12000$, $n = 6000$)
- Autoregressive Σ
- β_0 aligned with the bottom eigendirection of Σ

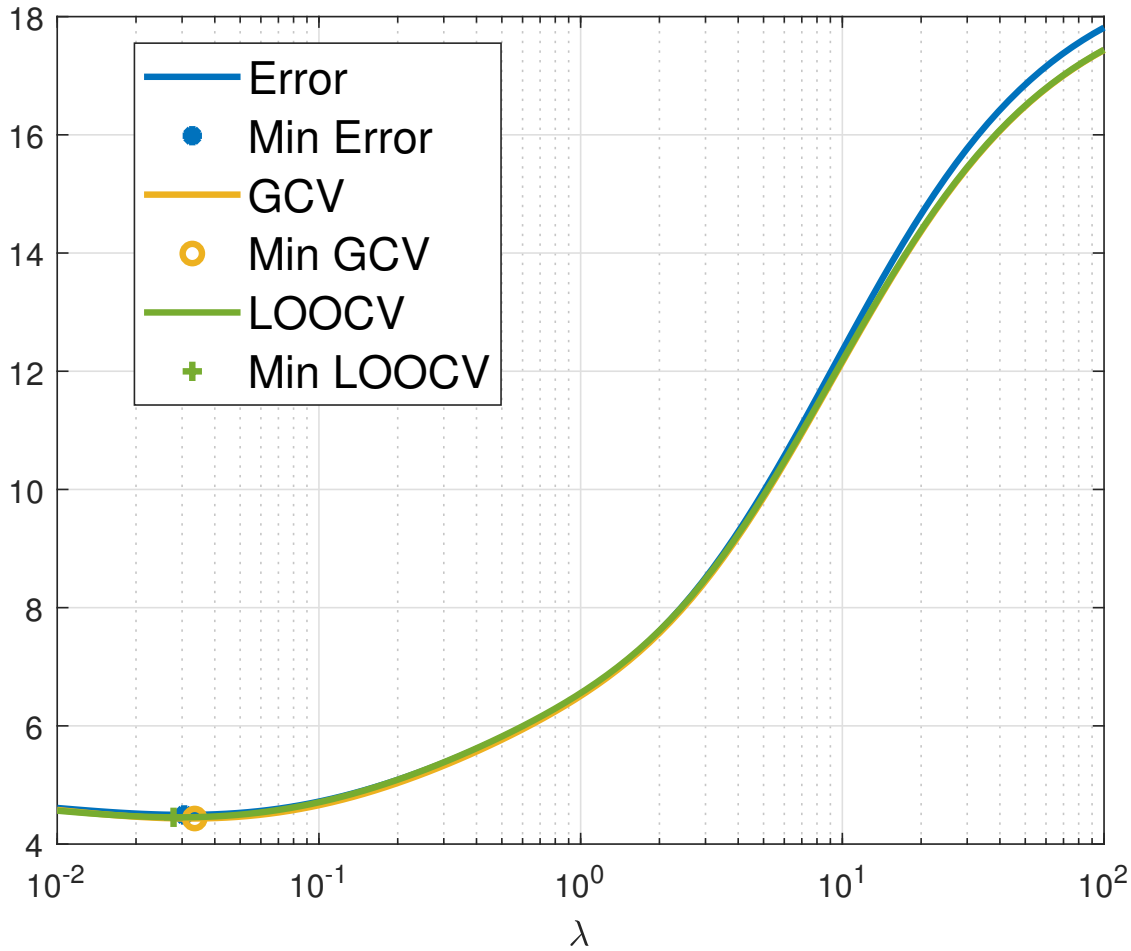
Non-linear features

- $x_i = \sigma(Wz_i)$, where $z_i \in \mathbb{R}^d$, $W \in \mathbb{R}^{p \times d}$ both random (two layer neural network with random first layer weights)
- σ is an activation function applied componentwise
  - σ is standardized when $\mathbb{E}[\sigma(G)] = 0$ and $\mathbb{E}[\sigma(G)^2] = 1$ for a standard normal $G \sim \mathcal{N}(0, 1)$
  - linear component of σ is $\mathbb{E}[G\sigma(G)]^2$
  - when the linear component is 0, σ is called purely non-linear
  - e.g. ReLU activation is not purely non-linear, while standardized absolute value activation is purely non-linear (any symmetric activation function can be made purely non-linear)

Non-linear features

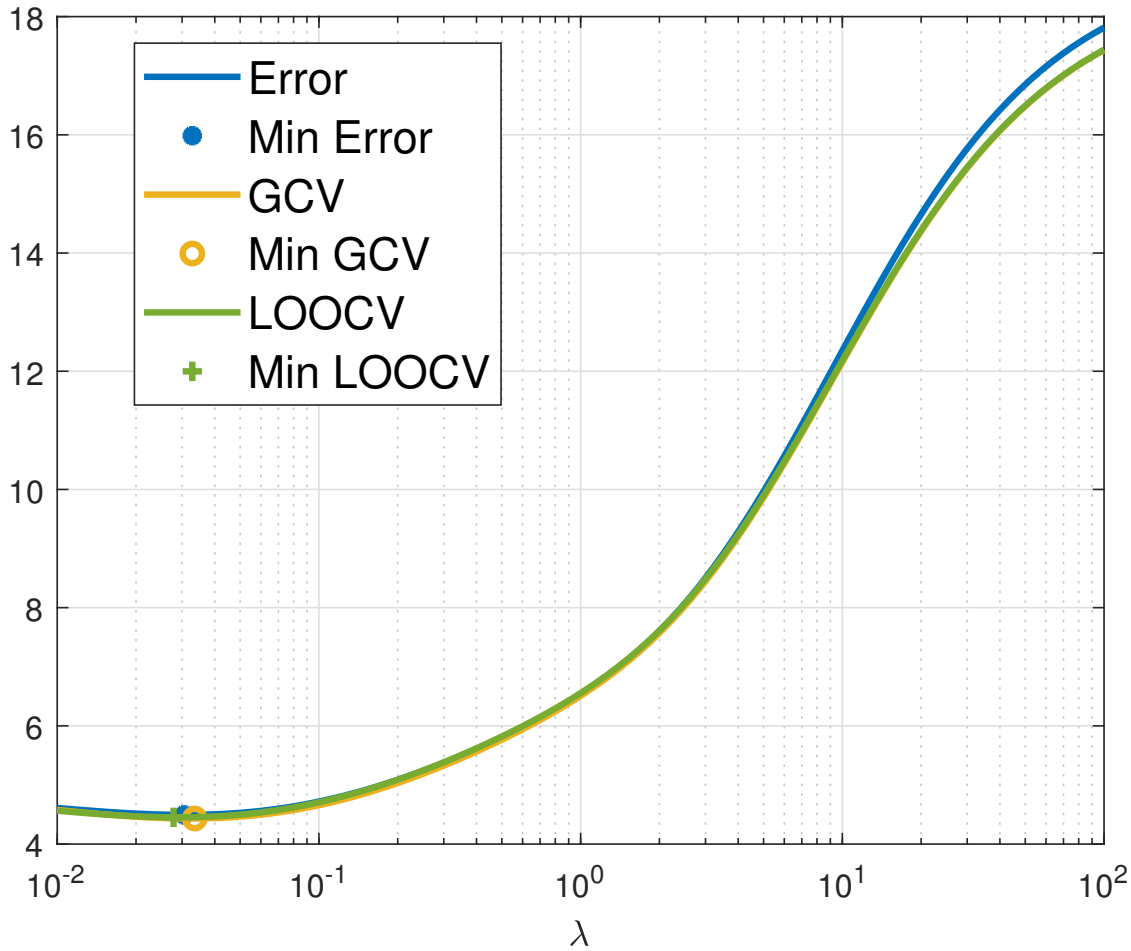
- $x_i = \sigma(Wz_i)$, where $z_i \in \mathbb{R}^d$, $W \in \mathbb{R}^{p \times d}$ both random (two layer neural network with random first layer weights)
- σ is an activation function applied componentwise
  - σ is standardized when $\mathbb{E}[\sigma(G)] = 0$ and $\mathbb{E}[\sigma(G)^2] = 1$ for a standard normal $G \sim \mathcal{N}(0, 1)$
  - linear component of σ is $\mathbb{E}[G\sigma(G)]^2$
  - when the linear component is 0, σ is called purely non-linear
  - e.g. ReLU activation is not purely non-linear, while standardized absolute value activation is purely non-linear (any symmetric activation function can be made purely non-linear)

