Linear regression

DS-GA 1013 / MATH-GA 2824 Mathematical Tools for Data Science

https://cims.nyu.edu/~cfgranda/pages/MTDS_spring20/index.html

Carlos Fernandez-Granda

Discussion

Regression

Goal: Estimate a response or dependent variable

Data: Several observed variables, known as covariates, features or independent variables

Probabilistic perspective

Response: random variable $\tilde{y}$

Features: random vector $\tilde{x}$

What estimator minimizes mean square error?

Minimum mean square error

We observe $\tilde{x} = x$

Uncertainty about $\tilde{y}$ is captured by pdf or pmf of $\tilde{y}$ given $\tilde{x} = x$

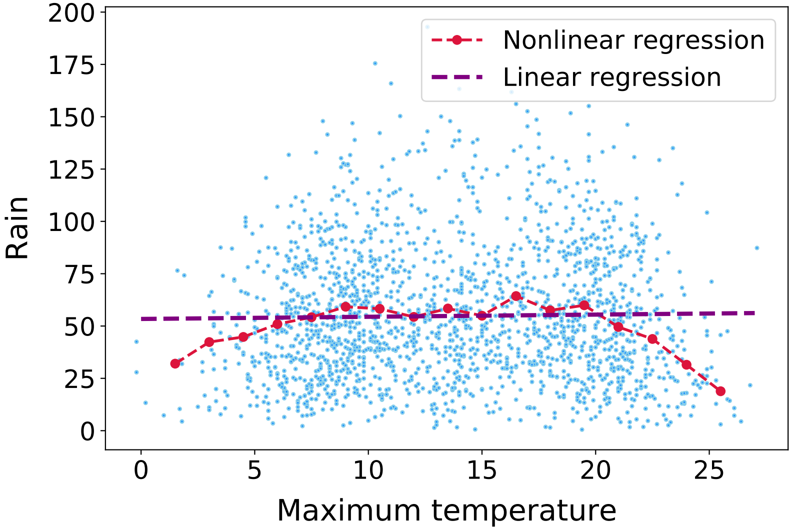
Let y' have that distribution

Minimizing mean square error is equivalent to solving

$$\min_c \mathrm{E}[(\tilde{y}' - c)^2]$$

Minimizer equals conditional mean $\mathrm{E}(\tilde{y} \,|\, \tilde{x} = x)$

Estimating rain from temperature



Are we done?

We need to know the average value of the response for every possible combination of the feature values

For p features with d possible values: d^p

For 5 features with 100 possible values: 10^{10}!

Curse of dimensionality

Linear regression

We need to make assumptions

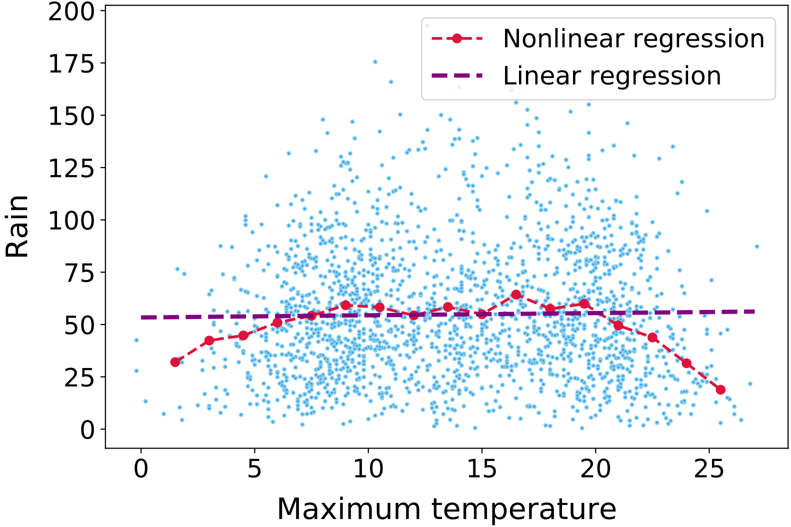
Simple but powerful assumption: Relationship is linear

$$\tilde{y} \approx \beta^T \tilde{x} + \beta_0.$$

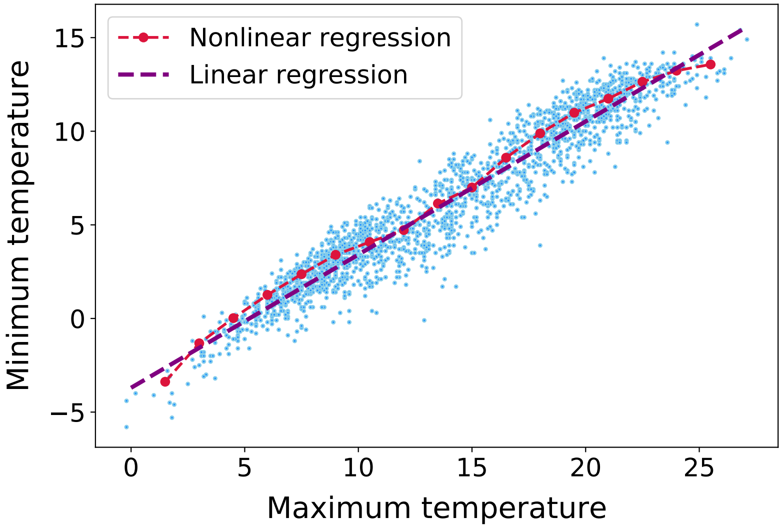
For fixed $\beta \in \mathbb{R}^p$ and $\beta_0 \in \mathbb{R}$

Mathematically, gradient of the regression function is constant

Estimating rain from temperature



Estimating minimum from maximum temperature



Centering

Minimizing mean square error

$$\arg\min_{\beta_0} \mathrm{E}((\tilde{y} - \tilde{x}^T\beta - \beta_0)^2) = \mathrm{E}(\tilde{y} - \tilde{x}^T\beta)$$

For any $\beta \in \mathbb{R}^p$

$$\begin{aligned}\min_{\beta_0} \mathrm{E}\left[(\tilde{y} - \tilde{x}^T\beta - \beta_0)^2\right] &= \mathrm{E}\left[(\tilde{y} - \tilde{x}^T\beta - \mathrm{E}(\tilde{y}) + \mathrm{E}(\tilde{x})^T\beta)^2\right] \\ &= \mathrm{E}\left[(c(\tilde{y}) - \beta^T c(\tilde{x}))^2\right]\end{aligned}$$

From now on, everything will be zero mean

Linear minimum MSE estimator

Goal: Find β minimizing

$$\begin{aligned}\mathrm{E}((\tilde{y} - \tilde{x}^T \beta)^2) &= \mathrm{E}\left(\tilde{y}^2\right) - 2\mathrm{E}\left(\tilde{y}\tilde{x}\right)^T \beta + \beta^T \mathrm{E}(\tilde{x}\tilde{x}^T)\beta \\ &= \beta^T \Sigma_{\tilde{x}} \beta - 2\Sigma_{\tilde{y}\tilde{x}}^T \beta + \mathrm{Var}\left(\tilde{y}\right)\end{aligned}$$

where the cross-covariance vector equals

$$\Sigma_{\tilde{y}\tilde{x}}[i] := \mathrm{E}\left(\tilde{y}\, \tilde{x}[i]\right), \quad 1 \leq i \leq p$$

Linear minimum MSE estimator

Quadratic form

$$f(\beta) := \beta^T \Sigma_{\tilde{x}} \beta - 2\Sigma_{\tilde{y}\tilde{x}}^T \beta + \mathrm{Var}(\tilde{y})$$

$$\nabla f(\beta) = 2\Sigma_{\tilde{x}} \beta - 2\Sigma_{\tilde{y}\tilde{x}}$$

$$\nabla^2 f(\beta) = 2\Sigma_{\tilde{x}}$$

Covariance matrices are positive semidefinite

For any vector $v \in \mathbb{R}^p$

$$v^T \Sigma_{\tilde{x}} v = \mathrm{Var}\left(v^T \tilde{x}\right) \geq 0$$

If $\Sigma_{\tilde{x}}$ is full rank, then positive definite

Quadratic form

For all $\beta_2 \in \mathbb{R}^p$

$$f(\beta_2) = \frac{1}{2}(\beta_2 - \beta_1)^T \nabla^2 f(\beta_1)(\beta_2 - \beta_1) + \nabla f(\beta_1)^T(\beta_2 - \beta_1) + f(\beta_1)$$

If $\nabla f(\beta^*) = 0$ then for any $\beta \neq \beta^*$

$$f(\beta) = \frac{1}{2}(\beta - \beta^*)^T \nabla^2 f(\beta^*)(\beta - \beta^*) + f(\beta^*) > f(\beta^*)$$

if $\nabla^2 f(\beta^*) = \Sigma_{\tilde{x}}$ is positive definite

$$\nabla f(\beta^*) = 2\Sigma_{\tilde{x}}\beta^* - 2\Sigma_{\tilde{y}\tilde{x}} = 0$$

Linear estimator

We need to compute coefficients $\Sigma_{\tilde{x}}^{-1}\Sigma_{\tilde{y}\tilde{x}}$ from data

Training data: (y_1, x_1), (y_2, x_2), ..., (y_n, x_n), where $y_i \in \mathbb{R}$ and $x_i \in \mathbb{R}^p$

We define a response vector $y \in \mathbb{R}^n$ and a feature matrix

$$X := \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}$$

Linear estimator

If features and response are iid samples from $\tilde{x}$ and $\tilde{y}$

$$\Sigma_{\tilde{x}} \approx \frac{1}{n} \sum_{i=1}^{n} x_i x_i^T = \frac{1}{n} X X^T$$

$$\Sigma_{\tilde{y}\tilde{x}} \approx \begin{bmatrix} \frac{1}{n} \sum_{i=1}^{n} x_i[1] y[1] \\ \frac{1}{n} \sum_{i=1}^{n} x_i[2] y[2] \\ \cdots \\ \frac{1}{n} \sum_{i=1}^{n} x_i[p] y[p] \end{bmatrix} = \frac{1}{n} X y$$

$$\Sigma_{\tilde{x}}^{-1} \Sigma_{\tilde{y}\tilde{x}} \approx (X X^T)^{-1} X y$$

Least squares cost function

Reasonable cost function beyond probabilistic assumptions

$$\beta_{\text{OLS}} := \arg\min_{\beta} \sum_{i=1}^{n} \left(y_i - x_i^T \beta \right)^2$$

Known as ordinary least squares (OLS) in statistics

Ordinary least squares

$$\sum_{i=1}^{n} \left(y_i - x_i^T \beta \right)^2 = \|y - X^T \beta\|_2^2$$
$$= \beta^T X X^T \beta - 2 y^T X^T \beta + y^T y$$

Quadratic form with

$$\nabla f(\beta) = 2XX^T \beta - 2Xy$$
$$\nabla^2 f(\beta) = 2XX^T$$

If X is full rank $v^T X X^T v = ||Xv||_2^2 > 0$ for $v \neq 0$

Ordinary least squares

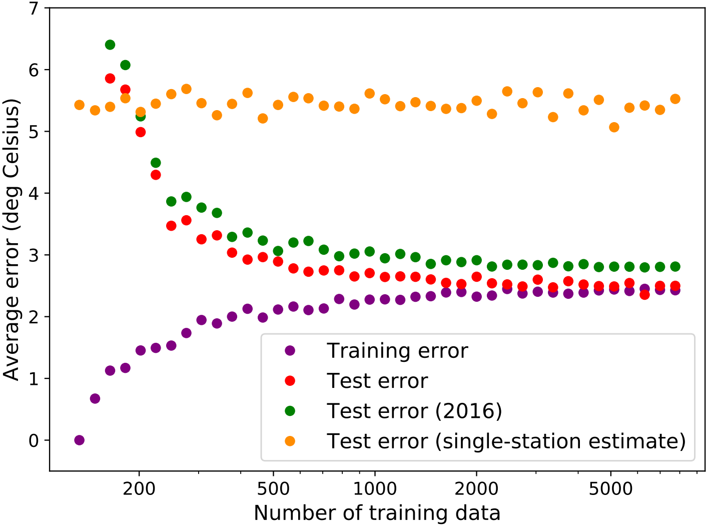
Setting $\nabla f(\beta_{\mathrm{OLS}}) = 0$ yields

$$\beta_{\mathrm{OLS}} = (XX^T)^{-1}Xy$$

Temperature prediction via linear regression

- Dataset of hourly temperatures measured at weather stations all over the US
- Goal: Predict temperature in Yosemite from other temperatures
- Response: Temperature in Yosemite
- Features: Temperatures in 133 other stations ($p = 133$) in 2015
- Test set: 10^3 measurements
- Additional test set: All measurements from 2016

Results



7
6
5
4
3
2
1
0
Average error (deg Celsius)
200
500
1000
2000
5000
Number of training data
Training error
Test error
Test error (2016)
Test error (single-station estimate)

Motivation

Fundamental tool to analyze linear functions

Singular-value decomposition

Every $A \in R^{m \times k}$, $m \geq k$, has a singular-value decomposition (SVD)

$$A = \begin{bmatrix} u_1 & u_2 & \cdots & u_k \end{bmatrix} \begin{bmatrix} s_1 & 0 & \cdots & 0 \\ 0 & s_2 & \cdots & 0 \\ & & \ddots & \\ 0 & 0 & \cdots & s_k \end{bmatrix} \begin{bmatrix} v_1 & v_2 & \cdots & v_k \end{bmatrix}^T$$

$$= USV^T$$

The singular values $s_1 \geq s_2 \geq \cdots \geq s_k$ are nonnegative

The left singular vectors $u_1, u_2, \ldots u_k \in \mathbb{R}^m$ are orthonormal

The right singular vectors $v_1, v_2, \ldots v_k \in \mathbb{R}^k$ are orthonormal

Singular-value decomposition

If $m < k$

$$A = \begin{bmatrix} u_1 & u_2 & \cdots & u_m \end{bmatrix} \begin{bmatrix} s_1 & 0 & \cdots & 0 \\ 0 & s_2 & \cdots & 0 \\ & & \ddots & \\ 0 & 0 & \cdots & s_m \end{bmatrix} \begin{bmatrix} v_1 & v_2 & \cdots & v_m \end{bmatrix}^T$$

$$= USV^T$$

The singular values $s_1 \geq s_2 \geq \cdots \geq s_m$ are nonnegative

The left singular vectors $u_1, u_2, \ldots u_m \in \mathbb{R}^m$ are orthonormal

The right singular vectors $v_1, v_2, \ldots v_m \in \mathbb{R}^k$ are orthonormal

Proof

Assume $m \geq k$ (otherwise apply argument to A^T)

Let $V \Lambda V^T$ be the eigendecomposition of $A^T A$

Eigenvalues are nonnegative because

$$\begin{aligned} ||Av_i||_2^2 &= v_i^T A^T A v_i \\ &= \lambda_i v_i^T v_i \\ &= \lambda_i \end{aligned}$$

Assumption: All eigenvalues are nonzero (general proof in notes)

Proof

For $1 \leq i \leq k$

$$s_i := \sqrt{\lambda_i}$$
$$u_i := \frac{1}{s_i} A v_i$$

$$\begin{aligned} ||u_i||_2^2 &= \frac{1}{s_i^2} v_i^T A^T A v_i \\ &= \frac{\lambda_i}{\lambda_i} v_i^T v_i = 1 \end{aligned}$$

$$\begin{aligned} \langle u_i, u_j \rangle &= \frac{v_i^T A^T A v_j}{s_i s_j} \\ &= \frac{\lambda_j v_i^T v_j}{s_i s_j} = 0 \end{aligned}$$

Proof

$$AV = US$$

$$A = USV^T$$

Great, but what does this mean?

