Linear regression

1 Overview

The topic of this chapter is linear regression. In Section 2 we motivate linear estimation, derive the linear estimate that minimizes mean square error in a probabilistic setting, and introduce ordinary-least-squares estimation. Section 3 is dedicated to the singular-value decomposition, a fundamental tool to analyze linear operators. In Section 4 we analyze the error incurred by the ordinary-least-squares estimator. Section 5 describes ridge regression, a method to enhance the performance of the least-squares estimators by leveraging regularization. Finally, Section 6 provides an analysis of gradient descent, and of the advantages of early stopping.

2 Mean square error and ordinary least squares

Regression is a fundamental problem in statistics. The goal is to estimate a quantity of interest called the *response* or *dependent variable* from the values of several observed variables known as *covariates*, *features* or *independent variables*. Let us model the response as a random variable $\tilde{y}$, and the features as the entries of a p-dimensional random vector $\tilde{x}$. Our goal is to produce an estimate of $\tilde{y}$ as a function of $\tilde{x}$. A popular evaluation measure for this problem is mean square error. If we observe that $\tilde{x}$ equals a fixed value x, the uncertainty about $\tilde{y}$ is captured by the distribution of $\tilde{y}$ given $\tilde{x} = x$. Let $\tilde{y}'$ be a random variable that follows that distribution. Minimizing the mean square error for the fixed observation $\tilde{x} = x$ is exactly equivalent to finding a constant vector c that minimizes $\mathrm{E}[(\tilde{y}' - c)^2]$. Lemma 4.1 in the notes on PCA establishes that the optimal vector is the mean of the distribution. Consequently, the optimal estimator is the conditional mean $\mathrm{E}(\tilde{y} \,|\, \tilde{x} = x)$, as stated in the following theorem.

Theorem 2.1 (MMSE estimator). *Let $\tilde{x}$ and $\tilde{y}$ be real-valued random variables or random vectors. If $\tilde{x} = x$ then the minimum-mean-square-error (MMSE) estimator of $\tilde{y}$ given $\tilde{x}$ is the conditional expectation of $\tilde{y}$ given $\tilde{x} = x$, i.e.*

$$\mathrm{E}(\tilde{y} \,|\, \tilde{x} = x) = \arg\min_{w} \mathrm{E}\left[(\tilde{y} - w)^2 \,|\, \tilde{x} = x\right]. \tag{1}$$

The theorem suggests that in order to solve the regression problem, all we need to do is compute the average value of the response corresponding to every possible value of the features. The catch is that when there is more than one or two features this requires too many data. As a simple example, consider a problem with p features each taking d different values. In order to be able to perform estimation, we need to compute the expected value of the response conditioned on every possible value of the feature vector. However there are $N = d^p$ possible values! For even moderate values of p and d the number is huge: if $p = 5$, and $d = 100$ then $N = 10^{10}$! This is known as the curse of dimensionality (where dimensionality refers to the dimension of the feature vector).

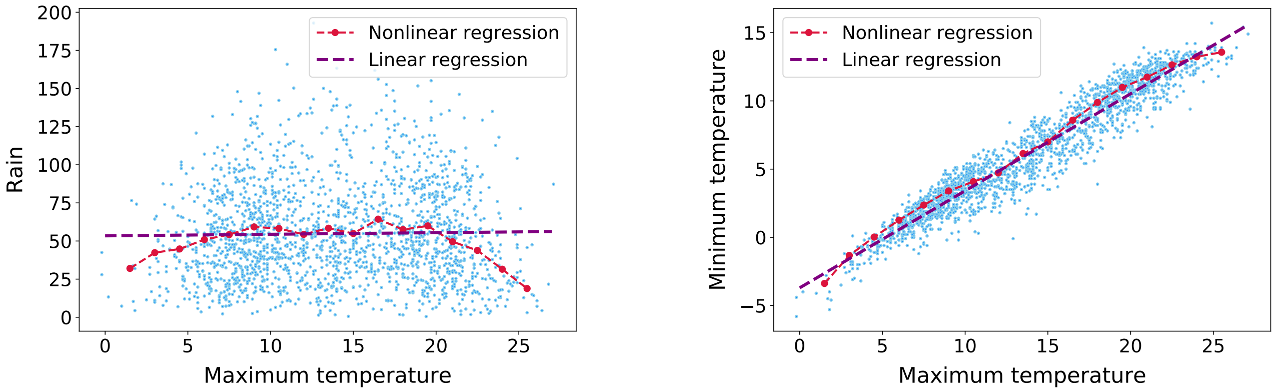


Figure 1: Regression models for weather measurements gathered at a weather station in Oxford over 150 years. On the left, the response is the monthly amount of rain, and the feature the maximum temperature during the same month. On the right, the response and the feature are the minimum and maximum monthly temperature respectively. The linear regression model is compute by minimizing the least-squares fit. The nonlinear regression model is computed by averaging the response over all values of the feature in bins of width equal to 1.5 degrees. In the case of the rain, the linear model cannot capture the fact that at high temperatures, rain and temperature are negatively correlated (see Figure 2 in the lecture notes on PCA).