ReLU features (near-zero optimal regularization)



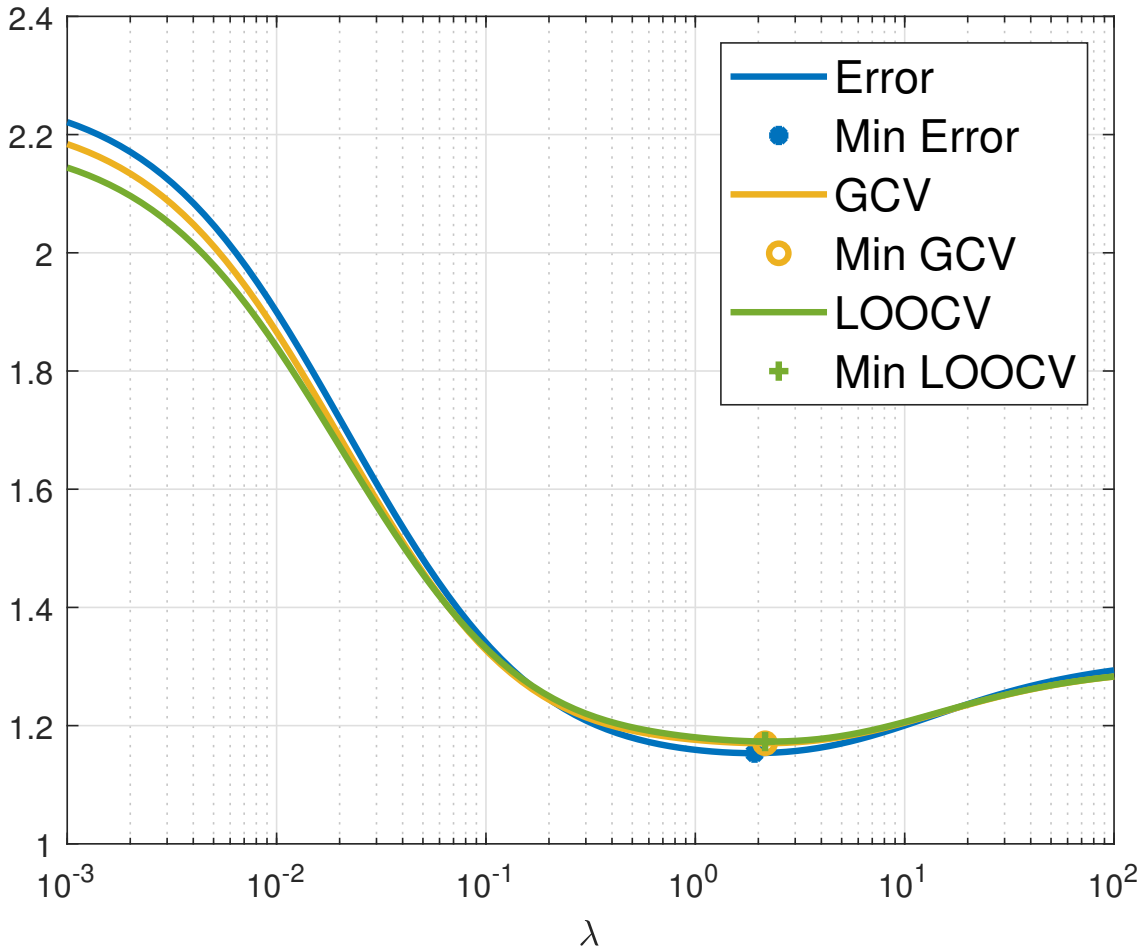
- $n = 1000$, $d = 100$, $p = 2000$
- High SNR case where optimal $\lambda \to 0$

ReLU features (near-zero optimal regularization)



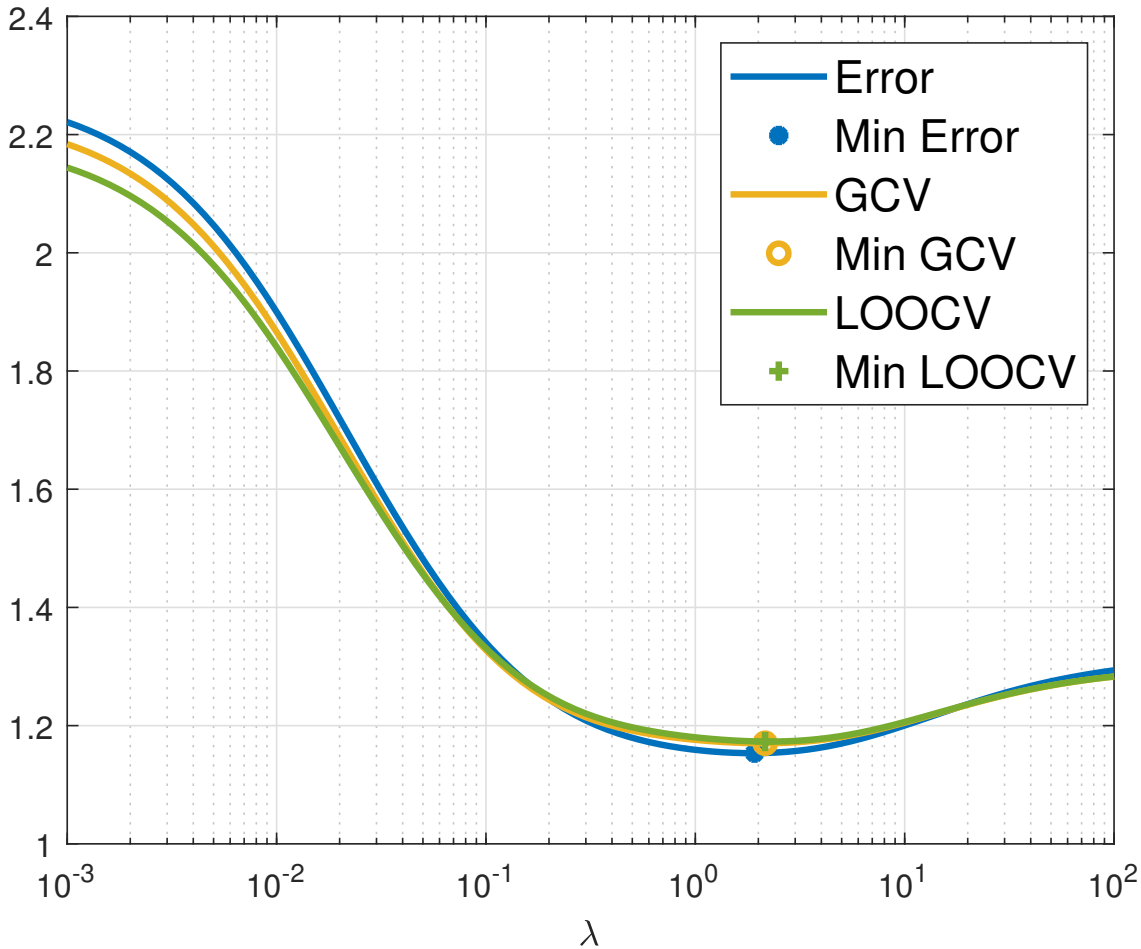
- $n = 1000$, $d = 100$, $p = 2000$
- High SNR case where optimal $\lambda \to 0$

ReLU features (positive optimal regularization)