Linear maps

The SVD decomposes the action of a matrix $A \in \mathbb{R}^{m \times k}$ on a vector $w \in \mathbb{R}^k$ into:

1. Rotation

$$V^T w = \sum_{i=1}^{k} \langle v_i, w \rangle e_i$$

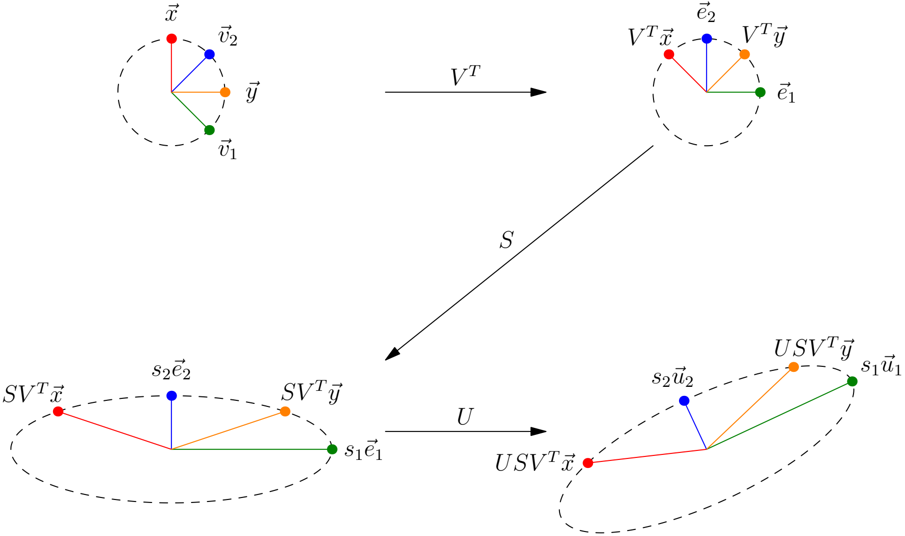
2. Scaling

$$S V^T w = \sum_{i=1}^{k} s_i \langle v_i, w \rangle e_i$$

3. Rotation

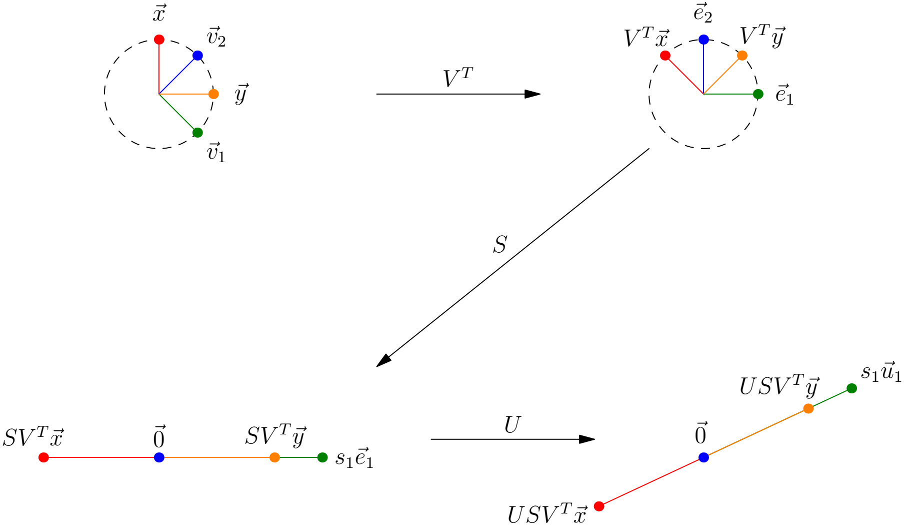
$$U S V^T w = \sum_{i=1}^{k} s_i \langle v_i, w \rangle u_i$$

Linear maps



$\vec{x}$
$\vec{v}_2$
$\vec{y}$
$\vec{v}_1$
V^T
$\vec{e}_2$
$V^T\vec{x}$
$V^T\vec{y}$
$\vec{e}_1$
S
$s_2\vec{e}_2$
$SV^T\vec{x}$
$SV^T\vec{y}$
$s_1\vec{e}_1$
U
$USV^T\vec{y}$
$s_1\vec{u}_1$
$s_2\vec{u}_2$
$USV^T\vec{x}$

Linear maps ($s_2 := 0$)



By the spectral theorem

$$\max_{\{\|w\|_2=1 \mid w\in\mathbb{R}^k\}} \|Aw\|_2^2 = w^T A^T A w$$

$$= s_1^2 \qquad \text{achieved by } v_1$$

By the spectral theorem

$$s_1 = \max_{\{\|w\|_2=1 \mid w \in \mathbb{R}^k\}} \|Aw\|_2$$

$$s_i = \max_{\{\|w\|_2=1 \mid w \in \mathbb{R}^k,\, w \perp v_1, \ldots, v_{i-1}\}} \|Aw\|_2$$

$$v_1 = \arg\max_{\{\|w\|_2=1 \mid w \in \mathbb{R}^k\}} \|Aw\|_2$$

$$v_i = \arg\max_{\{\|w\|_2=1 \mid w \in \mathbb{R}^k,\, w \perp v_1, \ldots, v_{i-1}\}} \|Aw\|_2, \qquad 2 \leq i \leq k$$

OLS estimator

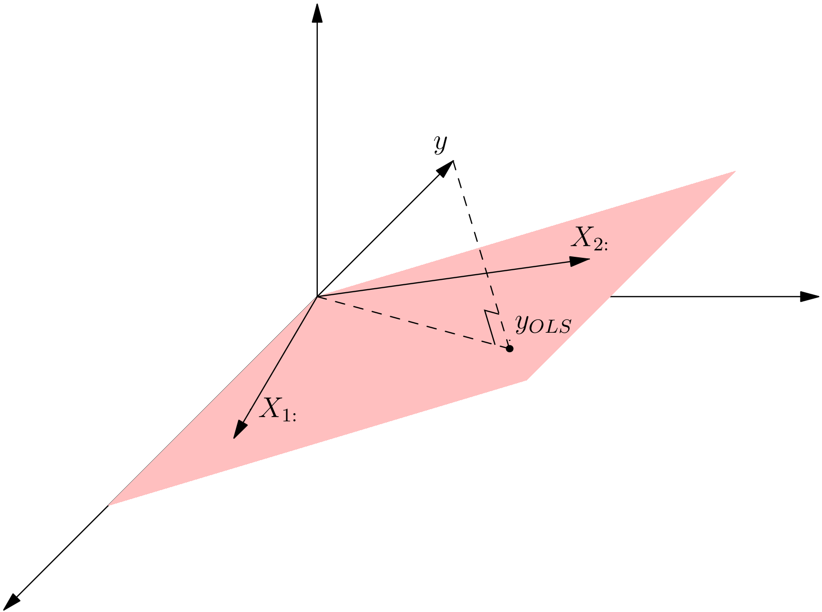
$$
\begin{aligned}
\beta_{\text{OLS}} &= \left(XX^T\right)^{-1} Xy \\
&= (US^2U^T)^{-1}USV^Ty \\
&= US^{-2}U^TUSV^Ty \\
&= US^{-1}V^Ty
\end{aligned}
$$

Geometric interpretation

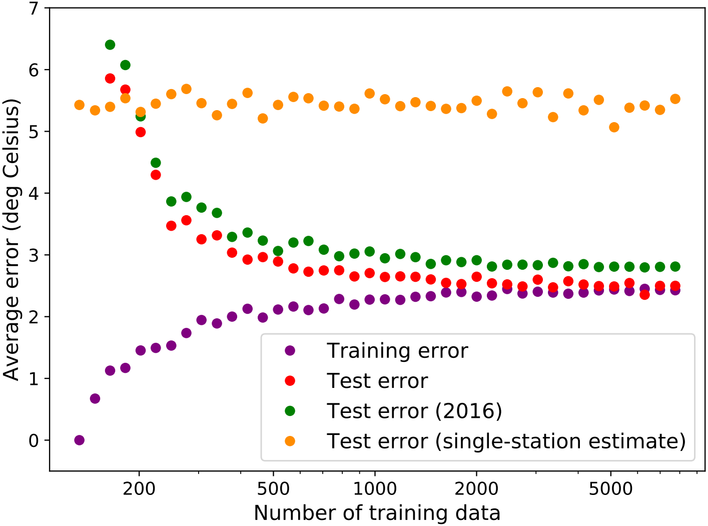
- Any vector $X^T\beta$ is in the span of the rows of X
- The OLS estimate is the closest vector to y that can be represented in this way
- This is the projection of y onto the row space of X

$$
\begin{aligned}
X^T\beta_{\text{OLS}} &= X^T U S^{-1} V^T y \\
&= V S U^T U S^{-1} V^T y \\
&= V V^T y
\end{aligned}
$$

Geometric interpretation



Goal: Understand this



Additive model

Features, noise, and response are random

$$\tilde{y} = \tilde{x}^T \beta_{\text{true}} + \tilde{z}$$

Optimal linear estimator $\Sigma_{\tilde{x}}^{-1} \Sigma_{\tilde{y}\tilde{x}}$

Optimal MSE for additive model

$$
\begin{aligned}
&\mathrm{E}\left[(\tilde{y} - \tilde{x}^T \Sigma_{\tilde{x}}^{-1} \Sigma_{\tilde{y}\tilde{x}})^2\right] \\
&= \mathrm{E}(\tilde{y}^2) + \Sigma_{\tilde{y}\tilde{x}}^T \Sigma_{\tilde{x}}^{-1} \mathrm{E}(\tilde{x}\tilde{x}^T) \Sigma_{\tilde{x}}^{-1} \Sigma_{\tilde{y}\tilde{x}} - 2\mathrm{E}(\tilde{y}\tilde{x}^T) \Sigma_{\tilde{x}}^{-1} \Sigma_{\tilde{y}\tilde{x}} \\
&= \mathrm{Var}(\tilde{y}) - \Sigma_{\tilde{y}\tilde{x}}^T \Sigma_{\tilde{x}}^{-1} \Sigma_{\tilde{y}\tilde{x}} = \mathrm{Var}(\tilde{z})
\end{aligned}
$$

$$
\begin{aligned}
\mathrm{Var}(\tilde{y}) &= \mathrm{Var}(\tilde{x}^T \beta_{\text{true}} + \tilde{z}) \\
&= \beta_{\text{true}}^T \mathrm{E}\left(\tilde{x}\tilde{x}^T\right) \beta_{\text{true}} + \mathrm{Var}\left(\tilde{z}\right) \\
&= \beta_{\text{true}}^T \Sigma_{\tilde{x}} \beta_{\text{true}} + \mathrm{Var}\left(\tilde{z}\right)
\end{aligned}
$$

$$
\begin{aligned}
\Sigma_{\tilde{y}\tilde{x}} &= \mathrm{E}\left(\tilde{x}(\tilde{x}^T \beta_{\text{true}} + \tilde{z})\right) \\
&= \Sigma_{\tilde{x}} \beta_{\text{true}}
\end{aligned}
$$

Optimal MSE for additive model

Can we do better than $\mathrm{Var}(\tilde{z})$?

Are we done here?

Training data

$$\tilde{y}_{\text{train}} := X^T \beta_{\text{true}} + \tilde{z}_{\text{train}}$$

- Feature matrix $X \in \mathbb{R}^{p \times n}$ is deterministic
- Coefficients $\beta_{\text{true}} \in \mathbb{R}^p$ are deterministic
- Noise $\tilde{z}_{\text{train}}$ is an n-dimensional iid Gaussian vector with zero mean and variance σ^2

Maximum likelihood

Under this model, OLS is equivalent to maximum likelihood

Assume we observe y_{train}

$$\mathcal{L}_{y_{\text{train}}}(\beta) = \frac{1}{\sqrt{(2\pi\sigma^2)^n}} \exp\left(-\frac{1}{2\sigma^2}\left|\left|y_{\text{train}} - X^T\beta\right|\right|_2^2\right)$$

$$\begin{aligned}
\beta_{\text{ML}} &= \arg\max_{\beta} \mathcal{L}_{y_{\text{train}}}(\beta) \\
&= \arg\max_{\beta} \log \mathcal{L}_{y_{\text{train}}}(\beta) \\
&= \arg\min_{\beta} \left|\left|y_{\text{train}} - X^T\beta\right|\right|_2^2
\end{aligned}$$

Decomposition of OLS cost function

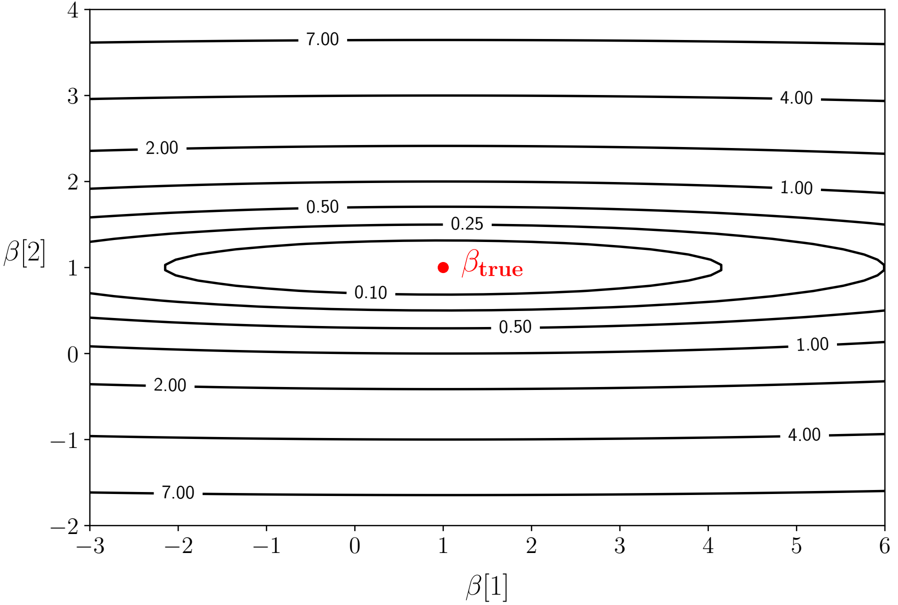
$$\arg\min_{\beta} \|\tilde{y}_{\text{train}} - X^T\beta\|_2^2$$

$$= \arg\min_{\beta} \|\tilde{z}_{\text{train}} - X^T(\beta - \beta_{\text{true}})\|_2^2$$

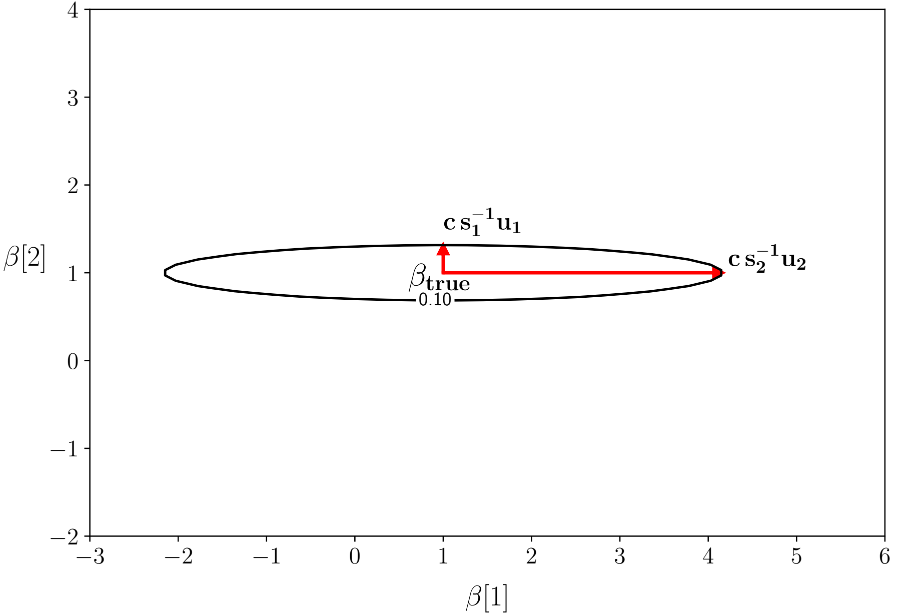
$$= \arg\min_{\beta} (\beta - \beta_{\text{true}})^T X X^T (\beta - \beta_{\text{true}}) - 2\tilde{z}_{\text{train}}^T X^T (\beta - \beta_{\text{true}}) + \tilde{z}_{\text{train}}^T \tilde{z}_{\text{train}}$$

$$= \arg\min_{\beta} (\beta - \beta_{\text{true}})^T X X^T (\beta - \beta_{\text{true}}) - 2\tilde{z}_{\text{train}}^T X^T \beta$$

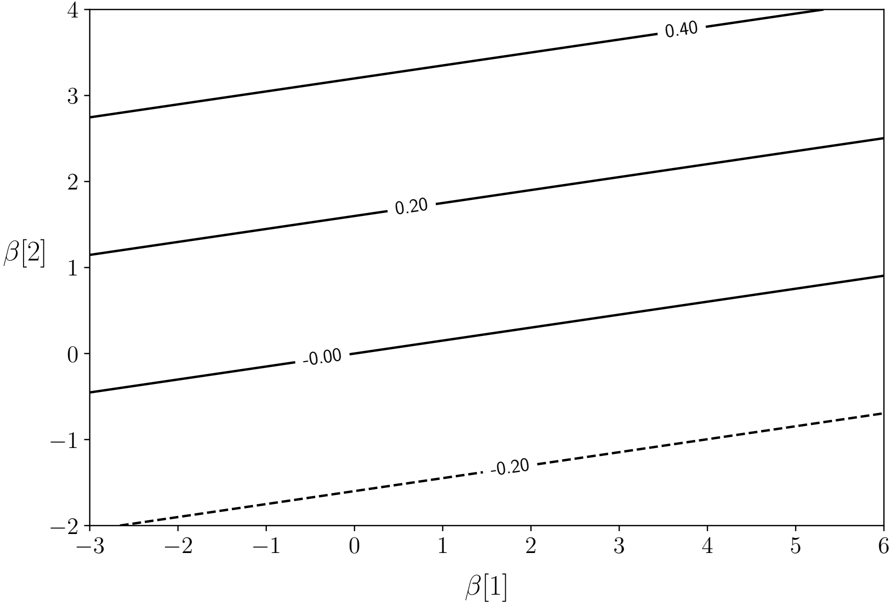
$$(\beta - \beta_{\text{true}})^T X X^T (\beta - \beta_{\text{true}})$$