In general, tackling the regression problem requires making assumptions about the relationship between the response and the features. A simple, yet often surprisingly effective, assumption is that the relationship is linear (or rather affine), i.e. that there exists a constant vector $\beta \in \mathbb{R}^p$ and a constant $\beta_0 \in \mathbb{R}$ such that

$$\tilde{y} \approx \beta^T \tilde{x} + \beta_0. \tag{2}$$

Mathematically, the gradient of the regression function is constant, which means that the rate of change in the response with respect to the features does not depend on the feature values. This is illustrated in Figure 1, which compares a linear model with a nonlinear model for two simple examples where there is only one feature[1]. The slope of the nonlinear estimate varies depending on the feature, but the slope of the linear model is constrained to be constant.

The following lemma establishes that when fitting an affine model by minimizing mean square error, we can just center the response and the features, and fit a linear model without additive constants.

Lemma 2.2. *For any $\beta \in \mathbb{R}^p$ and any random variable $\tilde{y}$ and p-dimensional random vector $\tilde{x}$,*

$$\min_{\beta_0} \mathrm{E}\left[(\tilde{y} - \tilde{x}^T\beta - \beta_0)^2\right] = \mathrm{E}\left[(c(\tilde{y}) - c(\tilde{x})^T\beta)^2\right], \tag{3}$$

where $c(\tilde{y}) := \tilde{y} - \mathrm{E}(\tilde{y})$ and $c(\tilde{x}) := \tilde{x} - \mathrm{E}(\tilde{x})$.

[1] The data are available here.

Proof. By Lemma 4.1 in the notes on PCA, the optimal β_0 equals $\mathrm{E}(\tilde{y} - \tilde{x}^T\beta)$, so

$$\min_{\beta_0} \mathrm{E}\left[(\tilde{y} - \tilde{x}^T\beta - \beta_0)^2\right] = \mathrm{E}\left[(\tilde{y} - \tilde{x}^T\beta - \mathrm{E}(\tilde{y}) + \mathrm{E}(\tilde{x})^T\beta)^2\right] \tag{4}$$

$$= \mathrm{E}\left[(c(\tilde{y}) - \beta^T c(\tilde{x}))^2\right]. \tag{5}$$

□

From now on, we will assume that the response and the features are centered. The following theorem derives the optimal linear estimator in terms of MSE when the response and features are modeled as random variables.

Theorem 2.3 (Linear MMSE). *Let $\tilde{y}$ be a zero-mean random variable and $\tilde{x}$ a zero mean random vector with a full-rank covariance matrix equal to $\Sigma_{\tilde{x}}$, then*

$$\Sigma_{\tilde{x}}^{-1}\Sigma_{\tilde{x}\tilde{y}} = \arg\min_{\beta} \mathrm{E}\left[(\tilde{y} - \tilde{x}^T\beta)^2\right], \tag{6}$$

where $\Sigma_{\tilde{x}\tilde{y}}$ is the cross-covariance between $\tilde{x}$ and $\tilde{y}$:

$$\Sigma_{\tilde{x}\tilde{y}}[i] := \mathrm{E}\left(\tilde{x}[i]\tilde{y}\right), \quad 1 \leq i \leq p. \tag{7}$$

The MSE of this estimator equals $\mathrm{Var}(\tilde{y}) - \Sigma_{\tilde{x}\tilde{y}}^T\Sigma_{\tilde{x}}^{-1}\Sigma_{\tilde{x}\tilde{y}}$.

Proof. We have

$$\mathrm{E}((\tilde{y} - \tilde{x}^T\beta)^2) = \mathrm{E}\left(\tilde{y}^2\right) - 2\mathrm{E}\left(\tilde{y}\tilde{x}\right)^T\beta + \beta^T\mathrm{E}(\tilde{x}\tilde{x}^T)\beta \tag{8}$$

$$= \beta^T\Sigma_{\tilde{x}}\beta - 2\Sigma_{\tilde{x}\tilde{y}}^T\beta + \mathrm{Var}\left(\tilde{y}\right) := f(\beta). \tag{9}$$

The function f is a quadratic form. Its gradient and Hessian equal

$$\nabla f(\beta) = 2\Sigma_{\tilde{x}}\beta - 2\Sigma_{\tilde{x}\tilde{y}}, \tag{10}$$

$$\nabla^2 f(\beta) = 2\Sigma_{\tilde{x}}. \tag{11}$$

Covariance matrices are positive semidefinite. For any vector $v \in \mathbb{R}^p$

$$v^T\Sigma_{\tilde{x}}v = \mathrm{Var}\left(v^T\tilde{x}\right) \geq 0. \tag{12}$$