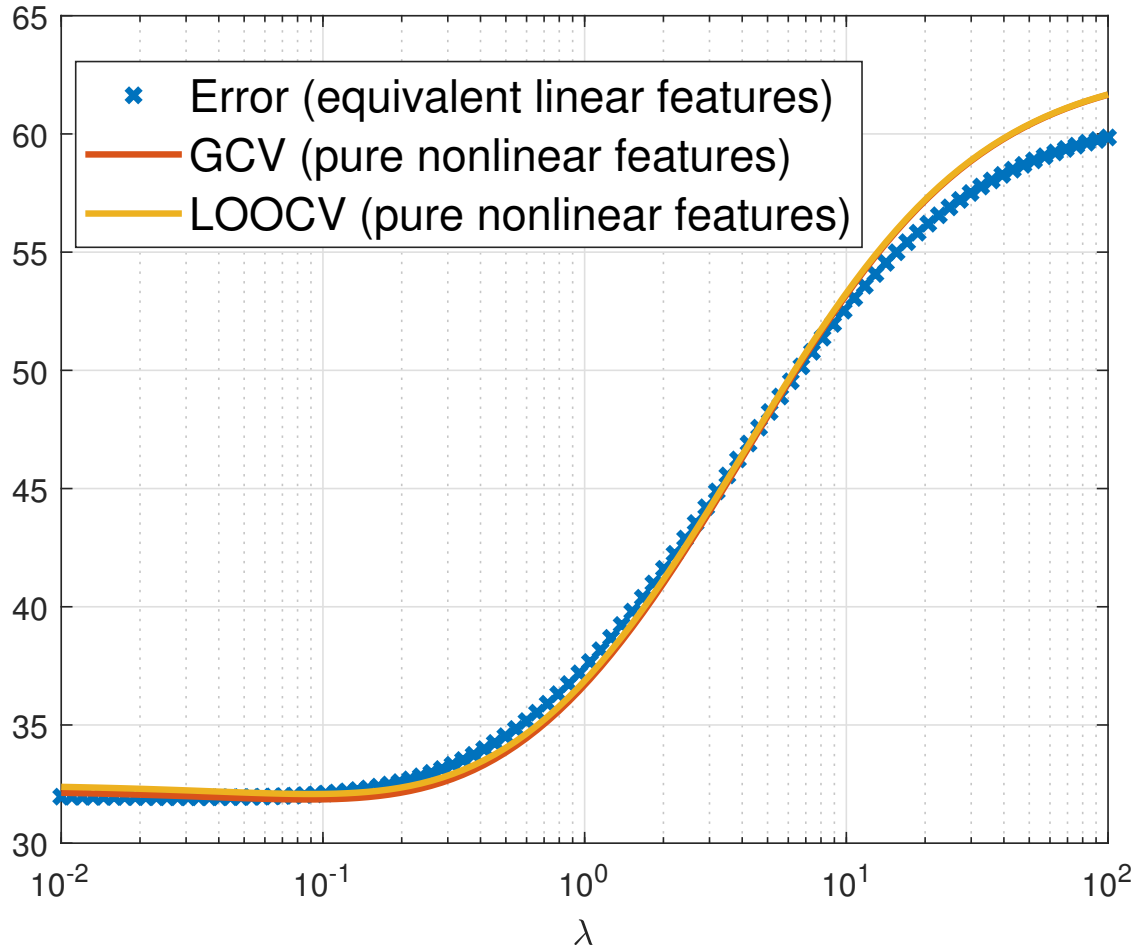
- $n = 1000$, $d = 100$, $p = 2000$
- Low SNR case where optimal $\lambda > 0$

ReLU features (positive optimal regularization)



- $n = 1000$, $d = 100$, $p = 2000$
- Low SNR case where optimal $\lambda > 0$

Purely non-linear features (linear features equivalence)



- $n = 1000$, $d = 500$, $p = 2000$
- Standardized absolute value activation

Outline

GCV versus prediction error: two key proof steps

Step 1: bias and variance decompositions of prediction error and GCV

Let $\widehat{\Sigma} := X^T X/n$ denote the sample covariance matrix.

- limiting bias-like components:
  - prediction error

$$\mathrm{err}_b(\lambda) := \lambda^2 \beta_0^T (\widehat{\Sigma} + \lambda I)^+ \Sigma (\widehat{\Sigma} + \lambda I)^+ \beta_0$$

  - gcv

$$\mathrm{gcv}_b(\lambda) := \frac{\lambda^2 \beta_0^T (\widehat{\Sigma} + \lambda I)^+ \widehat{\Sigma} (\widehat{\Sigma} + \lambda I)^+ \beta_0}{\left(1 - \mathrm{tr}\left[(\widehat{\Sigma} + \lambda I_p)^+ \widehat{\Sigma}\right]/n\right)^2}$$

- limiting variance-like components:
  - prediction error

$$\mathrm{err}_v(\lambda) := \sigma^2 \left[1 + \mathrm{tr}\left[(\widehat{\Sigma} + \lambda I_p)^+ \Sigma\right]/n\right] - \sigma^2 \,\mathrm{tr}\left[(\widehat{\Sigma} + \lambda I_p)^+ \Sigma (\widehat{\Sigma} + \lambda I_p)^+\right]/n$$

  - gcv

$$\mathrm{gcv}_v(\lambda) := \sigma^2 \left[\frac{1}{1 - \mathrm{tr}\left[(\widehat{\Sigma} + \lambda I_p)^+ \widehat{\Sigma}\right]/n}\right] - \frac{\sigma^2 \,\mathrm{tr}\left[(\widehat{\Sigma} + \lambda I_p)^+ \widehat{\Sigma} (\widehat{\Sigma} + \lambda I_p)^+\right]/n}{\left(1 - \mathrm{tr}\left[(\widehat{\Sigma} + \lambda I_p)^+ \widehat{\Sigma}\right]/n\right)^2}$$

GCV versus prediction error: two key proof steps

Step 1: bias and variance decompositions of prediction error and GCV

Let $\widehat{\Sigma} := X^T X/n$ denote the sample covariance matrix.

- limiting bias-like components:
  - prediction error

$$\mathrm{err}_b(\lambda) := \lambda^2 \beta_0^T (\widehat{\Sigma} + \lambda I)^+ \Sigma (\widehat{\Sigma} + \lambda I)^+ \beta_0$$

  - gcv

$$\mathrm{gcv}_b(\lambda) := \frac{\lambda^2 \beta_0^T (\widehat{\Sigma} + \lambda I)^+ \widehat{\Sigma} (\widehat{\Sigma} + \lambda I)^+ \beta_0}{\left(1 - \mathrm{tr}\left[(\widehat{\Sigma} + \lambda I_p)^+ \widehat{\Sigma}\right]/n\right)^2}$$

- limiting variance-like components:
  - prediction error

$$\mathrm{err}_v(\lambda) := \sigma^2 \left[1 + \mathrm{tr}\left[(\widehat{\Sigma} + \lambda I_p)^+ \Sigma\right]/n\right] - \sigma^2 \,\mathrm{tr}\left[(\widehat{\Sigma}+\lambda I_p)^+ \Sigma (\widehat{\Sigma}+\lambda I_p)^+\right]/n$$

  - gcv

$$\mathrm{gcv}_v(\lambda) := \sigma^2 \left[\frac{1}{1 - \mathrm{tr}\left[(\widehat{\Sigma} + \lambda I_p)^+ \widehat{\Sigma}\right]/n}\right] - \frac{\sigma^2 \,\mathrm{tr}\left[(\widehat{\Sigma} + \lambda I_p)^+ \widehat{\Sigma} (\widehat{\Sigma} + \lambda I_p)^+\right]/n}{\left(1 - \mathrm{tr}\left[(\widehat{\Sigma} + \lambda I_p)^+ \widehat{\Sigma}\right]/n\right)^2}$$

GCV versus prediction error: two key proof steps

Step 1: bias and variance decompositions of prediction error and GCV

Let $\widehat{\Sigma} := X^T X/n$ denote the sample covariance matrix.

- limiting bias-like components:
  - prediction error

$$\mathrm{err}_b(\lambda) := \lambda^2 \beta_0^T (\widehat{\Sigma} + \lambda I)^+ \Sigma (\widehat{\Sigma} + \lambda I)^+ \beta_0$$

  - gcv

$$\mathrm{gcv}_b(\lambda) := \frac{\lambda^2 \beta_0^T (\widehat{\Sigma} + \lambda I)^+ \widehat{\Sigma} (\widehat{\Sigma} + \lambda I)^+ \beta_0}{\left(1 - \mathrm{tr}\left[(\widehat{\Sigma} + \lambda I_p)^+ \widehat{\Sigma}\right]/n\right)^2}$$

- limiting variance-like components:
  - prediction error

$$\mathrm{err}_v(\lambda) := \sigma^2 \left[1 + \mathrm{tr}\left[(\widehat{\Sigma} + \lambda I_p)^+ \Sigma\right]/n\right] - \sigma^2 \,\mathrm{tr}\left[(\widehat{\Sigma}+\lambda I_p)^+ \Sigma (\widehat{\Sigma}+\lambda I_p)^+\right]/n$$

  - gcv

$$\mathrm{gcv}_v(\lambda) := \sigma^2 \left[\frac{1}{1 - \mathrm{tr}\left[(\widehat{\Sigma} + \lambda I_p)^+ \widehat{\Sigma}\right]/n}\right] - \frac{\sigma^2 \,\mathrm{tr}\left[(\widehat{\Sigma} + \lambda I_p)^+ \widehat{\Sigma} (\widehat{\Sigma} + \lambda I_p)^+\right]/n}{\left(1 - \mathrm{tr}\left[(\widehat{\Sigma} + \lambda I_p)^+ \widehat{\Sigma}\right]/n\right)^2}$$

GCV versus prediction error: two key proof steps

Step 1: bias and variance decompositions of prediction error and GCV

Let $\widehat{\Sigma} := X^T X/n$ denote the sample covariance matrix.