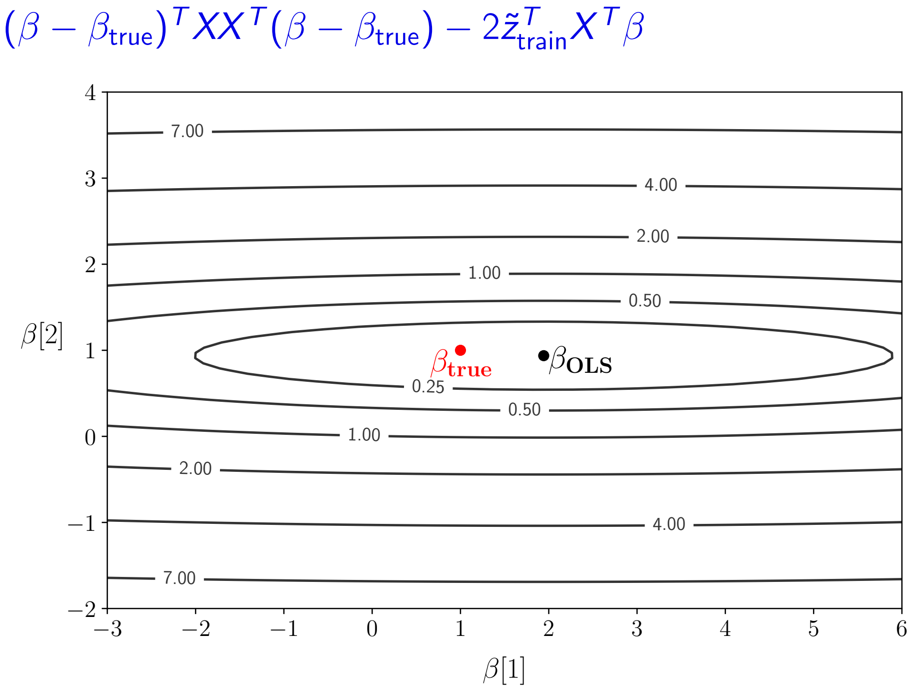


$$(\beta - \beta_{\text{true}})^T X X^T (\beta - \beta_{\text{true}})$$



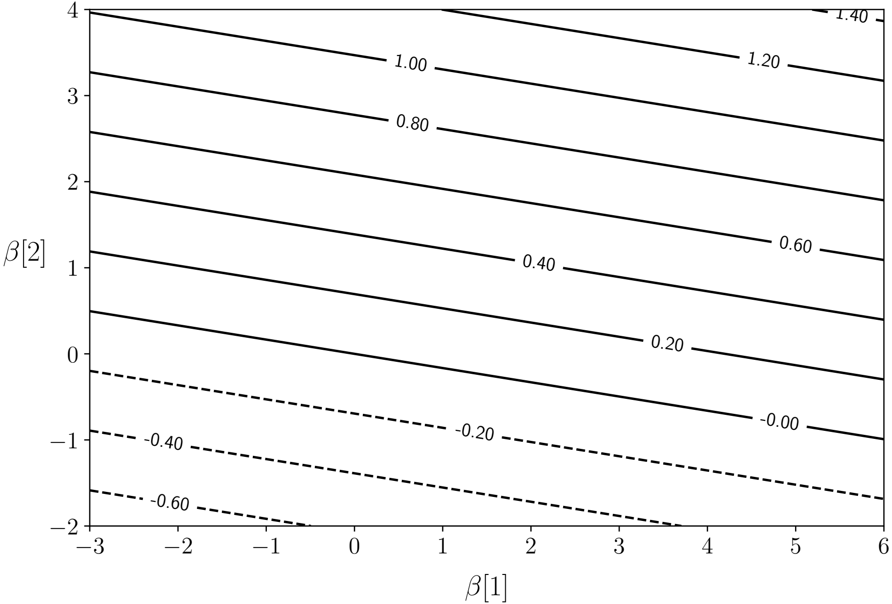
$-2\tilde{z}_{\text{train}}^T X^T \beta$



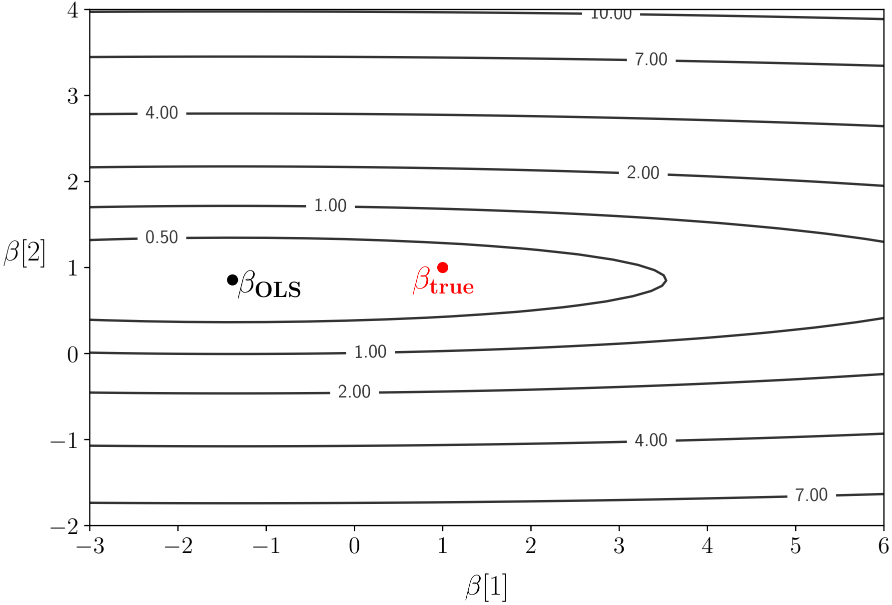


$(\beta - \beta_{\text{true}})^T X X^T (\beta - \beta_{\text{true}}) - 2\tilde{z}_{\text{train}}^T X^T \beta$
7.00
4.00
2.00
1.00
0.50
0.25
0.50
1.00
2.00
4.00
7.00
$\beta_{\textbf{true}}$
$\beta_{\textbf{OLS}}$
$\beta[2]$
$\beta[1]$
4
3
2
1
0
−1
−2
−3
−2
−1
0
1
2
3
4
5
6

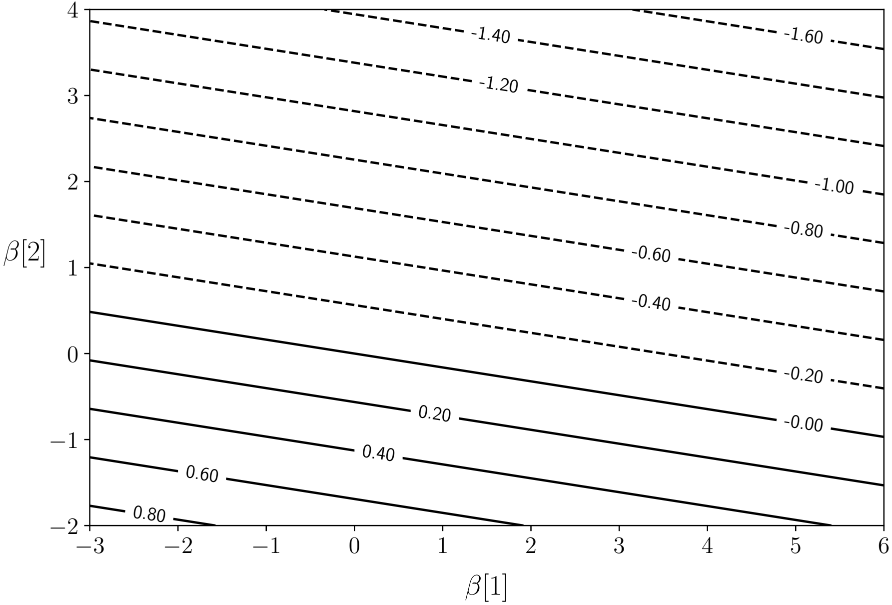
$-2\tilde{z}_{\text{train}}^T X^T \beta$



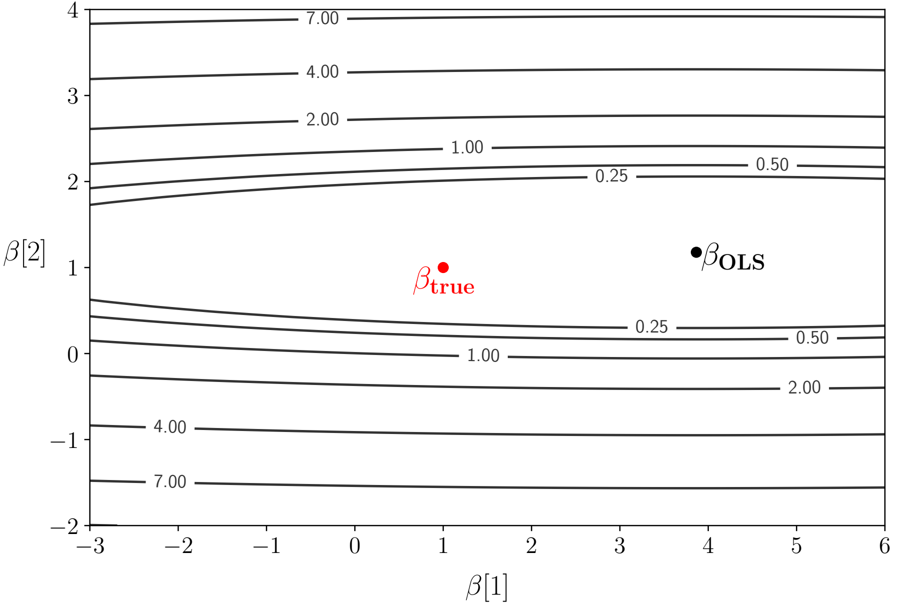
$$(\beta - \beta_{\text{true}})^T X X^T (\beta - \beta_{\text{true}}) - 2\tilde{z}_{\text{train}}^T X^T \beta$$



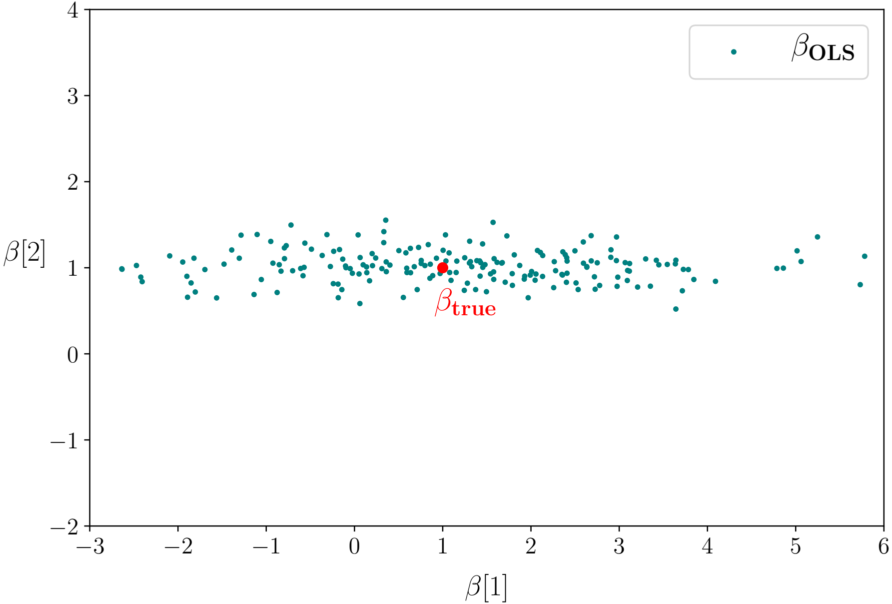
$-2\tilde{z}_{\text{train}}^T X^T \beta$



$$(\beta - \beta_{\text{true}})^T X X^T (\beta - \beta_{\text{true}}) - 2\tilde{z}_{\text{train}}^T X^T \beta$$



Minima for 200 realizations

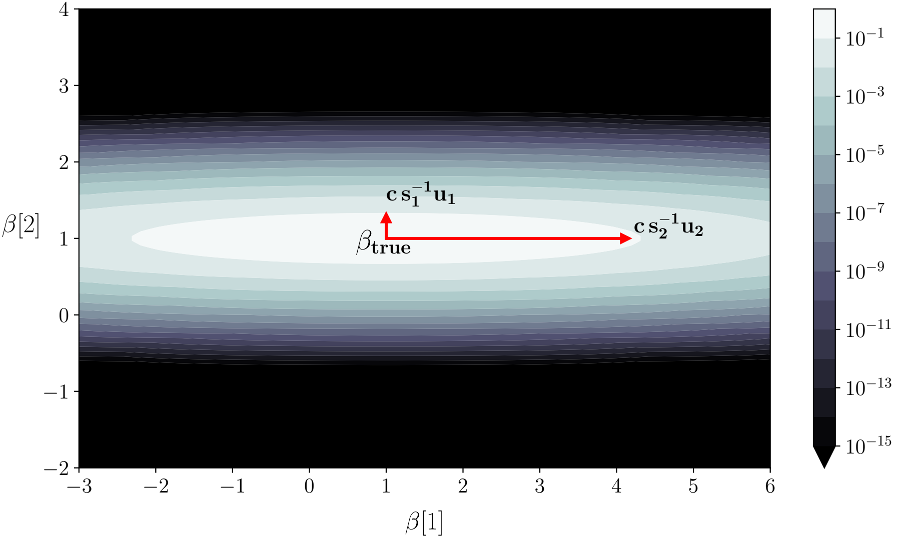


Minima

$$
\begin{aligned}
\beta_{\text{OLS}} &= (XX^T)^{-1} X \tilde{y}_{\text{train}} \\
&= (XX^T)^{-1} XX^T \beta_{\text{true}} + (XX^T)^{-1} X \tilde{z}_{\text{train}} \\
&= \beta_{\text{true}} + (XX^T)^{-1} X \tilde{z}_{\text{train}} \\
&= \beta_{\text{true}} + US^{-1} V^T \tilde{z}_{\text{train}}
\end{aligned}
$$

Distribution? Gaussian with mean β_{true} and covariance matrix $\sigma^2 US^{-2} U^T$

Minima



4
3
2
1
0
−1
−2
−3
−2
−1
0
1
2
3
4
5
6
β[2]
β[1]
c s₁⁻¹u₁
c s₂⁻¹u₂
β_true
10^−1
10^−3
10^−5
10^−7
10^−9
10^−11
10^−13
10^−15

Training error

$$\begin{aligned}
\tilde{y}_{\text{train}} - X^T \tilde{\beta}_{\text{OLS}} &= \tilde{y}_{\text{train}} - \mathcal{P}_{\text{row}(X)}\, \tilde{y}_{\text{train}} \\
&= X^T \beta_{\text{true}} + \tilde{z}_{\text{train}} - \mathcal{P}_{\text{row}(X)} \left(X^T \beta_{\text{true}} + \tilde{z}_{\text{train}} \right) \\
&= X^T \beta_{\text{true}} + \tilde{z}_{\text{train}} - X^T \beta_{\text{true}} - \mathcal{P}_{\text{row}(X)}\, \tilde{z}_{\text{train}} \\
&= \mathcal{P}_{\text{row}(X)^{\perp}}\, \tilde{z}_{\text{train}}
\end{aligned}$$

Goal: Characterize average training square error

$$\widetilde{E}^2_{\text{train}} := \frac{1}{n} \left|\left| \tilde{y}_{\text{train}} - X^T \tilde{\beta}_{\text{OLS}} \right|\right|_2^2$$
$$= \frac{1}{n} \left|\left| \mathcal{P}_{\text{row}(X)^\perp} \, \tilde{z}_{\text{train}} \right|\right|_2^2$$

Requires studying the projection of an iid Gaussian vector on a subspace

In $\mathbb{R}^n$ what fraction of the variance captured by subspace of dimension p?

Average training square error

$$\left|\left|\mathcal{P}_{\text{row}(X)^{\perp}}\,\tilde{z}_{\text{train}}\right|\right|_2^2 = \tilde{z}_{\text{train}}^T V_\perp V_\perp^T V_\perp V_\perp^T \tilde{z}_{\text{train}}$$
$$= \left|\left|V_\perp^T \tilde{z}_{\text{train}}\right|\right|_2^2$$

$V_\perp^T \tilde{z}_{\text{train}}$ is an $n - p$ dimensional Gaussian vector with covariance matrix

$$\Sigma_{V_\perp^T \tilde{z}_{\text{train}}} = V_\perp^T \Sigma_{\tilde{z}_{\text{train}}} V_\perp$$
$$= V_\perp^T \sigma^2 I V_\perp$$
$$= \sigma^2 I$$

It's an iid Gaussian vector!