Since $\Sigma_{\tilde{x}}$ is full rank, it is actually positive definite, i.e. the inequality is strict as long as $v \neq 0$. This means that the quadratic function is strictly convex and we can set its gradient to zero to find its unique minimum. For the sake of completeness, we provide a simple proof of this. The quadratic form is exactly equal to its second-order Taylor expansion around any point $\beta_1 \in \mathbb{R}^p$. For all $\beta_2 \in \mathbb{R}^p$

$$f(\beta_2) = \frac{1}{2}(\beta_2 - \beta_1)^T\nabla^2 f(\beta_1)(\beta_2 - \beta_1) + \nabla f(\beta_1)^T(\beta_2 - \beta_1) + f(\beta_1). \tag{13}$$

The equality can be verified by expanding the expression. This means that if $\nabla f(\beta^*) = 0$ then for any $\beta \neq \beta^*$

$$f(\beta) = \frac{1}{2}(\beta - \beta^*)^T\nabla^2 f(\beta^*)(\beta - \beta^*) + f(\beta^*) > f(\beta^*) \tag{14}$$

because $\nabla^2 f(\beta^*) = \Sigma_{\tilde{x}}$ is positive definite. The unique minimum can therefore be found by setting the gradient to zero. Finally, the corresponding MSE equals

$$\begin{aligned} \mathrm{E}\left[(\tilde{y} - \tilde{x}^T \Sigma_{\tilde{x}}^{-1} \Sigma_{\tilde{x}\tilde{y}})^2\right] &= \mathrm{E}(\tilde{y}^2) + \Sigma_{\tilde{x}\tilde{y}}^T \Sigma_{\tilde{x}}^{-1} \mathrm{E}(\tilde{x}\tilde{x}^T) \Sigma_{\tilde{x}}^{-1} \Sigma_{\tilde{x}\tilde{y}} - 2\mathrm{E}(\tilde{y}\tilde{x}^T) \Sigma_{\tilde{x}}^{-1} \Sigma_{\tilde{x}\tilde{y}} \\ &= \mathrm{Var}(\tilde{y}) - \Sigma_{\tilde{x}\tilde{y}}^T \Sigma_{\tilde{x}}^{-1} \Sigma_{\tilde{x}\tilde{y}}. \end{aligned} \tag{15}$$

□

The theorem shows that the optimal linear estimator only depends on the covariance and cross-covariance of the random variables. In practice, we must estimate these quantities from a finite number of data. Assume that we have available n examples consisting of feature vectors coupled with their respective response: (y_1, x_1), (y_2, x_2), ..., (y_n, x_n), where $y_i \in \mathbb{R}$ and $x_i \in \mathbb{R}^p$ for $1 \leq i \leq n$. We define a response vector $y \in \mathbb{R}^n$, such that $y[i] := y_i$, and a feature matrix $X \in \mathbb{R}^{p \times n}$ with columns equal to the feature vectors,

$$X := \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}. \tag{16}$$

If we interpret the feature data as samples of $\tilde{x}$ and the corresponding response values as samples of $\tilde{y}$, a reasonable estimate for the covariance matrix is the sample covariance matrix,

$$\frac{1}{n} X X^T = \frac{1}{n} \sum_{i=1}^n x_i x_i^T. \tag{17}$$

Similarly, the cross-covariance can be approximated by the sample cross-covariance, which contains the sample covariance between each feature and the response,

$$\frac{1}{n} X y = \begin{bmatrix} \frac{1}{n}\sum_{i=1}^n x_i[1] y[1] \\ \frac{1}{n}\sum_{i=1}^n x_i[2] y[2] \\ \cdots \\ \frac{1}{n}\sum_{i=1}^n x_i[p] y[p] \end{bmatrix}. \tag{18}$$

We therefore obtain the following approximation to the linear MMSE estimate derived in Theorem 2.3,

$$\Sigma_{\tilde{x}}^{-1} \Sigma_{\tilde{x}\tilde{y}} \approx (X X^T)^{-1} X y. \tag{19}$$

This estimate has an alternative interpretation, which does not require probabilistic assumptions: it minimizes the least-squares fit between the observed values of the response and the linear model. In the statistics literature, this method is known as ordinary least squares (OLS).

Theorem 2.4 (Ordinary least squares). *If $X := \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix} \in \mathbb{R}^{p \times n}$ is full rank and $n \geq p$, for any $y \in \mathbb{R}^n$ we have*

$$\begin{aligned} \beta_{\mathrm{OLS}} &:= \arg\min_{\beta} \sum_{i=1}^n \left(y_i - x_i^T \beta\right)^2 \qquad (20) \\ &= (X X^T)^{-1} X y. \qquad (21) \end{aligned}$$