- limiting bias-like components:
  - prediction error

$$\mathrm{err}_b(\lambda) := \lambda^2 \beta_0^T (\widehat{\Sigma} + \lambda I)^+ \Sigma (\widehat{\Sigma} + \lambda I)^+ \beta_0$$

  - gcv

$$\mathrm{gcv}_b(\lambda) := \frac{\lambda^2 \beta_0^T (\widehat{\Sigma} + \lambda I)^+ \widehat{\Sigma} (\widehat{\Sigma} + \lambda I)^+ \beta_0}{\left(1 - \mathrm{tr}\left[(\widehat{\Sigma} + \lambda I_p)^+ \widehat{\Sigma}\right]/n\right)^2}$$

- limiting variance-like components:
  - prediction error

$$\mathrm{err}_v(\lambda) := \sigma^2 \left[1 + \mathrm{tr}\left[(\widehat{\Sigma} + \lambda I_p)^+ \Sigma\right]/n\right] - \sigma^2 \,\mathrm{tr}\left[(\widehat{\Sigma}+\lambda I_p)^+ \Sigma (\widehat{\Sigma}+\lambda I_p)^+\right]/n$$

  - gcv

$$\mathrm{gcv}_v(\lambda) := \sigma^2 \left[\frac{1}{1 - \mathrm{tr}\left[(\widehat{\Sigma} + \lambda I_p)^+ \widehat{\Sigma}\right]/n}\right] - \frac{\sigma^2 \,\mathrm{tr}\left[(\widehat{\Sigma} + \lambda I_p)^+ \widehat{\Sigma} (\widehat{\Sigma} + \lambda I_p)^+\right]/n}{\left(1 - \mathrm{tr}\left[(\widehat{\Sigma} + \lambda I_p)^+ \widehat{\Sigma}\right]/n\right)^2}$$

GCV versus prediction error: two key proof steps

Step 2: bias and variance equivalences for prediction error and GCV

- bias component equivalence:

$$\lambda^2 \beta_0^T (\widehat{\Sigma}+\lambda I)^+ \Sigma (\widehat{\Sigma}+\lambda I)^+ \beta_0 - \frac{\lambda^2 \beta_0^T (\widehat{\Sigma}+\lambda I)^+ \widehat{\Sigma} (\widehat{\Sigma}+\lambda I)^+ \beta_0}{\left(1-\operatorname{tr}\left[(\widehat{\Sigma}+\lambda I_p)^+ \widehat{\Sigma}\right]/n\right)^2} \xrightarrow{\text{a.s.}} 0$$

- variance component equivalences:

$$\sigma^2 \operatorname{tr}\left[(\widehat{\Sigma}+\lambda I_p)^+ \Sigma (\widehat{\Sigma}+\lambda I_p)^+\right]/n - \frac{\sigma^2 \operatorname{tr}\left[(\widehat{\Sigma}+\lambda I_p)^+ \widehat{\Sigma} (\widehat{\Sigma}+\lambda I_p)^+\right]/n}{\left(1-\operatorname{tr}\left[(\widehat{\Sigma}+\lambda I_p)^+ \widehat{\Sigma}\right]/n\right)^2} \xrightarrow{\text{a.s.}} 0$$

$$\sigma^2 \operatorname{tr}\left[(\widehat{\Sigma}+\lambda I_p)^+ \Sigma\right]/n - \frac{\sigma^2 \operatorname{tr}\left[(\widehat{\Sigma}+\lambda I_p)^+ \widehat{\Sigma}\right]/n}{1-\operatorname{tr}\left[(\widehat{\Sigma}+\lambda I_p)^+ \widehat{\Sigma}\right]/n} \xrightarrow{\text{a.s.}} 0$$

Main message: the GCV denominator proves to be the right correction for for the excess optimism in the biased GCV numerator of training error

GCV versus prediction error: two key proof steps

Step 2: bias and variance equivalences for prediction error and GCV

- bias component equivalence:

$$\lambda^2 \beta_0^T (\widehat{\Sigma}+\lambda I)^+ \Sigma (\widehat{\Sigma}+\lambda I)^+ \beta_0 - \frac{\lambda^2 \beta_0^T (\widehat{\Sigma}+\lambda I)^+ \widehat{\Sigma} (\widehat{\Sigma}+\lambda I)^+ \beta_0}{\left(1-\operatorname{tr}\left[(\widehat{\Sigma}+\lambda I_p)^+ \widehat{\Sigma}\right]/n\right)^2} \xrightarrow{\text{a.s.}} 0$$

- variance component equivalences:

$$\sigma^2 \operatorname{tr}\left[(\widehat{\Sigma}+\lambda I_p)^+ \Sigma (\widehat{\Sigma}+\lambda I_p)^+\right]/n - \frac{\sigma^2 \operatorname{tr}\left[(\widehat{\Sigma}+\lambda I_p)^+ \widehat{\Sigma} (\widehat{\Sigma}+\lambda I_p)^+\right]/n}{\left(1-\operatorname{tr}\left[(\widehat{\Sigma}+\lambda I_p)^+ \widehat{\Sigma}\right]/n\right)^2} \xrightarrow{\text{a.s.}} 0$$

$$\sigma^2 \operatorname{tr}\left[(\widehat{\Sigma}+\lambda I_p)^+ \Sigma\right]/n - \frac{\sigma^2 \operatorname{tr}\left[(\widehat{\Sigma}+\lambda I_p)^+ \widehat{\Sigma}\right]/n}{1-\operatorname{tr}\left[(\widehat{\Sigma}+\lambda I_p)^+ \widehat{\Sigma}\right]/n} \xrightarrow{\text{a.s.}} 0$$

Main message: the GCV denominator proves to be the right correction for for the excess optimism in the biased GCV numerator of training error

GCV versus prediction error: two key proof steps

Step 2: bias and variance equivalences for prediction error and GCV

- bias component equivalence:

$$\lambda^2 \beta_0^T (\widehat{\Sigma}+\lambda I)^+ \Sigma (\widehat{\Sigma}+\lambda I)^+ \beta_0 - \frac{\lambda^2 \beta_0^T (\widehat{\Sigma}+\lambda I)^+ \widehat{\Sigma} (\widehat{\Sigma}+\lambda I)^+ \beta_0}{\left(1-\mathrm{tr}\left[(\widehat{\Sigma}+\lambda I_p)^+ \widehat{\Sigma}\right]/n\right)^2} \xrightarrow{\text{a.s.}} 0$$

- variance component equivalences:

$$\sigma^2 \,\mathrm{tr}\left[(\widehat{\Sigma}+\lambda I_p)^+ \Sigma (\widehat{\Sigma}+\lambda I_p)^+\right]/n - \frac{\sigma^2 \,\mathrm{tr}\left[(\widehat{\Sigma}+\lambda I_p)^+ \widehat{\Sigma} (\widehat{\Sigma}+\lambda I_p)^+\right]/n}{\left(1-\mathrm{tr}\left[(\widehat{\Sigma}+\lambda I_p)^+ \widehat{\Sigma}\right]/n\right)^2} \xrightarrow{\text{a.s.}} 0$$

$$\sigma^2 \,\mathrm{tr}\left[(\widehat{\Sigma}+\lambda I_p)^+ \Sigma\right]/n - \frac{\sigma^2 \,\mathrm{tr}\left[(\widehat{\Sigma}+\lambda I_p)^+ \widehat{\Sigma}\right]/n}{1-\mathrm{tr}\left[(\widehat{\Sigma}+\lambda I_p)^+ \widehat{\Sigma}\right]/n} \xrightarrow{\text{a.s.}} 0$$

Main message: the GCV denominator proves to be the right correction for for the excess optimism in the biased GCV numerator of training error

GCV versus prediction error: two key proof steps

Step 2: bias and variance equivalences for prediction error and GCV

- bias component equivalence:

$$\lambda^2 \beta_0^T (\widehat{\Sigma}+\lambda I)^+ \Sigma (\widehat{\Sigma}+\lambda I)^+ \beta_0 - \frac{\lambda^2 \beta_0^T (\widehat{\Sigma}+\lambda I)^+ \widehat{\Sigma} (\widehat{\Sigma}+\lambda I)^+ \beta_0}{\left(1-\mathrm{tr}\left[(\widehat{\Sigma}+\lambda I_p)^+ \widehat{\Sigma}\right]/n\right)^2} \xrightarrow{\text{a.s.}} 0$$