ℓ_2 norm of d-dimensional iid standard Gaussian vector

$$\begin{aligned}
\mathrm{E}\left(||\tilde{w}||_2^2\right) &= \mathrm{E}\left(\sum_{i=1}^{d} \tilde{w}[i]^2\right) \\
&= \sum_{i=1}^{d} \mathrm{E}\left(\tilde{w}[i]^2\right) \\
&= d
\end{aligned}$$

ℓ_2 norm of d-dimensional iid standard Gaussian vector

$$\mathrm{E}\left[\left(||\tilde{w}||_2^2\right)^2\right] = \mathrm{E}\left[\left(\sum_{i=1}^{d} \tilde{w}[i]^2\right)^2\right]$$

$$= \sum_{i=1}^{d} \sum_{j=1}^{d} \mathrm{E}\left(\tilde{w}[i]^2 \tilde{w}[j]^2\right)$$

$$= \sum_{i=1}^{d} \mathrm{E}\left(\tilde{w}[i]^4\right) + 2 \sum_{i=1}^{d-1} \sum_{j=i+1}^{d} \mathrm{E}\left(\tilde{w}[i]^2\right) \mathrm{E}\left(\tilde{w}[j]^2\right)$$

$$= 3d + d(d-1) \quad \text{(4th moment of standard Gaussian = 3)}$$

$$= d(d+2)$$

$$\mathrm{Var}\left(||\tilde{w}||_2^2\right) = \mathrm{E}\left[\left(||\tilde{w}||_2^2\right)^2\right] - \mathrm{E}^2\left(||\tilde{w}||_2^2\right)$$

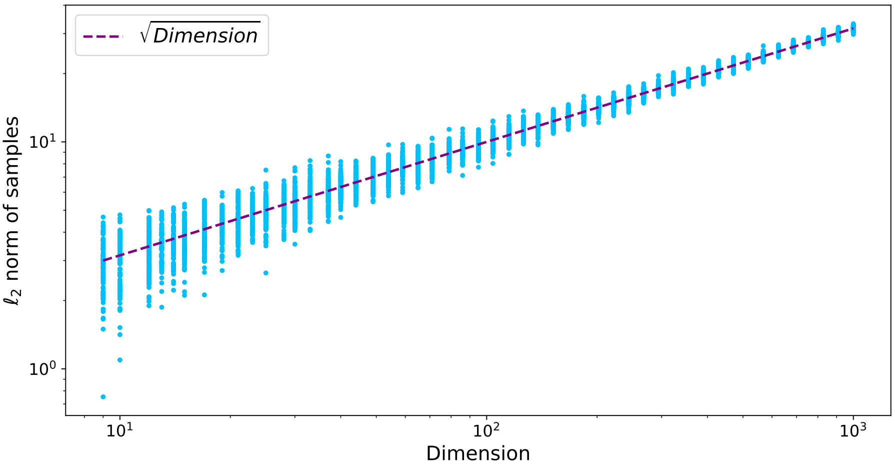
$$= 2d$$

ℓ_2 norm of d-dimensional iid standard Gaussian vector

As d grows, the std scales as $1/\sqrt{d}$ with respect to the mean

Geometrically, how do Gaussians look in high dimensions?

ℓ_2 norm of d-dimensional iid standard Gaussian vector



Average training square error

$$\widetilde{E}^2_{\text{train}} = \frac{1}{n} \left|\left| V_\perp^T \tilde{z}_{\text{train}} \right|\right|_2^2$$
$$= \frac{\sigma^2}{n} \left|\left|\tilde{w}\right|\right|_2^2$$

Dimension? $n - p$

$$\mathrm{E}\left(\widetilde{E}^2_{\text{train}}\right) = \sigma^2 \left(1 - \frac{p}{n}\right)$$

$$\mathrm{Var}(\widetilde{E}^2_{\text{train}}) = \frac{2\sigma^4 (n-p)}{n^2}$$

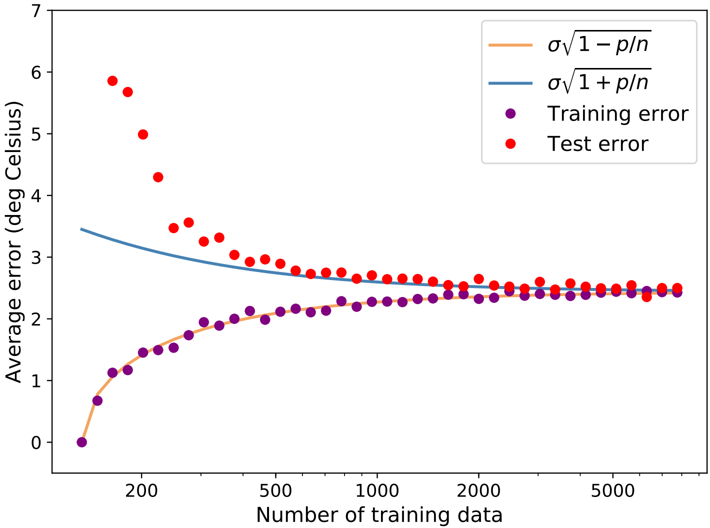
Average training square error

$$\text{Training error} \approx \sigma\sqrt{1-\frac{p}{n}}$$

When $p << n$, error = noise

When $p \approx n$, error is very small: good news?

Observed training square error



Test data

Training data

$$\tilde{y}_{\text{train}} := X^T \beta_{\text{true}} + \tilde{z}_{\text{train}}$$

Test data

$$\tilde{y}_{\text{test}} := \tilde{x}_{\text{test}}^T \beta_{\text{true}} + \tilde{z}_{\text{test}}$$

$\tilde{x}_{\text{test}}$ is zero mean

$\tilde{z}_{\text{test}}$ is zero-mean Gaussian with variance σ^2

Test error

Goal: Characterize mean square of

$$\begin{aligned}\widetilde{E}_{\text{test}} &:= \tilde{y}_{\text{test}} - \tilde{x}_{\text{test}}^T \tilde{\beta}_{\text{OLS}} \\ &= \tilde{z}_{\text{test}} + \tilde{x}_{\text{test}}^T \left(\beta_{\text{true}} - \tilde{\beta}_{\text{OLS}}\right)\end{aligned}$$

where $\tilde{\beta}_{\text{OLS}}$ is computed from the training data

By independence

$$\operatorname{Var}\left(\tilde{y}_{\text{test}} - \tilde{x}_{\text{test}}^T \tilde{\beta}_{\text{OLS}}\right) = \sigma^2 + \operatorname{Var}\left(\tilde{x}_{\text{test}}^T \left(\beta_{\text{true}} - \tilde{\beta}_{\text{OLS}}\right)\right)$$

Everything is zero mean so mean square = variance

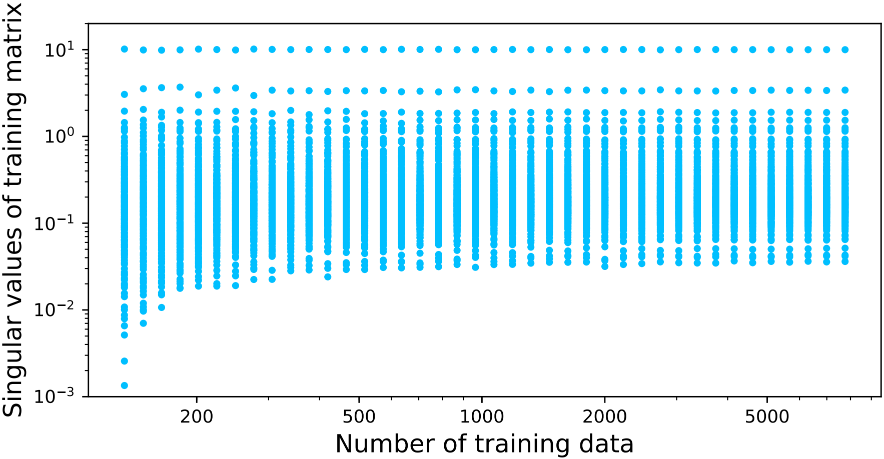
Coefficient error

Let USV^T be the SVD of X

$$\beta_{\text{OLS}} - \beta_{\text{true}} = US^{-1}V^T\tilde{z}_{\text{train}}$$
$$= \sum_{i=1}^{p} \frac{v_i^T\tilde{z}_{\text{train}}}{s_i} u_i$$

Potentially worrying: singular values can be very small

Singular values for temperature dataset



Mean square test error

$$\mathrm{E}\left[\left(\tilde{x}_{\text{test}}^T\left(\beta_{\text{true}}-\tilde{\beta}_{\text{OLS}}\right)\right)^2\right]=\mathrm{E}\left[\left(\sum_{i=1}^{p}\frac{v_i^T\tilde{z}_{\text{train}}\; u_i^T\tilde{x}_{\text{test}}}{s_i}\right)^2\right]$$

$$=\sum_{i=1}^{p}\frac{\mathrm{E}\left[(v_i^T\tilde{z}_{\text{train}})^2\right]\mathrm{E}\left[(u_i^T\tilde{x}_{\text{test}})^2\right]}{s_i^2}$$

$$\mathrm{E}\left(\frac{v_i^T\tilde{z}_{\text{train}}\; u_i^T\tilde{x}_{\text{test}}}{s_i}\frac{v_j^T\tilde{z}_{\text{train}}\; u_j^T\tilde{x}_{\text{test}}}{s_j}\right)=\frac{\mathrm{E}\left(u_i^T\tilde{x}_{\text{test}}u_j^T\tilde{x}_{\text{test}}\right)}{s_i s_j}v_i^T\mathrm{E}\left(\tilde{z}_{\text{train}}\tilde{z}_{\text{train}}^T\right)v_j$$

$$=\frac{\mathrm{E}\left(u_i^T\tilde{x}_{\text{test}}u_j^T\tilde{x}_{\text{test}}\right)}{s_i s_j}v_i^T v_j$$

$$=0 \qquad \text{for } i\neq j$$

Mean square test error

$$\mathrm{E}\left[\left(\tilde{x}_{\text{test}}^T\left(\beta_{\text{true}} - \tilde{\beta}_{\text{OLS}}\right)\right)^2\right] = \sum_{i=1}^{p} \frac{\mathrm{E}\left[(v_i^T \tilde{z}_{\text{train}})^2\right]\mathrm{E}\left[(u_i^T \tilde{x}_{\text{test}})^2\right]}{s_i^2}$$

$$= \sum_{i=1}^{p} \frac{v_i^T \mathrm{E}(\tilde{z}_{\text{train}} \tilde{z}_{\text{train}}^T) v_i u_i^T \mathrm{E}(\tilde{x}_{\text{test}} \tilde{x}_{\text{test}}^T) u_i}{s_i^2}$$

$$= \sigma^2 \sum_{i=1}^{p} \frac{u_i^T \Sigma_{\tilde{x}_{\text{test}}} u_i}{s_i^2}$$

$$\mathrm{E}(\widetilde{E}_{\text{test}}^2) = \sigma^2 + \sigma^2 \sum_{i=1}^{p} \frac{\mathrm{Var}(u_i^T \tilde{x}_{\text{test}})}{s_i^2}$$

Are small singular values problematic?

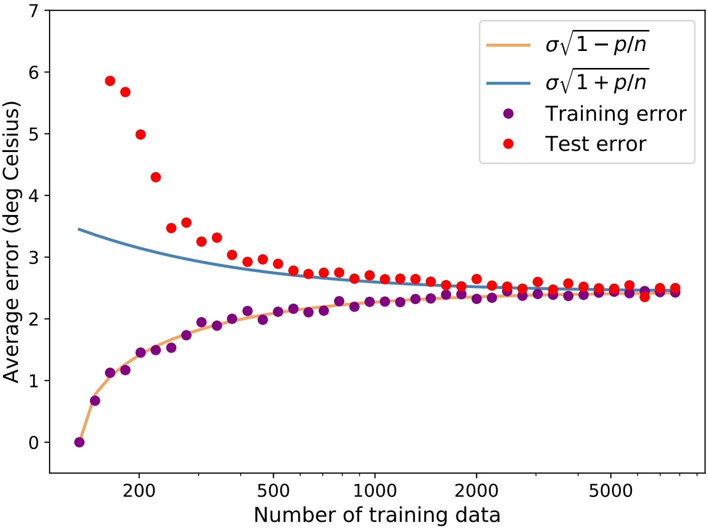
Mean square test error

$$
\begin{aligned}
\frac{s_i^2}{n} &= \frac{u_i X X^T u_i}{n} \\
&= u_i^T \Sigma_{\mathcal{X}} u_i \\
&= \mathrm{var}\left(\mathcal{P}_{u_i}\, \mathcal{X}\right)
\end{aligned}
$$

$$
\begin{aligned}
\mathrm{E}(\widetilde{E}^2_{\text{test}}) &= \sigma^2 + \sigma^2 \sum_{i=1}^{p} \frac{\mathrm{Var}(u_i^T \tilde{x}_{\text{test}})}{s_i^2} \\
&\approx \sigma^2 \left(1 + \frac{p}{n}\right)
\end{aligned}
$$

if sample variance $\approx$ test variance, no!

Observed test square error



Mean square error and least squares

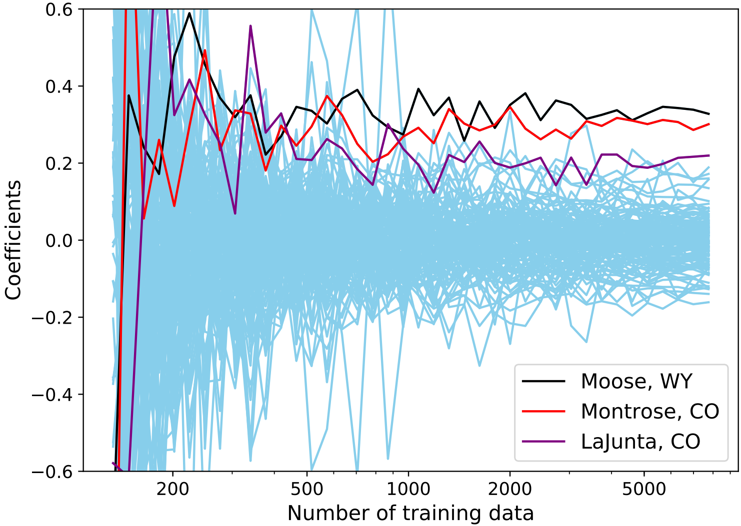
The singular-value decomposition

Error analysis

Ridge regression

Gradient descent

Temperature prediction via linear regression



Motivation

Overfitting often reflected in large coefficients that cancel out to match the noise

Possible solution: Penalize large-norm solutions when fitting the model

Adding a penalty term to promote a particular structure is called regularization

Ridge regression

For a fixed regularization parameter $\lambda > 0$

$$\beta_{\mathsf{RR}} := \arg\min_{\beta} \|y - X^T\beta\|_2^2 + \lambda\|\beta\|_2^2$$

When $\lambda \to 0$ then $\beta_{\mathsf{RR}} \to \beta_{\mathsf{LS}}$

When $\lambda \to \infty$ then $\beta_{\mathsf{RR}} \to 0$

Ridge regression

β_{RR} is the solution to a modified least-squares problem

$$\begin{aligned}\beta_{\text{RR}} &= \arg\min_{\beta} \left|\left| \begin{bmatrix} y \\ 0 \end{bmatrix} - \begin{bmatrix} X^T \\ \sqrt{\lambda} I \end{bmatrix} \beta \right|\right|_2^2 \\ &= \left(\begin{bmatrix} X & \sqrt{\lambda} I \end{bmatrix} \begin{bmatrix} X & \sqrt{\lambda} I \end{bmatrix}^T \right)^{-1} \begin{bmatrix} X & \sqrt{\lambda} I \end{bmatrix} \begin{bmatrix} y \\ 0 \end{bmatrix} \\ &= \left(X X^T + \lambda I \right)^{-1} X y\end{aligned}$$

Problem

How to calibrate regularization parameter

Should we choose that λ that yields the best fit?