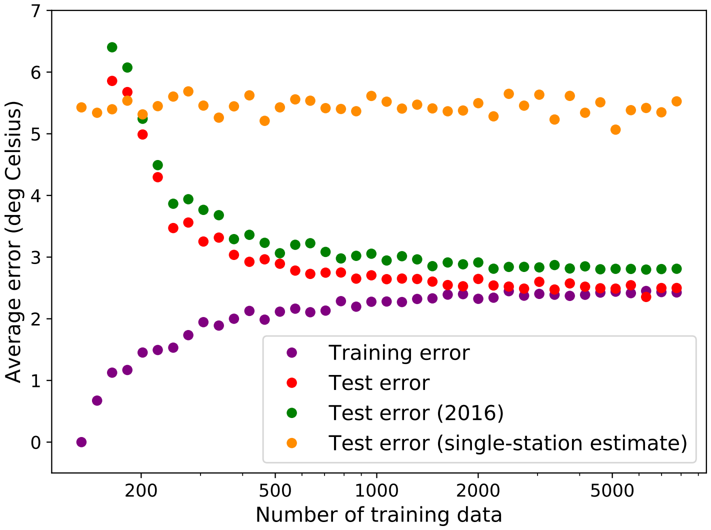


Figure 2: Performance of the least-squares estimator on the temperature data described in Example 2.5. The graph shows the square root of the MSE (RMSE) achieved by the model on the training and test sets, and on the 2016 data, for different number of training data and compares it to the RMSE of the best single-station estimate.

Proof.

$$\sum_{i=1}^{n} \left(y_i - x_i^T \beta\right)^2 = \|y - X^T \beta\|_2^2 \tag{22}$$

$$= \beta^T X X^T \beta - 2 y^T X^T \beta + y^T y := f(\beta). \tag{23}$$

The function f is a quadratic form. Its gradient and Hessian equal

$$\nabla f(\beta) = 2 X X^T \beta - 2 X y, \tag{24}$$

$$\nabla^2 f(\beta) = 2 X X^T. \tag{25}$$

Since X is full rank, $X X^T$ is positive definite because for any nonzero vector v

$$v^T X X^T v = \left|\left|X^T v\right|\right|_2^2 > 0. \tag{26}$$

By the same argument in Theorem 2.3, the unique minimum can be found by setting the gradient to zero. □

In practice, large-scale least-squares problems are not solved by using the closed-form solution, due to the computational cost of inverting the sample covariance matrix of the features, but rather by applying iterative optimization methods such as conjugate gradients.

Example 2.5 (Temperature prediction via linear regression)**.** We consider a dataset of hourly temperatures measured at weather stations all over the United States[2]. Our goal is to design a

[2] The data are available at http://www1.ncdc.noaa.gov/pub/data/uscrn/products

model that can be used to estimate the temperature in Yosemite Valley from the temperatures of 133 other stations, in case the sensor in Yosemite fails. We perform estimation by fitting a linear model where the response is the temperature in Yosemite and the features are the rest of the temperatures ($p = 133$). We use 10^3 measurements from 2015 as a test set, and train a linear model using a variable number of training data also from 2015 but disjoint from the test data. In addition, we test the linear model on data from 2016. Figure 2 shows the results. With enough data, the linear model achieves an error of roughly 2.5°C on the test data, and 2.8°C on the 2016 data. The linear model outperforms a naive single-station estimate, which uses the station that best predicts the temperature in Yosemite for the training data. △

3 The singular-value decomposition

In order to gain further insight into linear models we introduce a fundamental tool in linear algebra: the singular-value decomposition.

Theorem 3.1 (Singular-value decomposition). *Every real matrix $A \in R^{m\times k}$, $m \geq k$, has a singular-value decomposition (SVD) of the form*

$$\begin{align}
A &= \begin{bmatrix} u_1 & u_2 & \cdots & u_k \end{bmatrix} \begin{bmatrix} s_1 & 0 & \cdots & 0 \\ 0 & s_2 & \cdots & 0 \\ & & \ddots & \\ 0 & 0 & \cdots & s_k \end{bmatrix} \begin{bmatrix} v_1 & v_2 & \cdots & v_k \end{bmatrix}^T \tag{27}\\
&= USV^T, \tag{28}
\end{align}$$

where the singular values $s_1 \geq s_2 \geq \cdots \geq s_k$ are nonnegative real numbers, the left singular vectors $u_1, u_2, \ldots u_k \in \mathbb{R}^m$ form an orthonormal set, and the right singular vectors $v_1, v_2, \ldots v_k \in \mathbb{R}^k$ also form an orthonormal set.

If $m < k$ then the SVD is of the form

$$\begin{align}
A &= \begin{bmatrix} u_1 & u_2 & \cdots & u_m \end{bmatrix} \begin{bmatrix} s_1 & 0 & \cdots & 0 \\ 0 & s_2 & \cdots & 0 \\ & & \ddots & \\ 0 & 0 & \cdots & s_m \end{bmatrix} \begin{bmatrix} v_1 & v_2 & \cdots & v_m \end{bmatrix}^T \tag{29}\\
&= USV^T, \tag{30}
\end{align}$$

where $s_1 \geq s_2 \geq \cdots \geq s_m$ are nonnegative real numbers, and the singular vectors $u_1, u_2, \ldots u_m \in \mathbb{R}^m$, and $v_1, v_2, \ldots v_m \in \mathbb{R}^k$ form orthonormal sets.