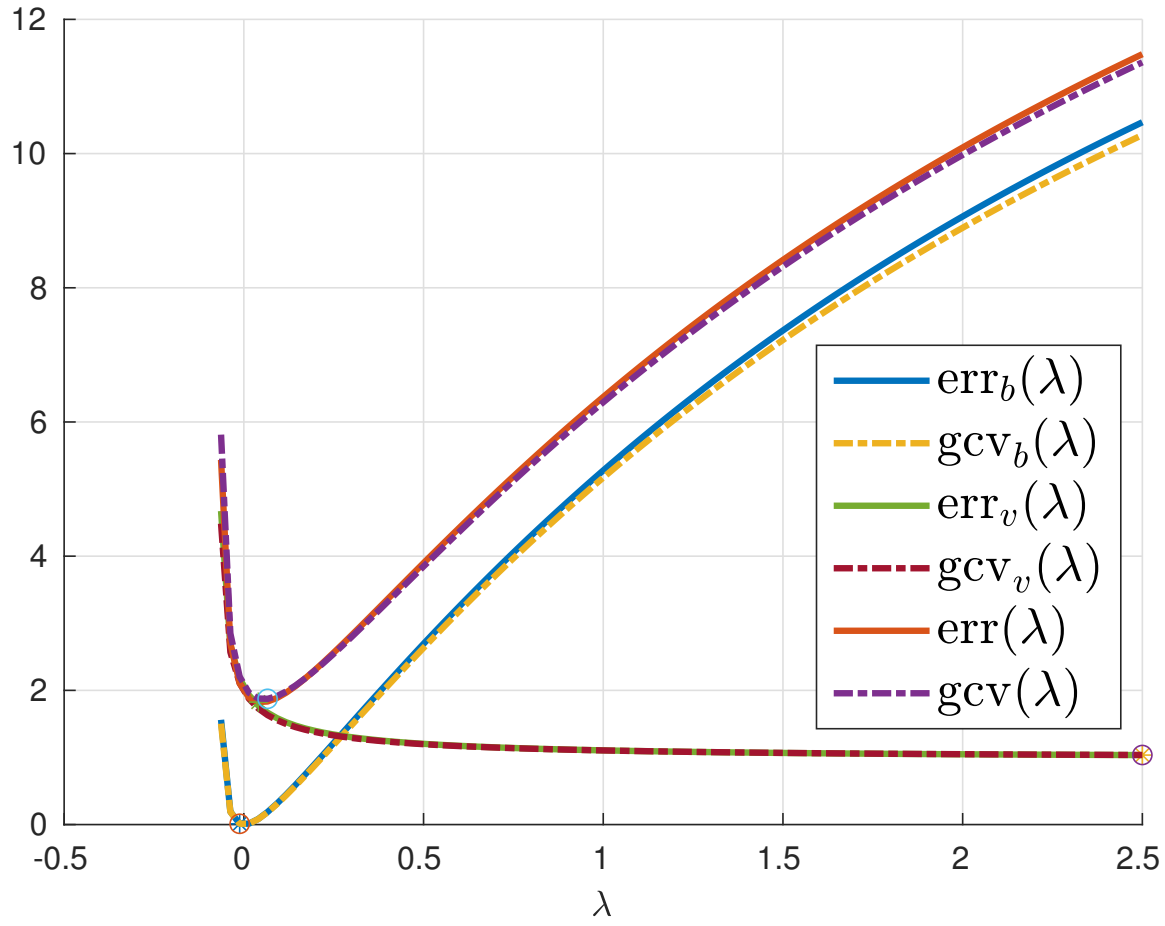
- variance component equivalences:

$$\sigma^2 \,\mathrm{tr}\left[(\widehat{\Sigma}+\lambda I_p)^+ \Sigma (\widehat{\Sigma}+\lambda I_p)^+\right]/n - \frac{\sigma^2 \,\mathrm{tr}\left[(\widehat{\Sigma}+\lambda I_p)^+ \widehat{\Sigma} (\widehat{\Sigma}+\lambda I_p)^+\right]/n}{\left(1-\mathrm{tr}\left[(\widehat{\Sigma}+\lambda I_p)^+ \widehat{\Sigma}\right]/n\right)^2} \xrightarrow{\text{a.s.}} 0$$

$$\sigma^2 \,\mathrm{tr}\left[(\widehat{\Sigma}+\lambda I_p)^+ \Sigma\right]/n - \frac{\sigma^2 \,\mathrm{tr}\left[(\widehat{\Sigma}+\lambda I_p)^+ \widehat{\Sigma}\right]/n}{1-\mathrm{tr}\left[(\widehat{\Sigma}+\lambda I_p)^+ \widehat{\Sigma}\right]/n} \xrightarrow{\text{a.s.}} 0$$

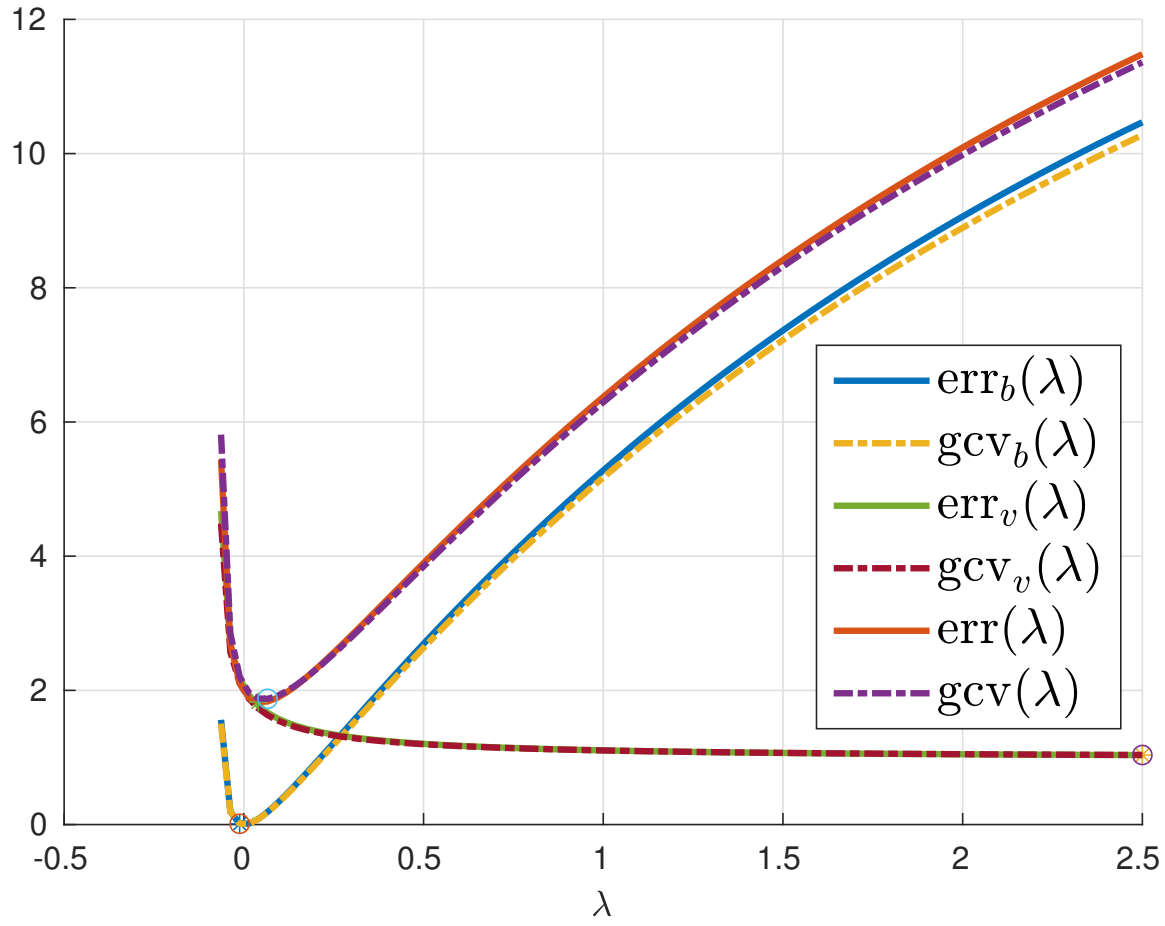
Main message: the GCV denominator proves to be the right correction for for the excess optimism in the biased GCV numerator of training error