Better option: Check fit on validation data

Cross validation

Given a set of examples

$$\left(y^{(1)}, x^{(1)}\right), \left(y^{(2)}, x^{(2)}\right), \ldots, \left(y^{(n)}, x^{(n)}\right),$$

1. Partition data into a training set $X_{\text{train}} \in \mathbb{R}^{n_{\text{train}} \times p}$, $y_{\text{train}} \in \mathbb{R}^{n_{\text{train}}}$ and a validation set $X_{\text{val}} \in \mathbb{R}^{n_{\text{val}} \times p}$, $y_{\text{val}} \in \mathbb{R}^{n_{\text{val}}}$

2. Fit model using the training set for every λ in a set Λ

$$\beta_{\text{RR}}(\lambda) := \arg\min_{\beta} \left|\left|y_{\text{train}} - X_{\text{train}}\beta\right|\right|_2^2 + \lambda \left|\left|\beta\right|\right|_2^2$$

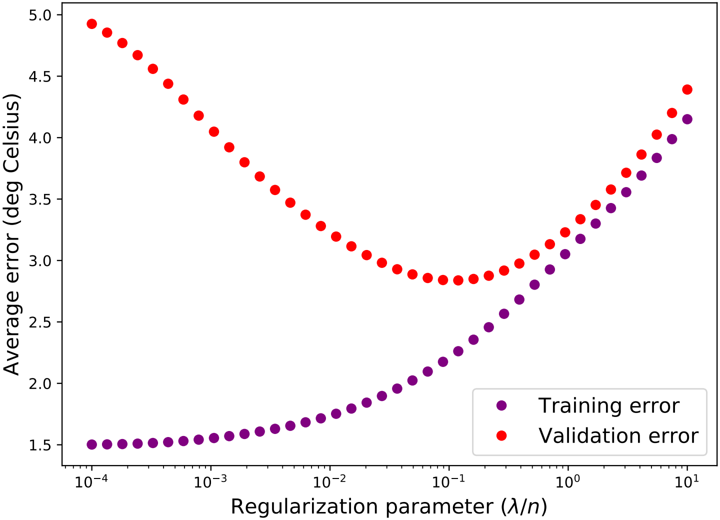
   and evaluate the fitting error on the validation set

$$\text{err}(\lambda) := \left|\left|y_{\text{val}} - X_{\text{val}}\beta_{\text{RR}}(\lambda)\right|\right|_2^2$$

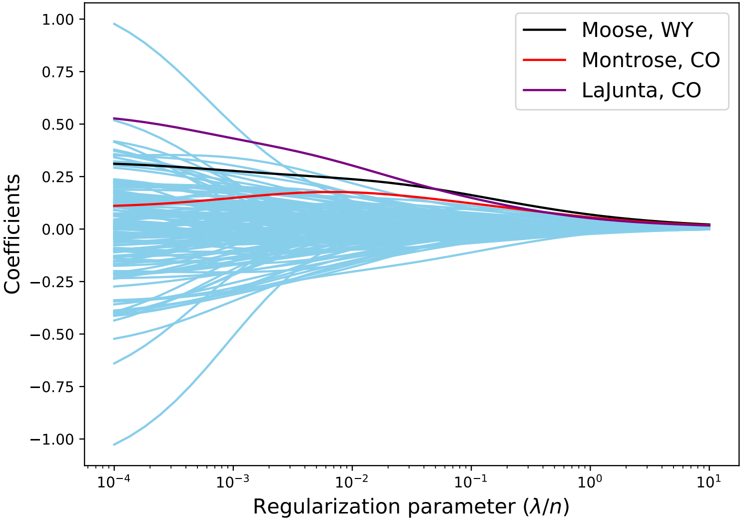
3. Choose the value of λ that minimizes the validation-set error

$$\lambda_{\text{cv}} := \arg\min_{\lambda \in \Lambda} \text{err}(\lambda)$$

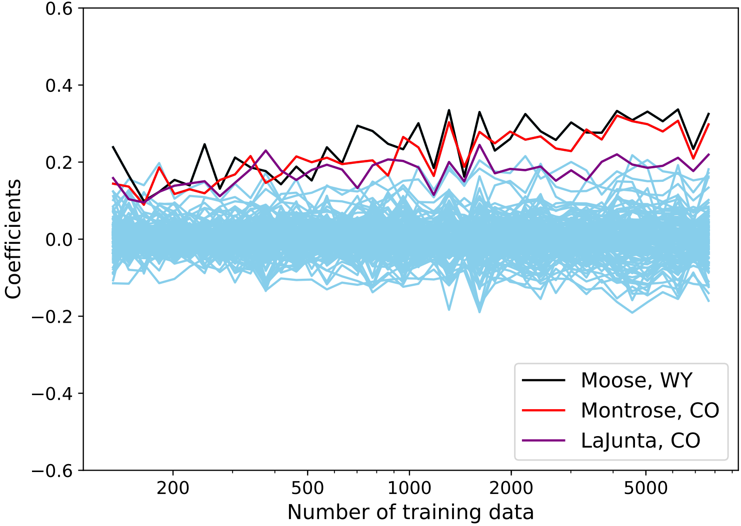
Temperature prediction via ridge regression ($n = 202$)



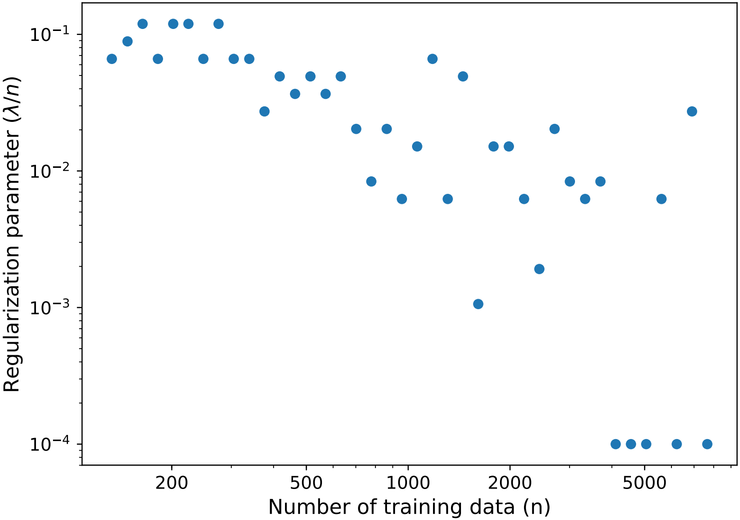
Temperature prediction via ridge regression ($n = 202$)



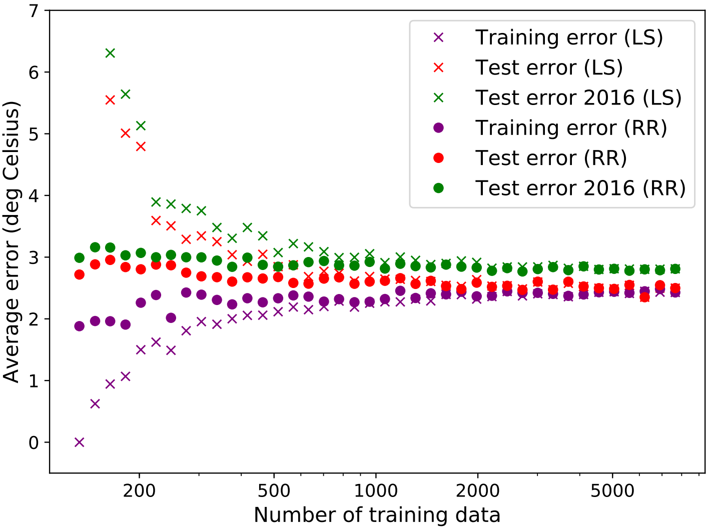
Temperature prediction via ridge regression



Temperature prediction via ridge regression



Temperature prediction via ridge regression



Additive model

$$\tilde{y}_{\text{train}} := X^T \beta_{\text{true}} + \tilde{z}_{\text{train}}$$

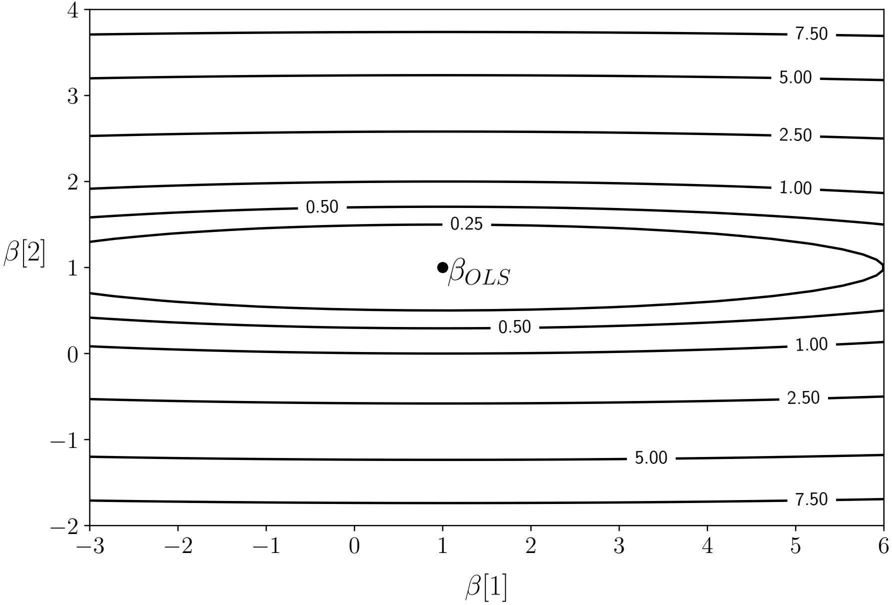
Goal: Understand how ridge regression works

Decomposition of ridge-regression cost function

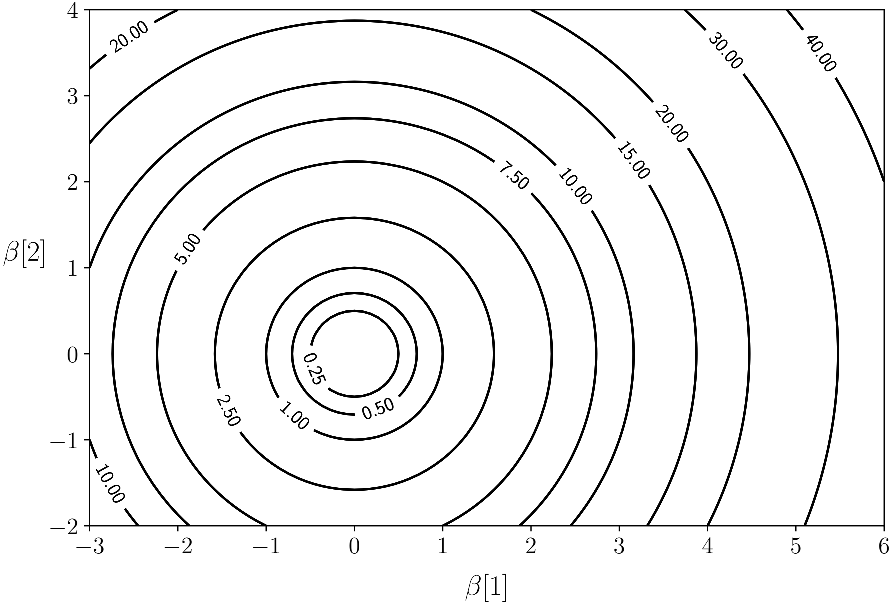
$$\arg\min_{\beta} \|\tilde{y}_{\text{train}} - X^T\beta\|_2^2 + \lambda\|\beta\|_2^2$$

$$= \arg\min_{\beta} (\beta - \beta_{\text{true}})^T X X^T (\beta - \beta_{\text{true}}) + \lambda\beta^T\beta - 2\tilde{z}_{\text{train}}^T X^T \beta$$

$(\beta - \beta_{\text{true}})^T X X^T (\beta - \beta_{\text{true}})$

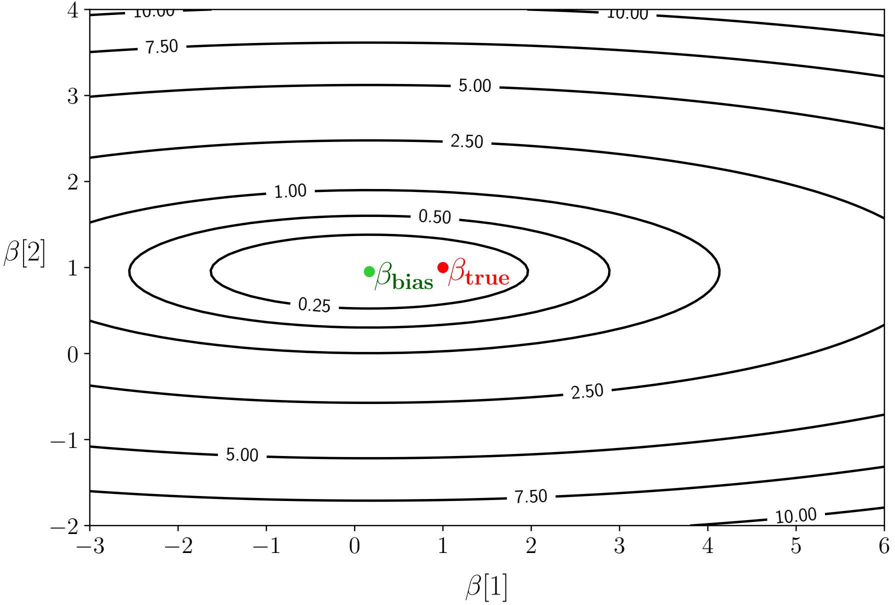


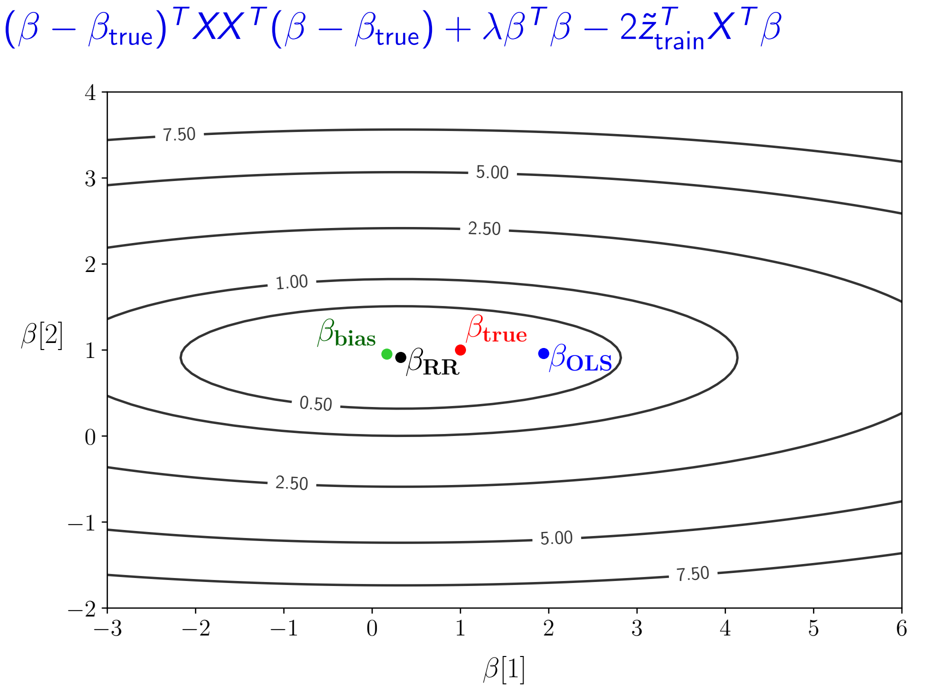
$\beta^T\beta$

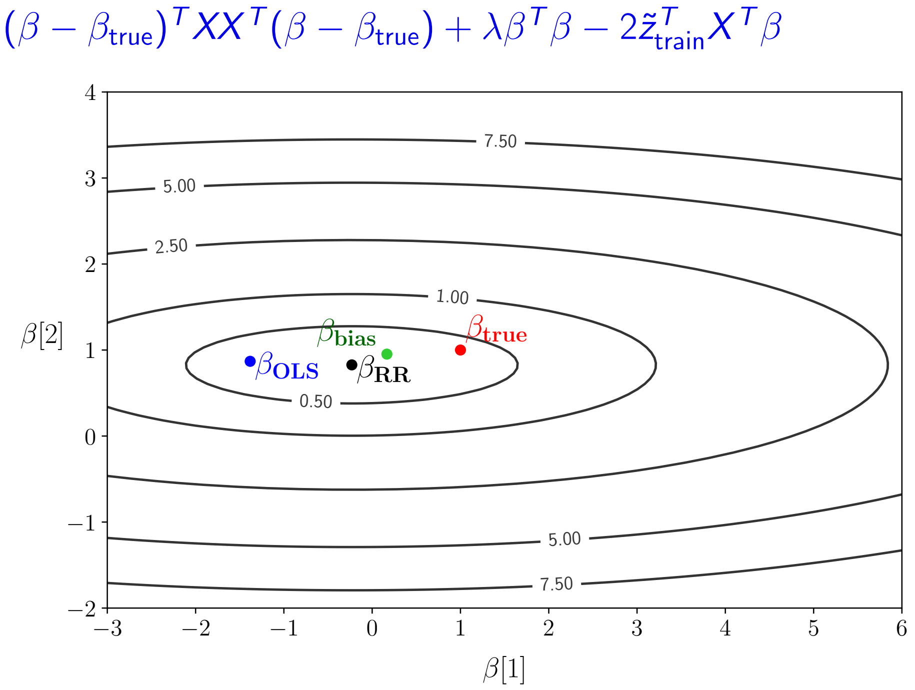


20.00
30.00
40.00
20.00
15.00
10.00
7.50
5.00
2.50
1.00
0.50
0.25
10.00
β[2]
β[1]
4
3
2
1
0
−1
−2
−3
−2
−1
0
1
2
3
4
5
6

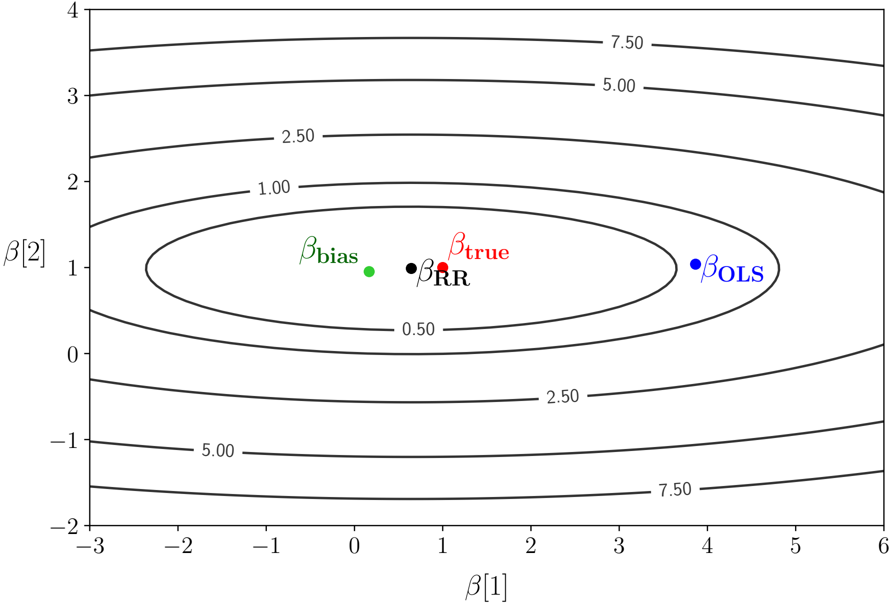
$$(\beta - \beta_{\text{true}})^T X X^T (\beta - \beta_{\text{true}}) + \lambda \beta^T \beta$$