Proof. We prove the case $m \geq k$; the case $m < k$ then follows directly by applying the result to the transpose of the matrix. Let $V\Lambda V^T$ be the eigendecomposition of A^TA, where Λ contains the eigenvalues $\lambda_1 \geq \lambda_2 \geq \ldots \lambda_k$. These eigenvalues are nonnegative, because for the ith eigenvector v_i

$$\begin{align}
||Av_i||_2^2 &= v_i^T A^T A v_i \tag{31}\\
&= \lambda_i v_i^T v_i \tag{32}\\
&= \lambda_i. \tag{33}
\end{align}$$

Let k_+ denote the number of nonzero eigenvalues. For $1 \leq i \leq k_+$ we define $s_i := \sqrt{\lambda_i}$ and

$$u_i := \frac{1}{s_i} A v_i. \tag{34}$$

These vectors are unit norm,

$$\begin{aligned} ||u_i||_2^2 &= \frac{1}{s_i^2} v_i^T A^T A v_i && (35) \\ &= \frac{\lambda_i}{\lambda_i} v_i^T v_i && (36) \\ &= 1, && (37) \end{aligned}$$

and orthogonal,

$$\begin{aligned} \langle u_i, u_j \rangle &= \frac{v_i^T A^T A v_j}{s_i s_j} && (38) \\ &= \frac{\lambda_j v_i^T v_j}{s_i s_j} && (39) \\ &= 0, && (40) \end{aligned}$$

because $v_i^T v_j = 0$ for $i \neq j$. Let $u_{k_+ +1}, u_{k_+ +2}, \ldots, u_k$ be an orthonormal set of vectors, which are also orthogonal to $u_1, \ldots, u_{k_+}$, and let $s_i := 0$, for $k_+ < i \leq k$. We define an orthogonal matrix $U := \begin{bmatrix} u_1 & u_2 & \cdots & u_k \end{bmatrix}$ and a diagonal matrix S, such that $S_{ii} := s_i$ for $1 \leq i \leq k$. Then,

$$AV = US. \tag{41}$$

Since V is an orthogonal matrix,

$$A = USV^T. \tag{42}$$

□

The SVD provides a very intuitive geometric interpretation of the action of a matrix $A \in \mathbb{R}^{m \times k}$ on a vector $w \in \mathbb{R}^k$, as illustrated in Figure 3:

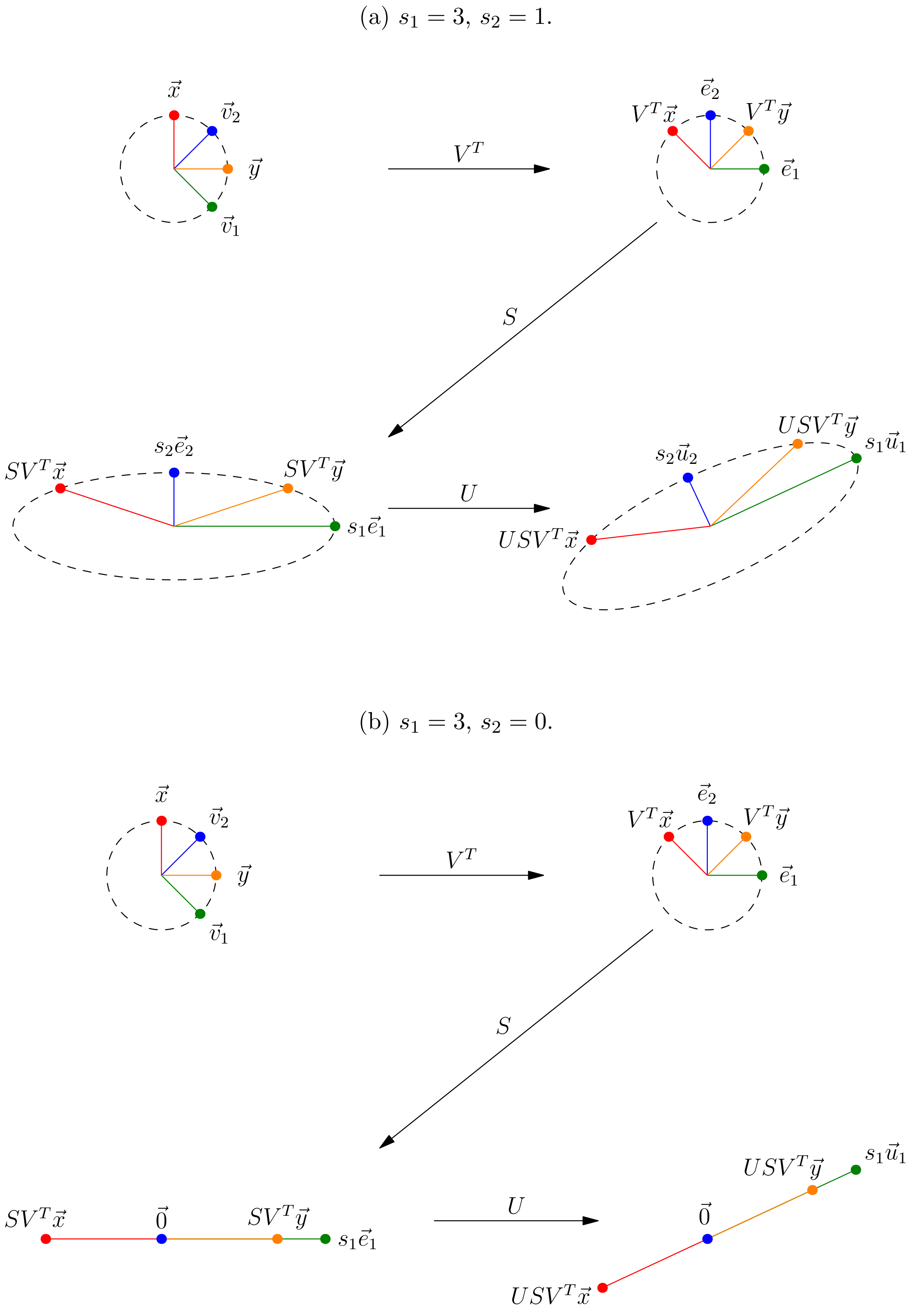
1. Rotation of w to align the component of w in the direction of the ith right singular vector v_i with the ith axis:

$$V^T w = \sum_{i=1}^{k} \langle v_i, w \rangle e_i, \tag{43}$$

   where e_i is the ith standard basis vector.

2. Scaling of each axis by the corresponding singular value

$$SV^T w = \sum_{i=1}^{k} s_i \langle v_i, w \rangle e_i. \tag{44}$$


(a) $s_1 = 3$, $s_2 = 1$.

(b) $s_1 = 3$, $s_2 = 0$.

Figure 3: The action of any matrix can be decomposed into three steps: rotation to align the right singular vectors to the axes, scaling by the singular values and a final rotation to align the axes with the left singular vectors. In image (b) the second singular value is zero, so the matrix projects two-dimensional vectors onto a one-dimensional subspace.

3. Rotation to align the ith axis with the ith left singular vector

$$USV^T w = \sum_{i=1}^{k} s_i \langle v_i, w \rangle u_i. \tag{45}$$

A consequence of the spectral theorem for symmetric matrices is that the maximum scaling produced by a matrix is equal to the maximum singular value. The maximum is achieved when the matrix is applied to any vector in the direction of the right singular vector v_1. If we restrict our attention to the orthogonal complement of v_1, then the maximum scaling is the second singular value, due to the orthogonality of the singular vectors. In general, the direction of maximum scaling orthogonal to the first $i-1$ left singular vectors is equal to the ith singular value and occurs in the direction of the ith singular vector.

Theorem 3.2. *For any matrix $A \in \mathbb{R}^{m \times k}$, the singular values satisfy*

$$s_1 = \max_{\left\{ \|w\|_2=1 \mid w \in \mathbb{R}^k \right\}} \|Aw\|_2, \tag{46}$$

$$s_i = \max_{\left\{ \|w\|_2=1 \mid w \in \mathbb{R}^k,\, w \perp v_1,\dots,v_{i-1} \right\}} \|Aw\|_2, \tag{47}$$

$$\tag{48}$$

and the right singular vectors satisfy

$$v_1 = \arg\max_{\left\{ \|w\|_2=1 \mid w \in \mathbb{R}^k \right\}} \|Aw\|_2, \tag{49}$$

$$v_i = \arg\max_{\left\{ \|w\|_2=1 \mid w \in \mathbb{R}^k,\, w \perp v_1,\dots,v_{i-1} \right\}} \|Aw\|_2, \qquad 2 \leq i \leq k. \tag{50}$$

Proof. If USV^T is the SVD of A, then the eigendecomposition of $A^T A$ equals $V^T S^2 V$ where S^2 is a diagonal matrix containing the square singular values in its diagonal. The result then follows by Theorem 5.3 in the lecture notes on PCA applied to the symmetric matrix $A^T A$, since for any w $||Aw||_2^2 = w^T A^T A w$. □

The SVD provides a geometric interpretation of the OLS estimator derived in Theorem 2.4. Let $X = USV^T$ be the SVD of the feature matrix, then

$$\beta_{\text{OLS}} = \left(XX^T\right)^{-1} Xy \tag{51}$$

$$= (US^2U^T)^{-1} USV^T y \tag{52}$$

$$= US^{-2}U^T USV^T y \tag{53}$$

$$= US^{-1}V^T y. \tag{54}$$

The OLS estimator is obtained by inverting the action of the feature matrix. This is achieved by computing the components of the response vector in the direction of the right singular vectors, scaling by the inverse of the corresponding singular values, and then rotating so that each component is aligned with the corresponding left singular vector. The matrix $\left(XX^T\right)^{-1} X$ is called a left inverse or pseudoinverse of X^T because $\left(XX^T\right)^{-1} XX^T = I$.