Bias and variance equivalence numerical illustration



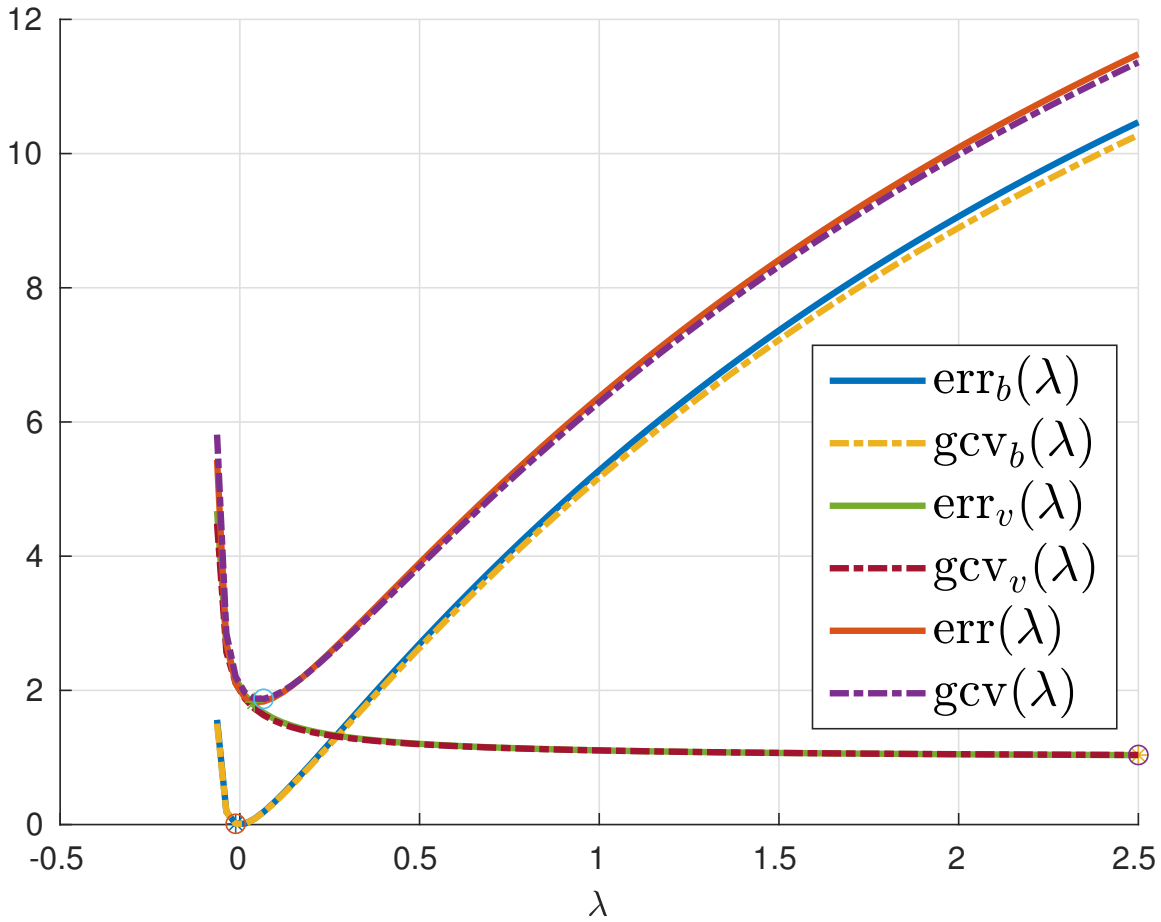
- Underparametrized ($n = 6000$, $p = 3000$)
- Bias minimized at $\lambda = 0$ and variance decreases as λ increases
- Optimal λ always positive is underparametrized regime!

Bias and variance equivalence numerical illustration



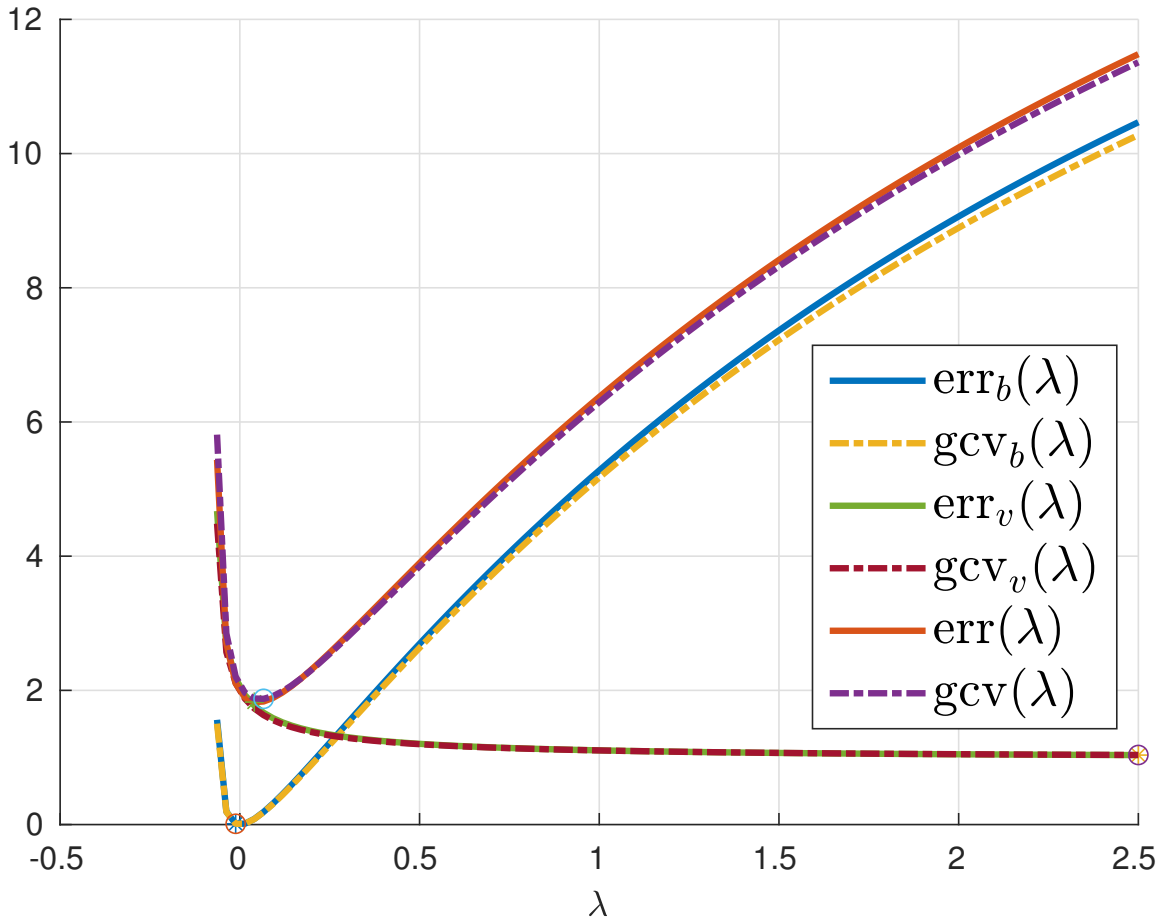
- Underparametrized ($n = 6000$, $p = 3000$)
- Bias minimized at $\lambda = 0$ and variance decreases as λ increases
- Optimal λ always positive is underparametrized regime!