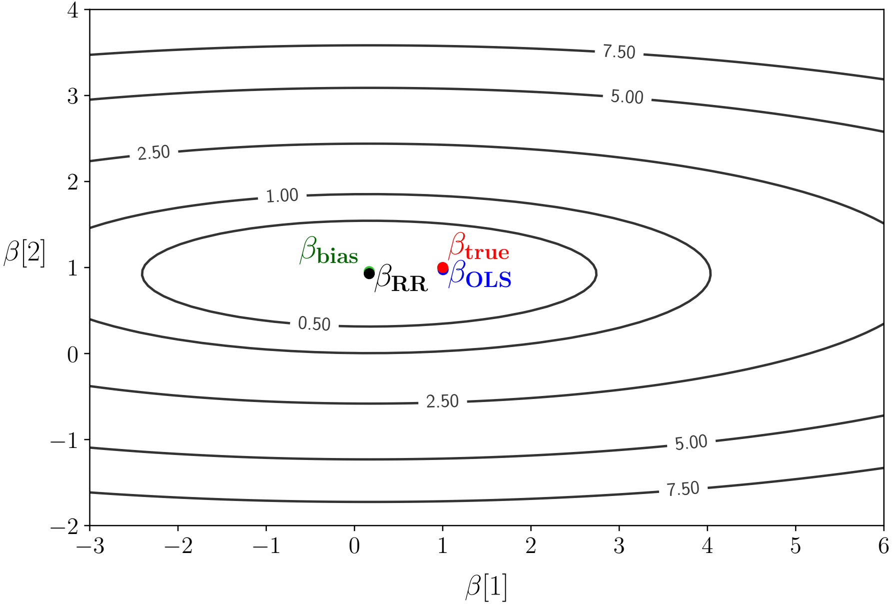

$(\beta - \beta_{\text{true}})^T X X^T (\beta - \beta_{\text{true}}) + \lambda \beta^T \beta - 2\tilde{z}_{\text{train}}^T X^T \beta$
7.50
5.00
2.50
1.00
0.50
$\beta_{\mathbf{bias}}$
$\beta_{\mathbf{RR}}$
$\beta_{\mathbf{true}}$
$\beta_{\mathbf{OLS}}$
$\beta[2]$
$\beta[1]$


$(\beta - \beta_{\text{true}})^T X X^T (\beta - \beta_{\text{true}}) + \lambda \beta^T \beta - 2\tilde{z}_{\text{train}}^T X^T \beta$
7.50
5.00
2.50
1.00
0.50
5.00
7.50
$\beta_{\mathbf{bias}}$
$\beta_{\mathbf{true}}$
$\beta_{\mathbf{OLS}}$
$\beta_{\mathbf{RR}}$
$\beta[2]$
$\beta[1]$
4
3
2
1
0
−1
−2
−3
−2
−1
0
1
2
3
4
5
6

$$(\beta - \beta_{\text{true}})^T X X^T (\beta - \beta_{\text{true}}) + \lambda \beta^T \beta - 2\tilde{z}_{\text{train}}^T X^T \beta$$



$$(\beta - \beta_{\text{true}})^T X X^T (\beta - \beta_{\text{true}}) + \lambda \beta^T \beta - 2\tilde{z}_{\text{train}}^T X^T \beta$$



Ridge-regression coefficient estimate

$$
\begin{aligned}
\tilde{\beta}_{\text{RR}} &= \left(XX^T + \lambda I\right)^{-1} X \left(X^T \beta_{\text{true}} + \tilde{z}_{\text{train}}\right) \\
&= \left(US^2U^T + \lambda UU^T\right)^{-1} \left(US^2U^T \beta_{\text{true}} + USV^T \tilde{z}_{\text{train}}\right) \\
&= \left(U(S^2 + \lambda I)U^T\right)^{-1} \left(US^2U^T \beta_{\text{true}} + USV^T \tilde{z}_{\text{train}}\right) \\
&= U(S^2 + \lambda I)^{-1} U^T \left(US^2U^T \beta_{\text{true}} + USV^T \tilde{z}_{\text{train}}\right) \\
&= {\color{red} U(S^2 + \lambda I)^{-1} S^2 U^T \beta_{\text{true}}} + {\color{blue} U\left(S^2 + \lambda I\right)^{-1} SV^T \tilde{z}_{\text{train}}}
\end{aligned}
$$

Ridge-regression coefficient estimate

$$\tilde{\beta}_{\text{RR}} = U(S^2 + \lambda I)^{-1} S^2 U^T \beta_{\text{true}} + U \left(S^2 + \lambda I\right)^{-1} S V^T \tilde{z}_{\text{train}}$$

Distribution? Gaussian with mean

$$\beta_{\text{bias}} := \sum_{j=1}^{p} \frac{s_j^2 \left\langle u_j, \beta_{\text{true}} \right\rangle}{s_j^2 + \lambda} u_j$$

and covariance matrix

$$\Sigma_{\text{RR}} := \sigma^2 U \operatorname{diag}_{j=1}^{p} \left(\frac{s_j^2}{(s_j^2 + \lambda)^2} \right) U^T$$

Bias

In contrast to OLS, ridge regression produces systematic error

$$\mathrm{E}(\beta_{\text{true}} - \tilde{\beta}_{\text{RR}}) = \sum_{j=1}^{p} \left(\frac{\lambda \langle u_j, \beta_{\text{true}} \rangle}{s_j^2 + \lambda} - \frac{s_j \langle v_j, \mathrm{E}(\tilde{z}_{\text{train}}) \rangle}{s_j^2 + \lambda} \right) u_j$$

$$= \sum_{j=1}^{p} \frac{\lambda \langle u_j, \beta_{\text{true}} \rangle}{s_j^2 + \lambda} u_j$$

Bias grows with λ, so what's the point?

Variance

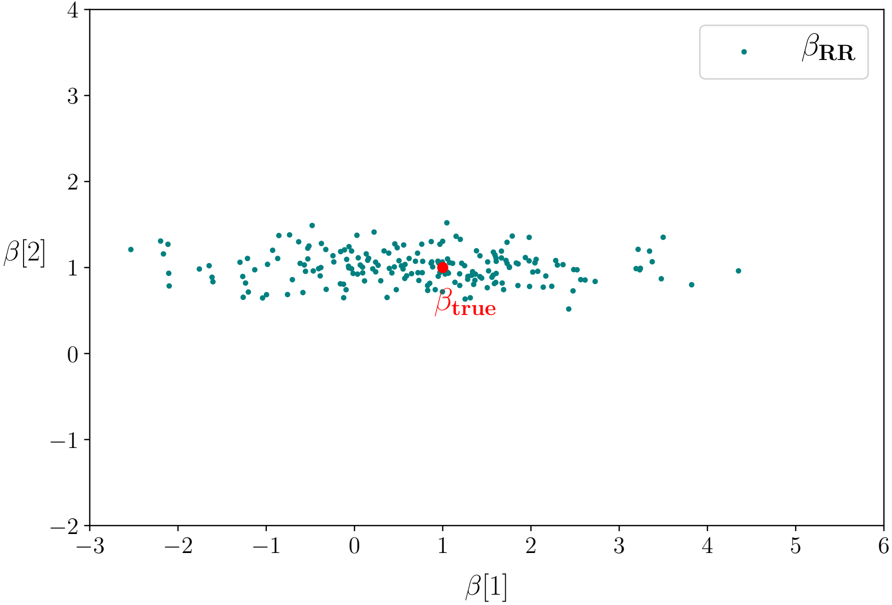
Variance in direction of u_i equals $\frac{\sigma^2 s_i^2}{(s_i^2+\lambda)^2}$

Small s_i blow up variance of OLS

If $\lambda \gg s_i^2$, then the variance $\approx \sigma^2 s_i^2/\lambda^2 \ll \sigma^2/s_i^2$ if s_i small

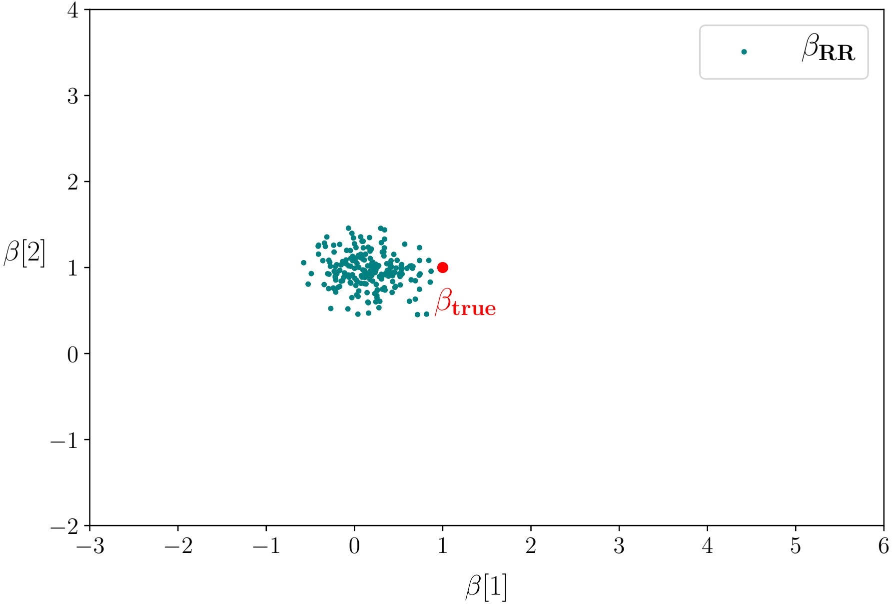
Ideal λ achieves bias-variance tradeoff

$\lambda = 0.005$

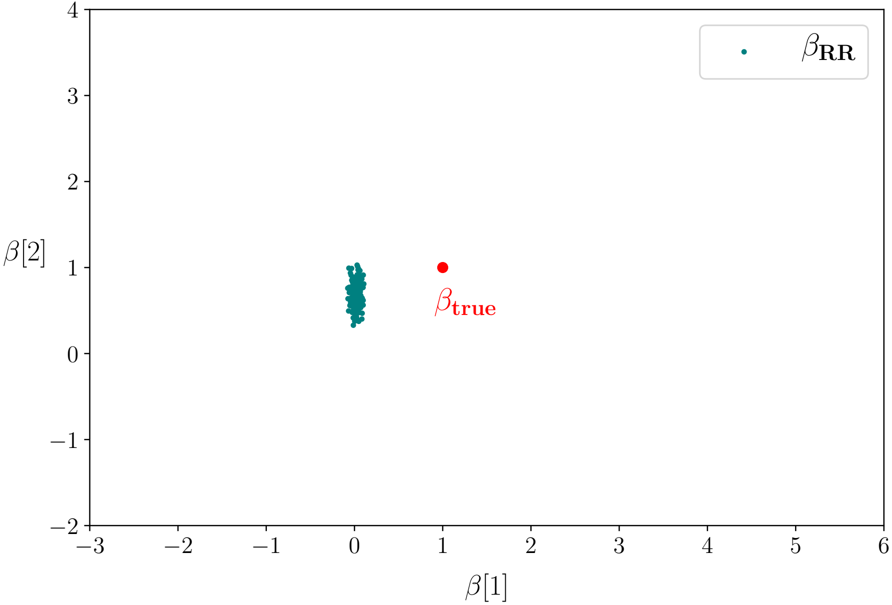


$\beta_{\mathbf{RR}}$
$\beta_{\mathbf{true}}$
$\beta[2]$
$\beta[1]$
4
3
2
1
0
−1
−2
−3
−2
−1
0
1
2
3
4
5
6

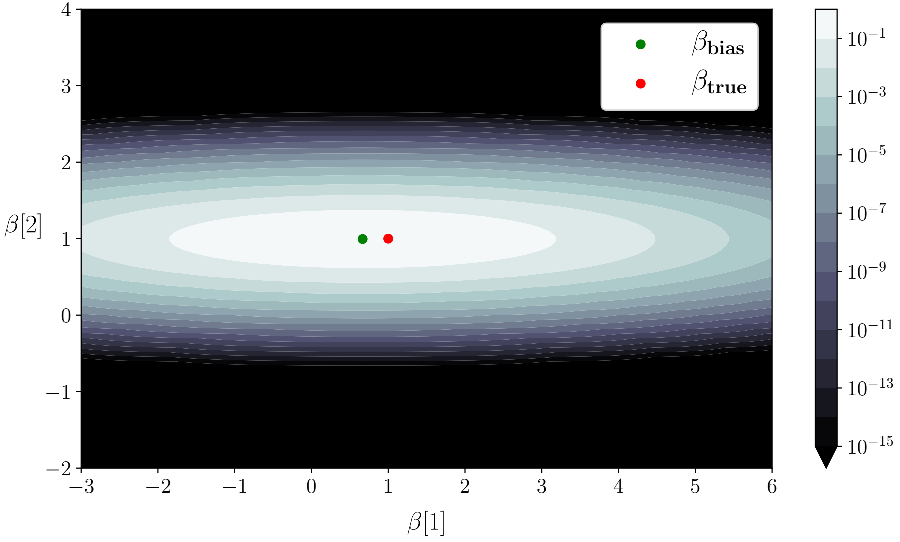
$\lambda = 0.05$



$\lambda = 0.5$

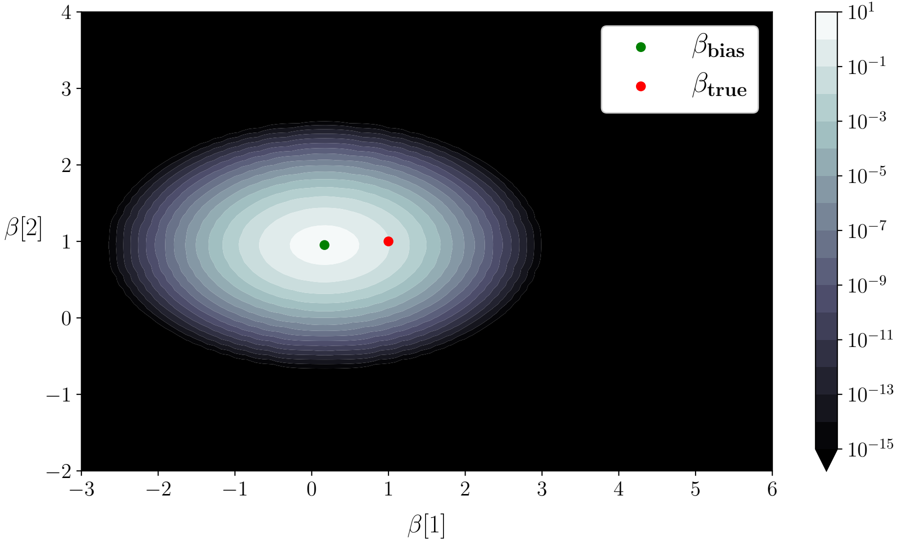


$\lambda = 0.005$



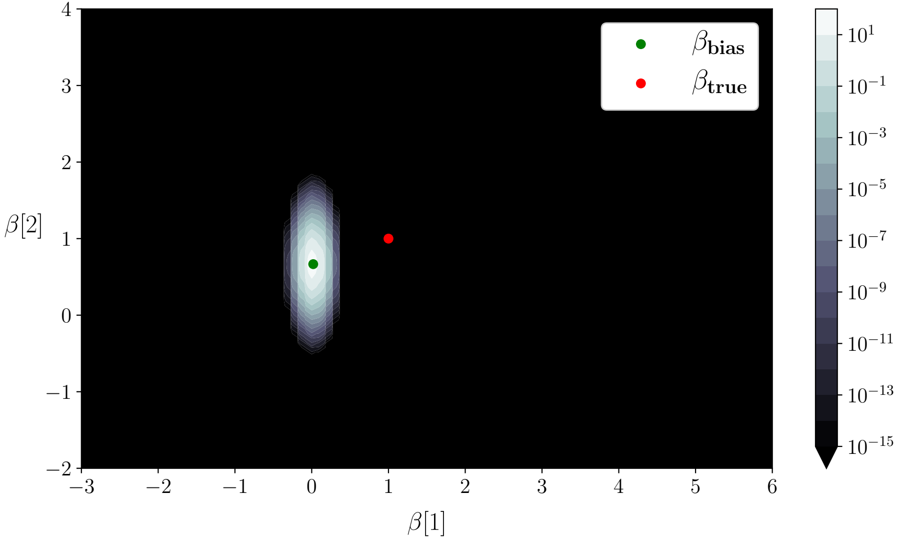
$\beta_{\mathbf{bias}}$
$\beta_{\mathbf{true}}$
$\beta[2]$
$\beta[1]$
4
3
2
1
0
−1
−2
−3
−2
−1
0
1
2
3
4
5
6
10^{-1}
10^{-3}
10^{-5}
10^{-7}
10^{-9}
10^{-11}
10^{-13}
10^{-15}