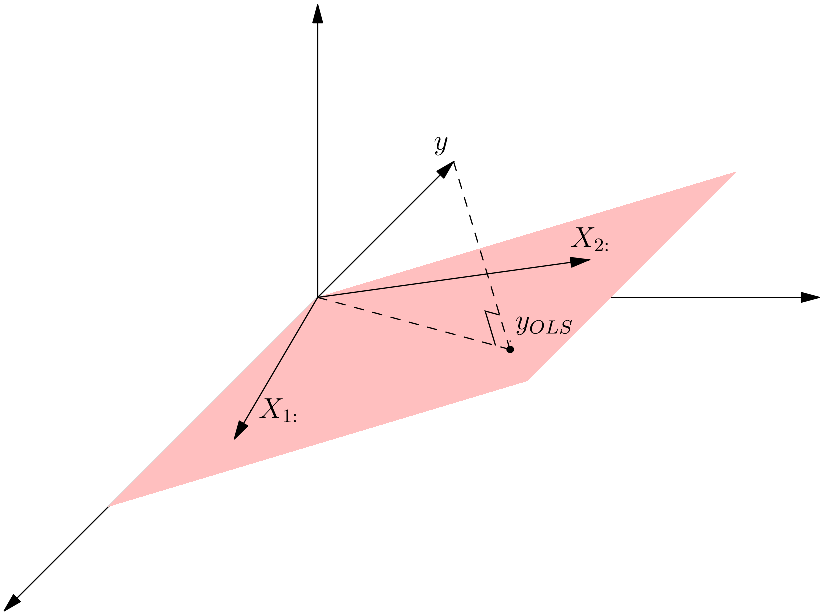


Figure 4: Illustration of Lemma 3.3 for a problem with two features corresponding to the two rows of the feature matrix $X_{1:}$ and $X_{2:}$. The least-squares solution is the orthogonal projection of the data onto the subspace spanned by these vectors.

The OLS estimator can be derived from a purely geometric viewpoint. The goal is to approximate the response vector y by a linear combination of the corresponding features. Each feature is represented by a row of X. The linear coefficients weight these rows. Equivalently, we want to find the vector in the row space of the feature matrix X that is closest to y. By definition, that vector is the orthogonal projection of y onto row(X). Figure 4 illustrates this geometric perspective. The following lemma provides a formal proof.

Lemma 3.3. *Let $X \in \mathbb{R}^{p \times n}$ be full-rank feature matrix, where $n \geq p$, and let $y \in \mathbb{R}^n$ be a response vector. The OLS estimate $X^T \beta_{\text{OLS}}$ of y given X, where*

$$\beta_{\text{OLS}} := \arg \min_{\beta \in \mathbb{R}^p} \left|\left| y - X^T \beta \right|\right|_2 , \tag{55}$$

is equal to the orthogonal projection of y onto the row space of X.

Proof. Let USV^T be the SVD of X. By Eq. (54)

$$\begin{aligned} X^T \beta_{\text{OLS}} &= X^T U S^{-1} V^T y && (56)\\ &= V S U^T U S^{-1} V^T y && (57)\\ &= V V^T y. && (58) \end{aligned}$$

Since the rows of V form an orthonormal basis for the row space of X the proof is complete. □

4 Analysis of ordinary least squares for an additive model

In this section we analyze the OLS estimator for a regression problem when the data are indeed generated by a linear model, perturbed by an additive term that accounts for model inaccuracy and noisy fluctuations. The model is parametrized by a vector of *true* linear coefficients $\beta_{\text{true}} \in \mathbb{R}^p$. We first consider a probabilistic perspective where the features are modeled by a p-dimensional random vector $\tilde{x}$, the noise by a scalar random variable $\tilde{z}$, and the response equals

$$\tilde{y} = \tilde{x}^T \beta_{\text{true}} + \tilde{z}. \tag{59}$$

If the noise $\tilde{z}$ and the feature vector are independent, then the MSE achieved by the MMSE estimator equals the variance of the noise.

Theorem 4.1 (MSE for additive model)**.** *Let $\tilde{x}$ and $\tilde{z}$ in Eq. (59) be zero mean and independent. Then the MSE achieved of the MMSE estimator of $\tilde{y}$ given $\tilde{x}$ is equal to the variance of $\tilde{z}$.*

Proof. By independence of $\tilde{x}$ and $\tilde{z}$, and linearity of expectation we have

$$\begin{aligned}
\operatorname{Var}(\tilde{y}) &= \operatorname{Var}(\tilde{x}^T \beta_{\text{true}} + \tilde{z}) && (60)\\
&= \beta_{\text{true}}^T \operatorname{E}\left(\tilde{x}\tilde{x}^T\right) \beta_{\text{true}} + \operatorname{Var}\left(\tilde{z}\right) && (61)\\
&= \beta_{\text{true}}^T \Sigma_{\tilde{x}} \beta_{\text{true}} + \operatorname{Var}\left(\tilde{z}\right), && (62)\\
\Sigma_{\tilde{x}\tilde{y}} &= \operatorname{E}\left(\tilde{x}(\tilde{x}^T \beta_{\text{true}} + \tilde{z})\right) && (63)\\
&= \Sigma_{\tilde{x}} \beta_{\text{true}} && (64)
\end{aligned}$$