Bias and variance equivalence numerical illustration



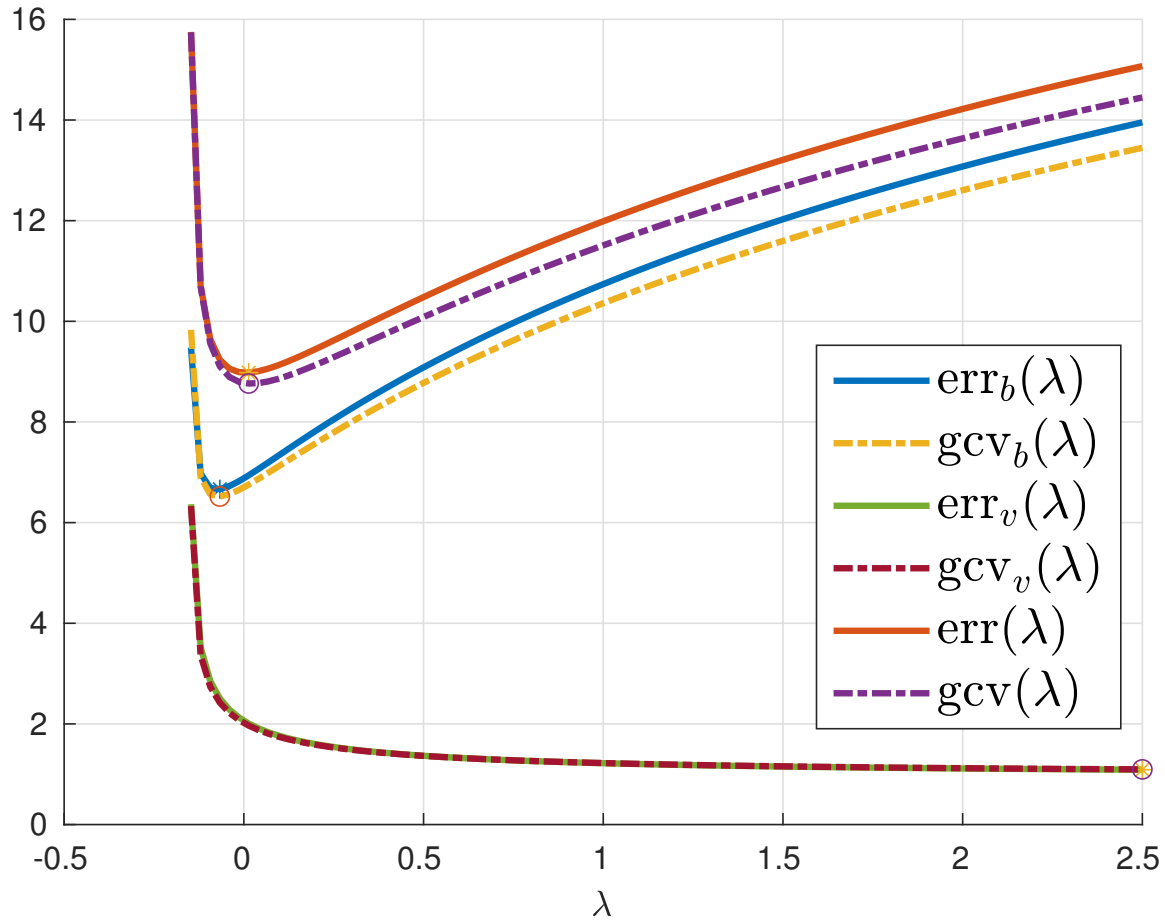
- Underparametrized ($n = 6000$, $p = 3000$)
- Bias minimized at $\lambda = 0$ and variance decreases as λ increases
- Optimal λ always positive is underparametrized regime!

Bias and variance equivalence numerical illustration



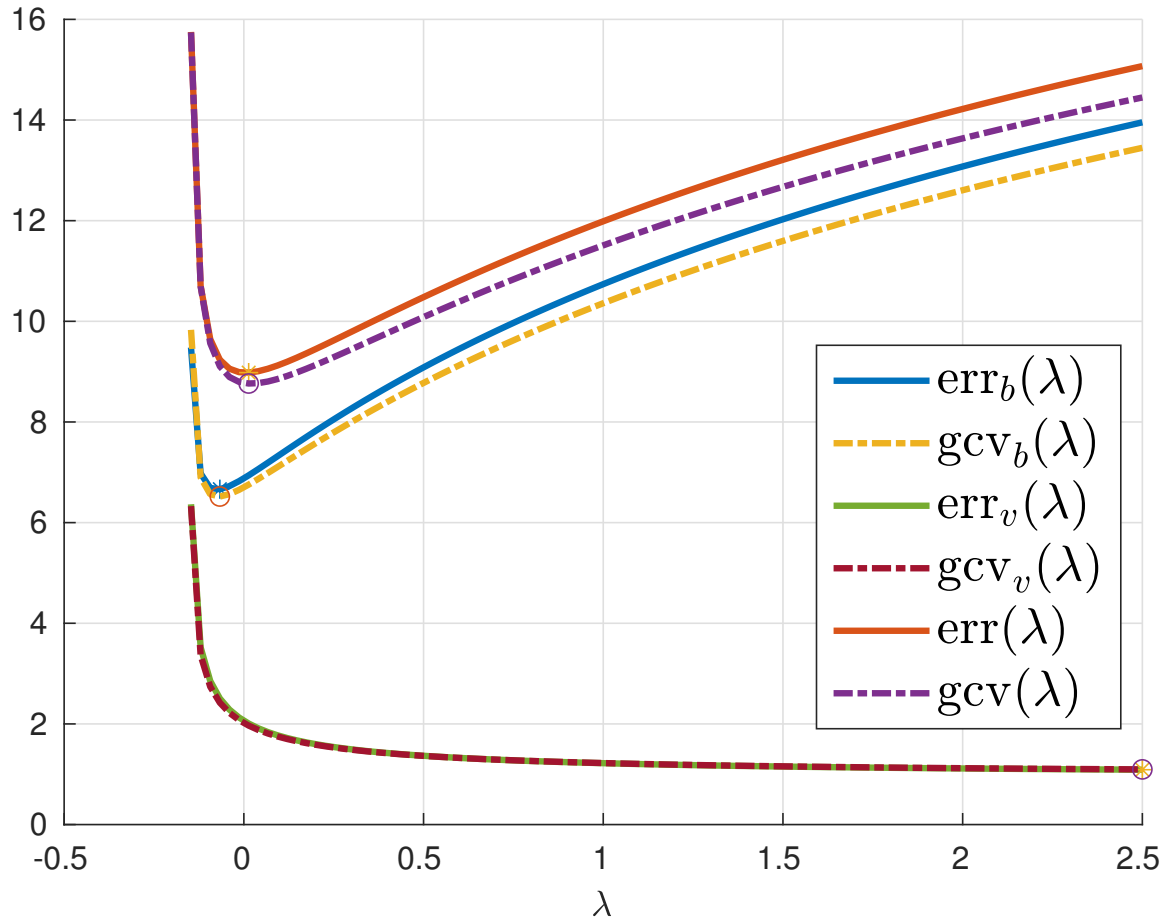
- Underparametrized ($n = 6000$, $p = 3000$)
- Bias minimized at $\lambda = 0$ and variance decreases as λ increases
- Optimal λ always positive is underparametrized regime!

Bias and variance equivalence numerical illustration



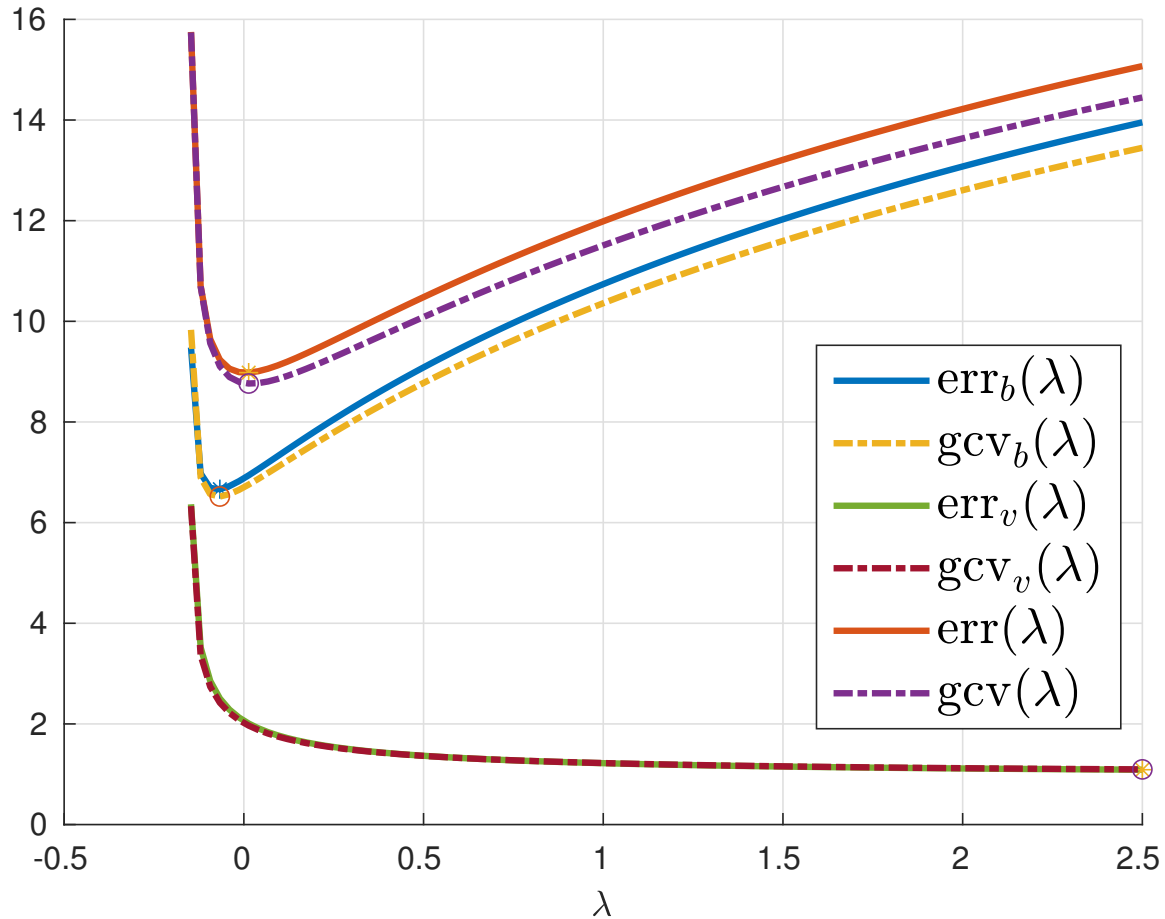
- Overparametrized ($p = 12000$, $n = 6000$)
- Bias no longer minimized $\lambda = 0$ and variance still decreasing in λ
- Optimal λ may be negative in overparametrized regime!