$\lambda = 0.05$



$\beta_{\mathbf{bias}}$
$\beta_{\mathbf{true}}$
$\beta[2]$
$\beta[1]$
4
3
2
1
0
−1
−2
−3
−2
−1
0
1
2
3
4
5
6
10^{1}
10^{-1}
10^{-3}
10^{-5}
10^{-7}
10^{-9}
10^{-11}
10^{-13}
10^{-15}

$\lambda = 0.5$



$\beta_{\mathbf{bias}}$
$\beta_{\mathbf{true}}$
$\beta[1]$
$\beta[2]$
10^{1}
10^{-1}
10^{-3}
10^{-5}
10^{-7}
10^{-9}
10^{-11}
10^{-13}
10^{-15}

Gradient descent

Intuition: Make local progress in the steepest direction $-\nabla f(x)$

Set the initial point $x^{(0)}$ to an arbitrary value

Update by setting

$$x^{(k+1)} := x^{(k)} - \alpha_k \nabla f\left(x^{(k)}\right)$$

where $\alpha_k > 0$ is the step size, until a stopping criterion is met

Least squares

Let $y \in \mathbb{R}^n$, $X \in \mathbb{R}^{p \times n}$, $\beta \in \mathbb{R}^p$

The gradient of the least-squares cost function

$$f(\beta) := \frac{1}{2} \left|\left| y - X^T \beta \right|\right|_2^2 = \frac{1}{2} y^T y + \frac{1}{2} \beta^T X X^T \beta - y^T X^T \beta$$

equals

$$\nabla f(\beta) = X(X^T \beta - y)$$

Gradient descent for least squares

Gradient descent updates are

$$\begin{aligned}\beta^{(k+1)} &= \beta^{(k)} + \alpha_k X \left(y - X^T \beta^{(k)}\right) \\ &= \beta^{(k)} + \alpha_k \sum_{i=1}^{n} \left(y_i - \langle x_i, \beta^{(k)} \rangle\right) x_i\end{aligned}$$

Gradient descent iterates, starting at origin

$$
\begin{aligned}
\beta^{(k+1)} &= \sum_{i=0}^{k} \left(I - \alpha X X^T\right)^i \alpha X y \\
&= \alpha \sum_{i=0}^{k} \left(U U^T - \alpha U S^2 U^T\right)^i U S V^T y \\
&= \alpha U \sum_{i=0}^{k} \left(I - \alpha S^2\right)^i U^T U S V^T y \\
&= \alpha U \operatorname{diag}_{j=1}^{p} \left(\sum_{i=0}^{k} \left(1 - \alpha s_j^2\right)^i \right) S V^T y \\
&= \alpha U \operatorname{diag}_{j=1}^{p} \left(\frac{1 - \left(1 - \alpha s_j^2\right)^{k+1}}{\alpha s_j} \right) V^T y
\end{aligned}
$$

Convergence

Condition for convergence? $\left|1 - \alpha s_j^2\right| < 1$

In that case

$$\lim_{k\to\infty} \beta^{(k)} = \lim_{k\to\infty} U \operatorname{diag}_{j=1}^{p} \left(\frac{1 - \left(1 - \alpha s_j^2\right)^k}{s_j} \right) V^T y$$

$$= US^{-1} V^T y = \beta_{\text{OLS}}$$

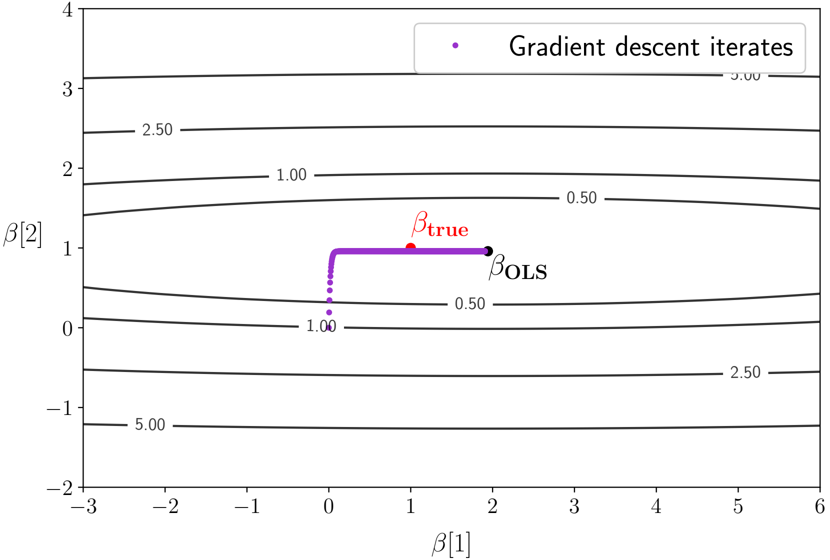
Guaranteed by $\alpha \leq 2/s_1^2$

Convergence rate

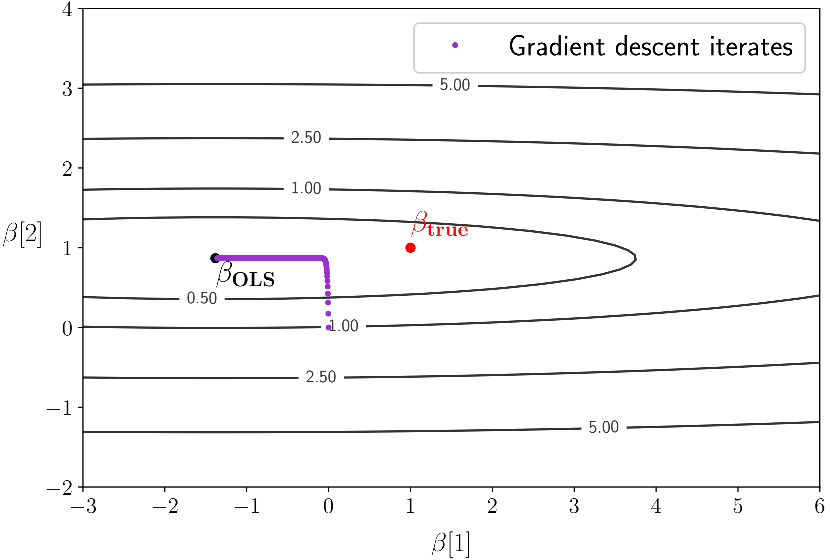
$$\beta^{(k+1)} = \alpha U \operatorname{diag}_{j=1}^{p} \left(\frac{1 - \left(1 - \alpha s_j^2\right)^{k+1}}{\alpha s_j} \right) V^T y$$

If $\alpha \approx 1/s_1^2$ convergence of each component governed by s_j/s_1

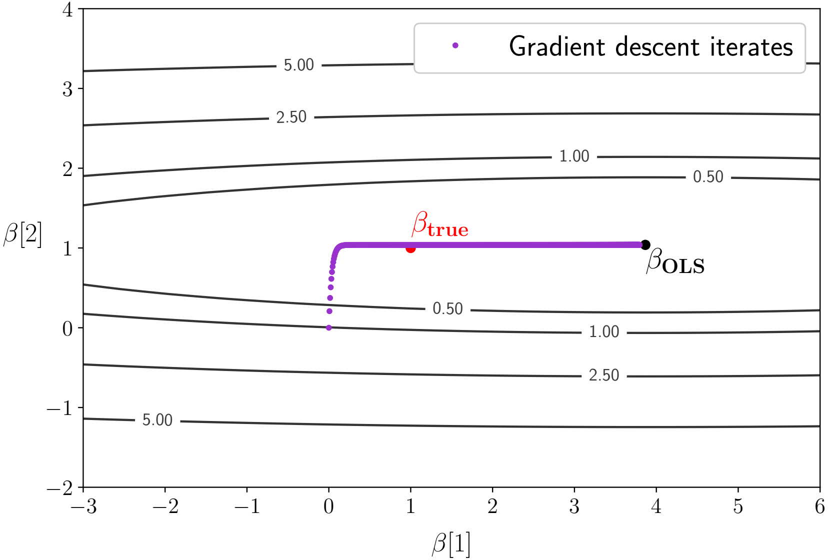
Additive model ($s_1 = 1$, $s_2 = 0.1$)



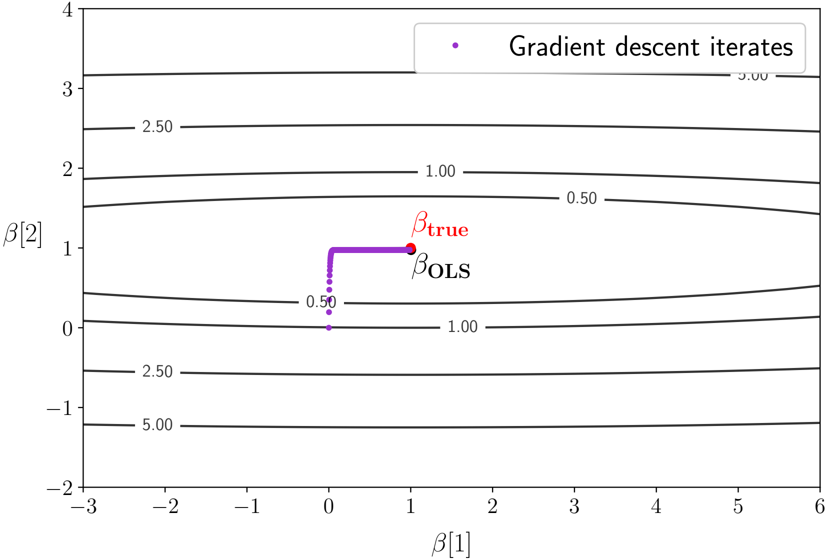
Additive model ($s_1 = 1$, $s_2 = 0.1$)



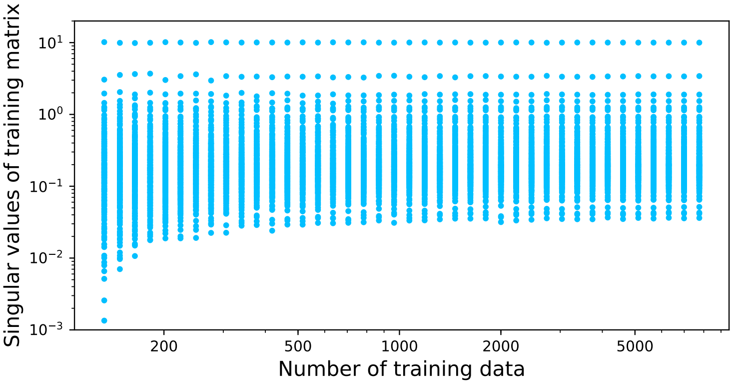
Additive model ($s_1 = 1$, $s_2 = 0.1$)



Additive model ($s_1 = 1$, $s_2 = 0.1$)



Temperature prediction via linear regression



Gradient descent for linear regression

Bad news: Convergence very slow

Wait, what do we care about?

Additive model

Assume additive model for regression problem

$$y_{\text{train}} := X^T \beta_{\text{true}} + \tilde{z}_{\text{train}}$$

Estimate coefficients via gradient descent up to iteration k

Gradient descent iterates

$$\tau_j := 1 - \alpha s_j^2$$

$$\begin{aligned}
\tilde{\beta}^{(k)} &= U \operatorname{diag}_{j=1}^{p} \left(\frac{1 - \tau_j^k}{s_j} \right) V^T \left(X^T \beta_{\text{true}} + \tilde{z}_{\text{train}} \right) \\
&= U \operatorname{diag}_{j=1}^{p} \left(\frac{1 - \tau_j^k}{s_j} \right) V^T \left(V S U^T \beta_{\text{true}} + \tilde{z}_{\text{train}} \right) \\
&= U \operatorname{diag}_{j=1}^{p} \left(1 - \tau_j^k \right) U^T \beta_{\text{true}} + U \operatorname{diag}_{j=1}^{p} \left(\frac{1 - \tau_j^k}{s_j} \right) V^T \tilde{z}_{\text{train}}
\end{aligned}$$

Gradient descent coefficient estimate

$$\tilde{\beta}_{\text{GD}} = U \operatorname{diag}_{j=1}^{p} \left(1 - \tau_j^k\right) U^T \beta_{\text{true}} + U \operatorname{diag}_{j=1}^{p} \left(\frac{1 - \tau_j^k}{s_j}\right) V^T \tilde{z}_{\text{train}}$$

Distribution? Gaussian with mean

$$\beta_{\text{bias}} := \sum_{j=1}^{p} \left(1 - (1 - \alpha s_j^2)^k\right) \langle u_j, \beta_{\text{true}} \rangle \, u_j$$

and covariance matrix

$$\Sigma_{\text{GD}} := \sigma^2 U \operatorname{diag}_{j=1}^{p} \left(\frac{(1 - (1 - \alpha s_j^2)^k)^2}{s_j^2}\right) U^T$$

Bias

Like ridge regression, early stopping produces systematic error

$$\mathrm{E}(\beta_{\text{true}} - \tilde{\beta}_{\text{GD}}) = \sum_{j=1}^{p} (1 - \alpha s_j^2)^k \langle u_j, \beta_{\text{true}} \rangle \, u_j$$

Bias decreases with k

Variance

Variance in direction of u_i equals $\frac{\sigma^2(1-(1-\alpha s_j^2)^k)^2}{s_j^2}$

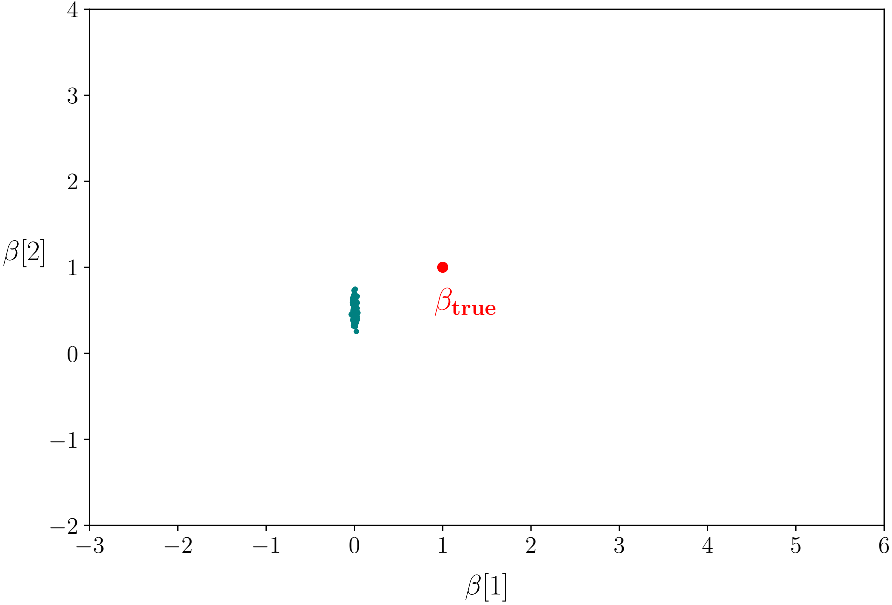
Small s_i blow up variance of OLS

For small k and αs_j, $(1-\alpha s_j^2)^k \approx 1 - k\alpha s_j^2$

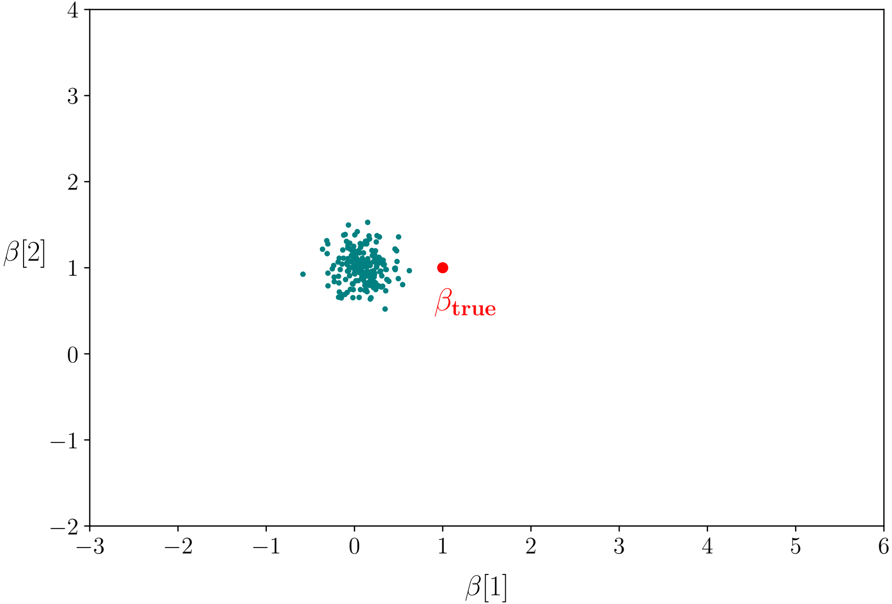
The corresponding variance component $\approx \sigma^2 k\alpha$

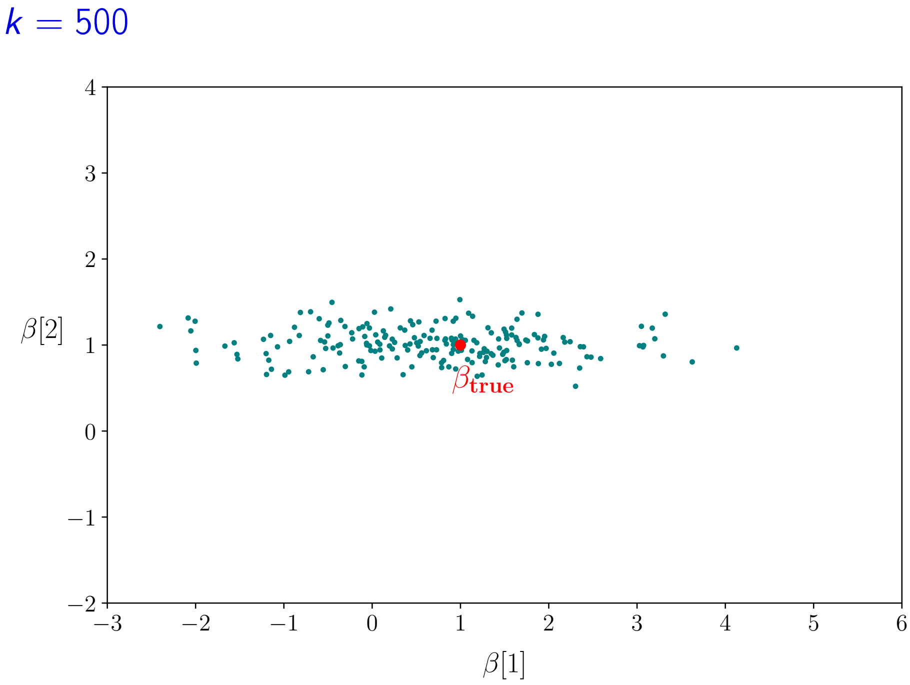
Ideal λ achieves bias-variance tradeoff

$k = 3$

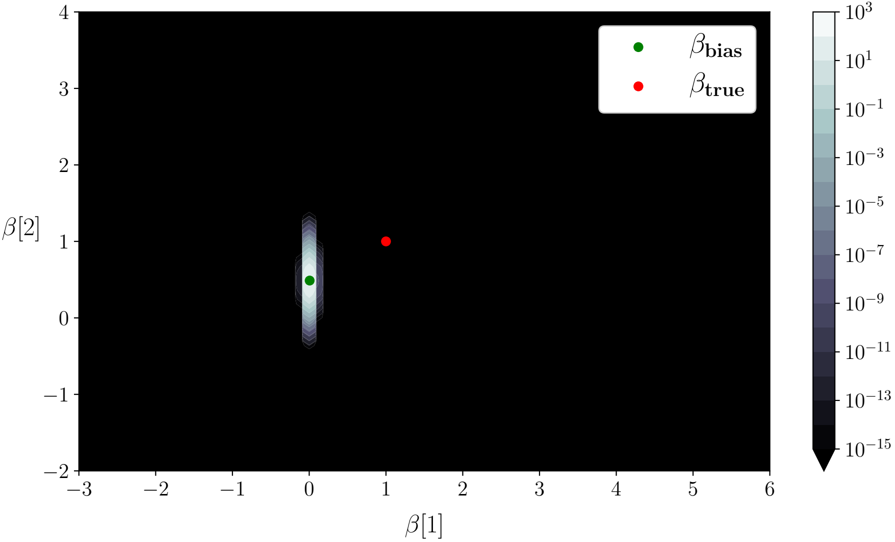


$k = 50$



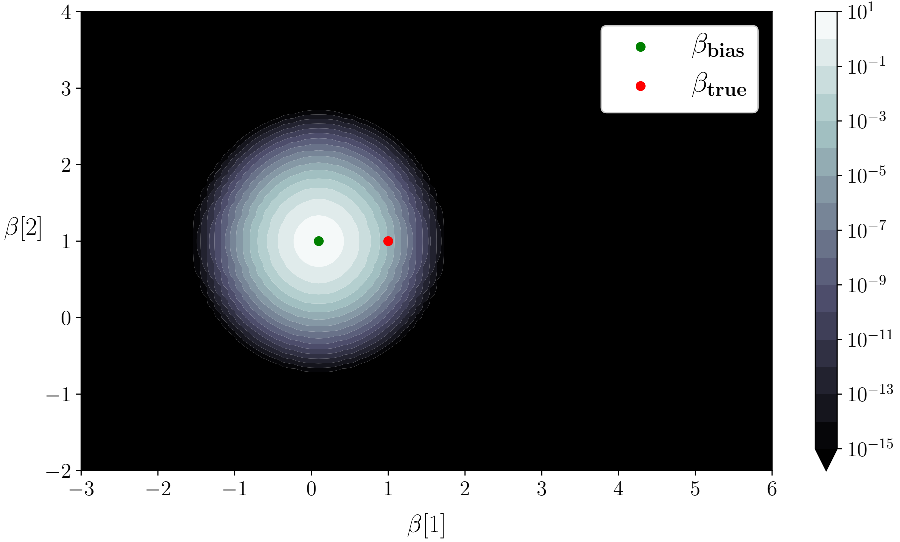

$k = 500$
$\beta_{\mathbf{true}}$
$\beta[2]$
$\beta[1]$
4
3
2
1
0
−1
−2
−3
−2
−1
0
1
2
3
4
5
6

$k = 3$



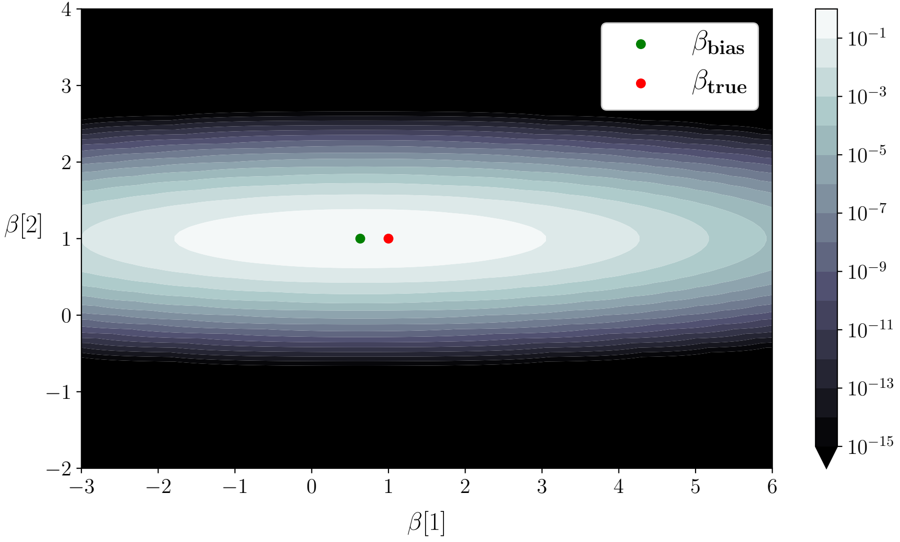
$\beta_{\mathbf{bias}}$
$\beta_{\mathbf{true}}$
$\beta[2]$
$\beta[1]$
4
3
2
1
0
−1
−2
−3
−2
−1
0
1
2
3
4
5
6
10^{3}
10^{1}
10^{-1}
10^{-3}
10^{-5}
10^{-7}
10^{-9}
10^{-11}
10^{-13}
10^{-15}

$k = 50$



β_bias
β_true
β[2]
β[1]
4
3
2
1
0
−1
−2
−3
−2
−1
0
1
2
3
4
5
6
10^1
10^−1
10^−3
10^−5
10^−7
10^−9
10^−11
10^−13
10^−15

$k = 500$

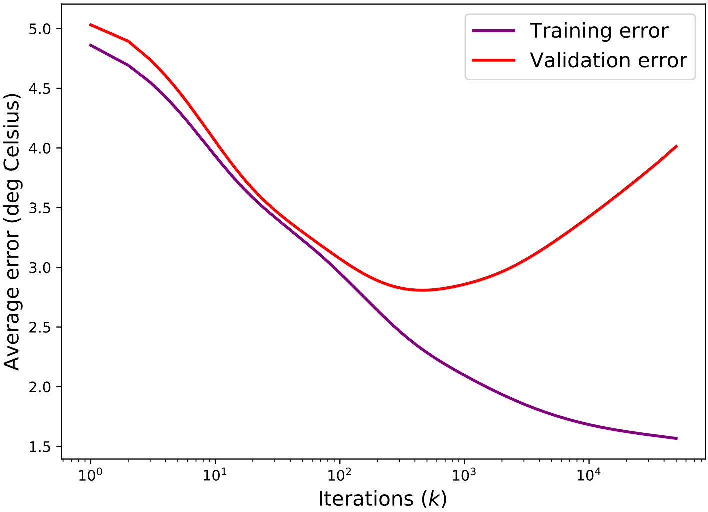


$\beta_{\mathbf{bias}}$
$\beta_{\mathbf{true}}$
$\beta[2]$
$\beta[1]$
4
3
2
1
0
−1
−2
−3
−2
−1
0
1
2
3
4
5
6
10^{-1}
10^{-3}
10^{-5}
10^{-7}
10^{-9}
10^{-11}
10^{-13}
10^{-15}

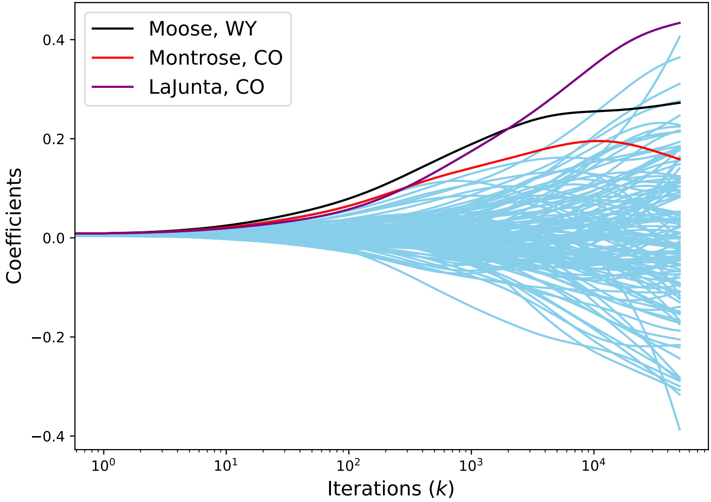
Temperature prediction via linear regression

- Dataset of hourly temperatures measured at weather stations all over the US
- Goal: Predict temperature in Yosemite from other temperatures
- Response: Temperature in Yosemite
- Features: Temperatures in 133 other stations ($p = 133$) in 2015
- Test set: 10^3 measurements
- Additional test set: All measurements from 2016

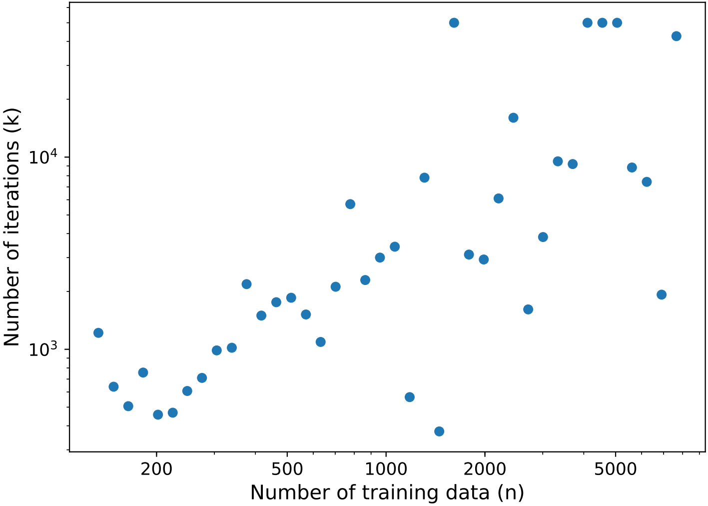
Gradient-descent estimator ($n = 200$)



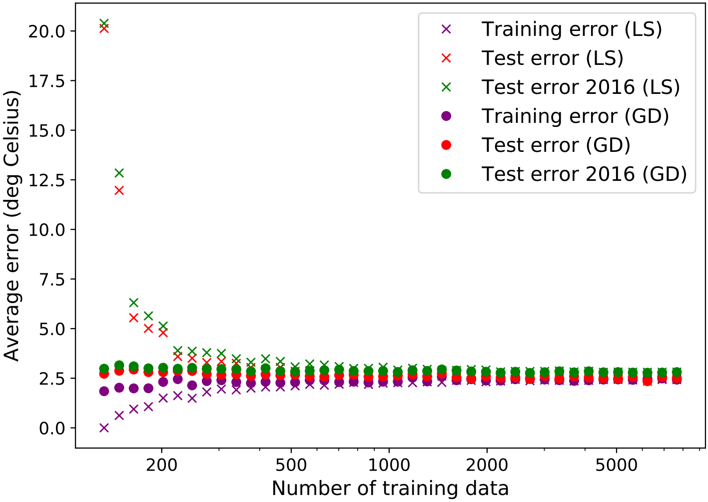
Gradient-descent estimator ($n = 200$)



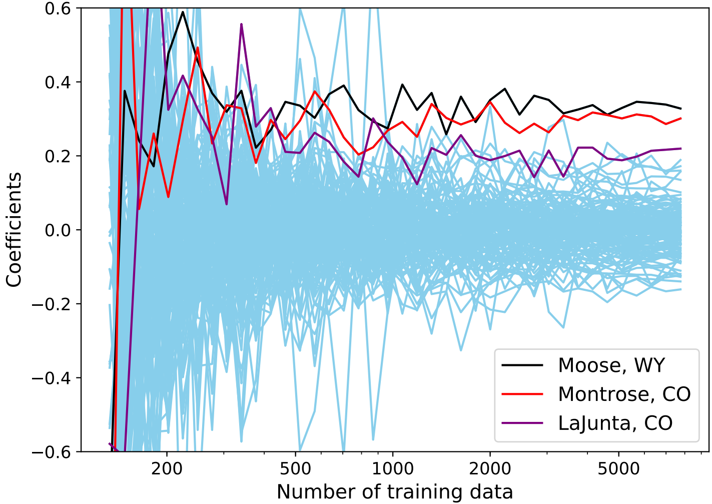
Selected number of iterations



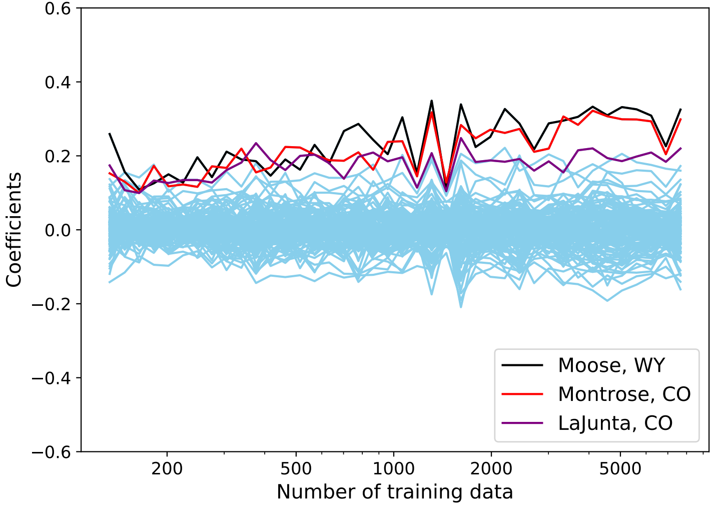
Comparison to least squares



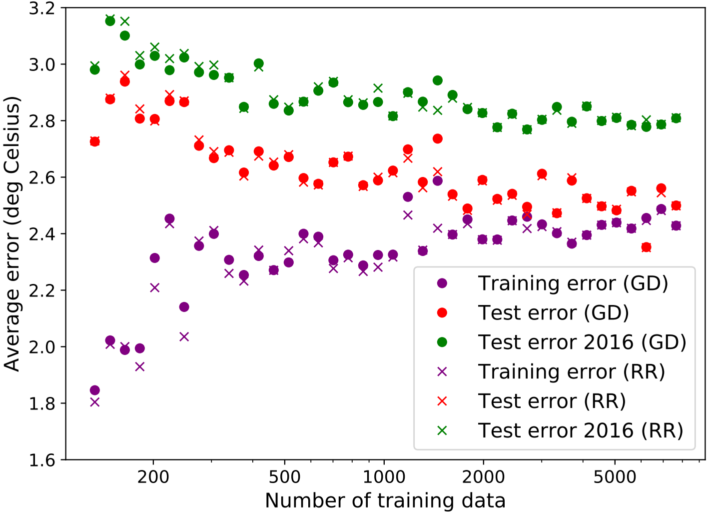
Least-squares coefficients



Gradient-descent coefficients



Comparison to ridge regression



Ridge-regression coefficients