Bias and variance equivalence numerical illustration



- Overparametrized ($p = 12000$, $n = 6000$)
- Bias no longer minimized $\lambda = 0$ and variance still decreasing in λ
- Optimal λ may be negative in overparametrized regime!

Bias and variance equivalence numerical illustration



- Overparametrized ($p = 12000$, $n = 6000$)
- Bias no longer minimized $\lambda = 0$ and variance still decreasing in λ
- Optimal λ may be negative in overparametrized regime!

Discussion and future directions

This work shows GCV and LOOCV uniformly track squared out-of-sample prediction error for ridge regression under proportional asymptotics.

Main tool:

$$(\widehat{\Sigma} + \lambda I_p)^+ \Sigma \asymp \frac{(\widehat{\Sigma} + \lambda I_p)^+ \widehat{\Sigma}}{1 - \mathrm{tr}[(\widehat{\Sigma} + \lambda I_p)^+ \widehat{\Sigma}]/n}$$

where for any two sequences of matrices A_p and B_p, $A_p \asymp B_p$ is used to mean $\mathrm{tr}[C_p(A_p - B_p)] \xrightarrow{\text{a.s.}} 0$ for any deterministic seq of matrices C_p of bnd trace norm

Going beyond . . .

- Equivalences for general functionals of out-of-sample distributions
- Equivalences for ridge variants and general estimators
- Finite sample analysis and rates of convergence

⋮

Discussion and future directions

This work shows GCV and LOOCV uniformly track squared out-of-sample prediction error for ridge regression under proportional asymptotics.

Main tool:

$$(\widehat{\Sigma} + \lambda I_p)^+ \Sigma \asymp \frac{(\widehat{\Sigma} + \lambda I_p)^+ \widehat{\Sigma}}{1 - \mathrm{tr}[(\widehat{\Sigma} + \lambda I_p)^+ \widehat{\Sigma}]/n}$$

where for any two sequences of matrices A_p and B_p, $A_p \asymp B_p$ is used to mean $\mathrm{tr}[C_p(A_p - B_p)] \xrightarrow{\text{a.s.}} 0$ for any deterministic seq of matrices C_p of bnd trace norm

Going beyond . . .

- Equivalences for general functionals of out-of-sample distributions
- Equivalences for ridge variants and general estimators
- Finite sample analysis and rates of convergence

⋮

Discussion and future directions

This work shows GCV and LOOCV uniformly track squared out-of-sample prediction error for ridge regression under proportional asymptotics.

Main tool:

$$(\widehat{\Sigma} + \lambda I_p)^+ \Sigma \asymp \frac{(\widehat{\Sigma} + \lambda I_p)^+ \widehat{\Sigma}}{1 - \mathrm{tr}[(\widehat{\Sigma} + \lambda I_p)^+ \widehat{\Sigma}]/n}$$

where for any two sequences of matrices A_p and B_p, $A_p \asymp B_p$ is used to mean $\mathrm{tr}[C_p(A_p - B_p)] \xrightarrow{\text{a.s.}} 0$ for any deterministic seq of matrices C_p of bnd trace norm

Going beyond . . .

- Equivalences for general functionals of out-of-sample distributions
- Equivalences for ridge variants and general estimators
- Finite sample analysis and rates of convergence

⋮

Discussion and future directions

This work shows GCV and LOOCV uniformly track squared out-of-sample prediction error for ridge regression under proportional asymptotics.

Main tool:

$$(\widehat{\Sigma} + \lambda I_p)^+ \Sigma \asymp \frac{(\widehat{\Sigma} + \lambda I_p)^+ \widehat{\Sigma}}{1 - \mathrm{tr}[(\widehat{\Sigma} + \lambda I_p)^+ \widehat{\Sigma}]/n}$$

where for any two sequences of matrices A_p and B_p, $A_p \asymp B_p$ is used to mean $\mathrm{tr}[C_p(A_p - B_p)] \xrightarrow{\text{a.s.}} 0$ for any deterministic seq of matrices C_p of bnd trace norm

Going beyond . . .

- Equivalences for general functionals of out-of-sample distributions
- Equivalences for ridge variants and general estimators
- Finite sample analysis and rates of convergence

⋮

Discussion and future directions

This work shows GCV and LOOCV uniformly track squared out-of-sample prediction error for ridge regression under proportional asymptotics.

Main tool:

$$(\widehat{\Sigma} + \lambda I_p)^+ \Sigma \asymp \frac{(\widehat{\Sigma} + \lambda I_p)^+ \widehat{\Sigma}}{1 - \mathrm{tr}[(\widehat{\Sigma} + \lambda I_p)^+ \widehat{\Sigma}]/n}$$

where for any two sequences of matrices A_p and B_p, $A_p \asymp B_p$ is used to mean $\mathrm{tr}[C_p(A_p - B_p)] \xrightarrow{\text{a.s.}} 0$ for any deterministic seq of matrices C_p of bnd trace norm

Going beyond . . .

- Equivalences for general functionals of out-of-sample distributions
- Equivalences for ridge variants and general estimators
- Finite sample analysis and rates of convergence

⋮

Discussion and future directions

This work shows GCV and LOOCV uniformly track squared out-of-sample prediction error for ridge regression under proportional asymptotics.

Main tool:

$$(\widehat{\Sigma} + \lambda I_p)^+ \Sigma \asymp \frac{(\widehat{\Sigma} + \lambda I_p)^+ \widehat{\Sigma}}{1 - \mathrm{tr}[(\widehat{\Sigma} + \lambda I_p)^+ \widehat{\Sigma}]/n}$$

where for any two sequences of matrices A_p and B_p, $A_p \asymp B_p$ is used to mean $\mathrm{tr}[C_p(A_p - B_p)] \xrightarrow{\text{a.s.}} 0$ for any deterministic seq of matrices C_p of bnd trace norm

Going beyond . . .

- Equivalences for general functionals of out-of-sample distributions
- Equivalences for ridge variants and general estimators
- Finite sample analysis and rates of convergence

Discussion and future directions

This work shows GCV and LOOCV uniformly track squared out-of-sample prediction error for ridge regression under proportional asymptotics.

Main tool:

$$(\widehat{\Sigma} + \lambda I_p)^+ \Sigma \asymp \frac{(\widehat{\Sigma} + \lambda I_p)^+ \widehat{\Sigma}}{1 - \mathrm{tr}[(\widehat{\Sigma} + \lambda I_p)^+ \widehat{\Sigma}]/n}$$

where for any two sequences of matrices A_p and B_p, $A_p \asymp B_p$ is used to mean $\mathrm{tr}[C_p(A_p - B_p)] \xrightarrow{\text{a.s.}} 0$ for any deterministic seq of matrices C_p of bnd trace norm

Going beyond . . .

- Equivalences for general functionals of out-of-sample distributions
- Equivalences for ridge variants and general estimators
- Finite sample analysis and rates of convergence

⋮

Thanks for listening!

Questions/comments/thoughts?