By Theorem 2.1,

$$\begin{aligned}
\text{MSE} &= \operatorname{Var}(\tilde{y}) - \Sigma_{\tilde{x}\tilde{y}}^T \Sigma_{\tilde{x}}^{-1} \Sigma_{\tilde{x}\tilde{y}} && (65)\\
&= \beta_{\text{true}}^T \Sigma_{\tilde{x}} \beta_{\text{true}} + \operatorname{Var}(\tilde{z}) - \beta_{\text{true}}^T \Sigma_{\tilde{x}} \Sigma_{\tilde{x}}^{-1} \Sigma_{\tilde{x}} \beta_{\text{true}} && (66)\\
&= \operatorname{Var}(\tilde{z}). && (67)
\end{aligned}$$

□

Achieving an error equal to the variance of the noise is the best case scenario, because we cannot possibly estimate the noise from the features if they are independent. However, the result assumes that we have access to the true joint statistics of the response and the features, which are not available in practice. Instead, the coefficient vector is computed using a finite training set of examples, and the goal is to use the coefficients to predict the response on new data. In order to analyze this more realistic setting, we assume that the available data are equal to the n-dimensional vector

$$\tilde{y}_{\text{train}} := X^T \beta_{\text{true}} + \tilde{z}_{\text{train}}, \tag{68}$$

where $X \in \mathbb{R}^{p \times n}$ contains n p-dimensional feature vectors and the noise $\tilde{z}_{\text{train}}$ is modeled as an n-dimensional iid Gaussian vector with zero mean and variance σ^2. In contrast to the model the feature matrix X is fixed and deterministic. OLS is equivalent to maximum-likelihood estimation under this model.

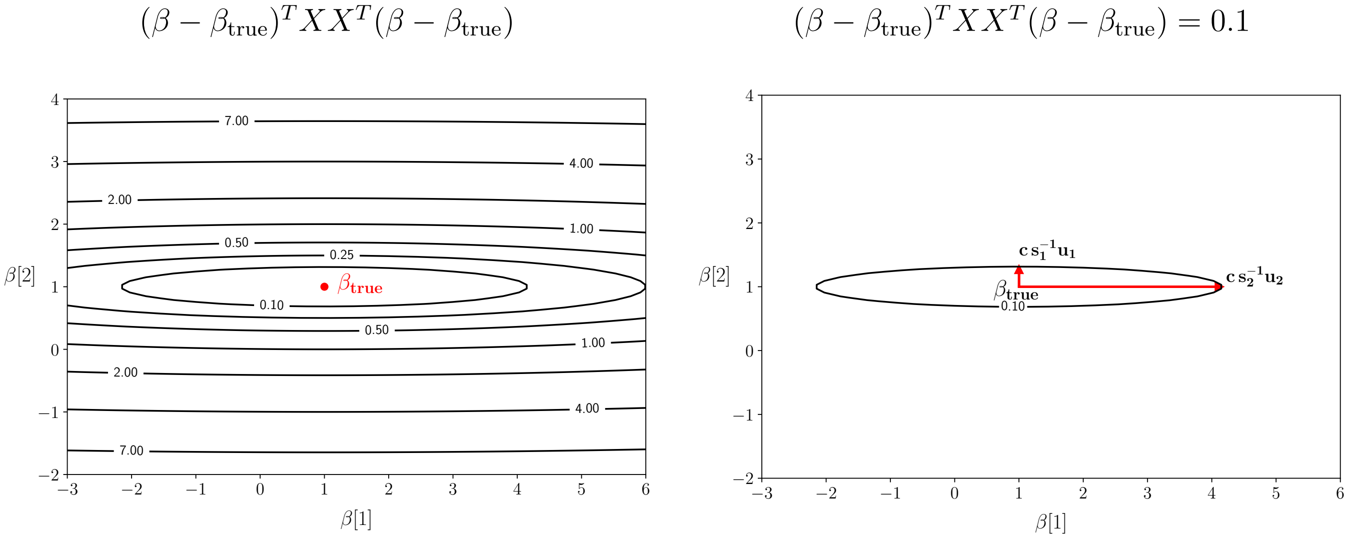


Figure 5: Deterministic quadratic component of the least-squares cost function (see Eq. (74)) for an example with two features where the left singular vectors of X align with the horizontal and vertical axes, and the singular values equal 1 and 0.1. The quadratic form is an ellipsoid centered at β_{true} with axes aligned with the left singular vectors. The curvature of the quadratic is proportional to the square of the singular values.

Lemma 4.2. *If the training data are interpreted as a realization of the random vector in Eq. (68) the maximum-likelihood estimate of the coefficients is equal to the OLS estimate.*

Proof. The likelihood is the probability density function of $\tilde{y}_{\text{train}}$ evaluated at the observed data y_{train} and interpreted as a function of the coefficient vector β,

$$\mathcal{L}_{y_{\text{train}}}(\beta) = \frac{1}{\sqrt{(2\pi\sigma^2)^n}} \exp\left(-\frac{1}{2\sigma^2} \left|\left|y_{\text{train}} - X^T \beta\right|\right|_2^2\right). \tag{69}$$

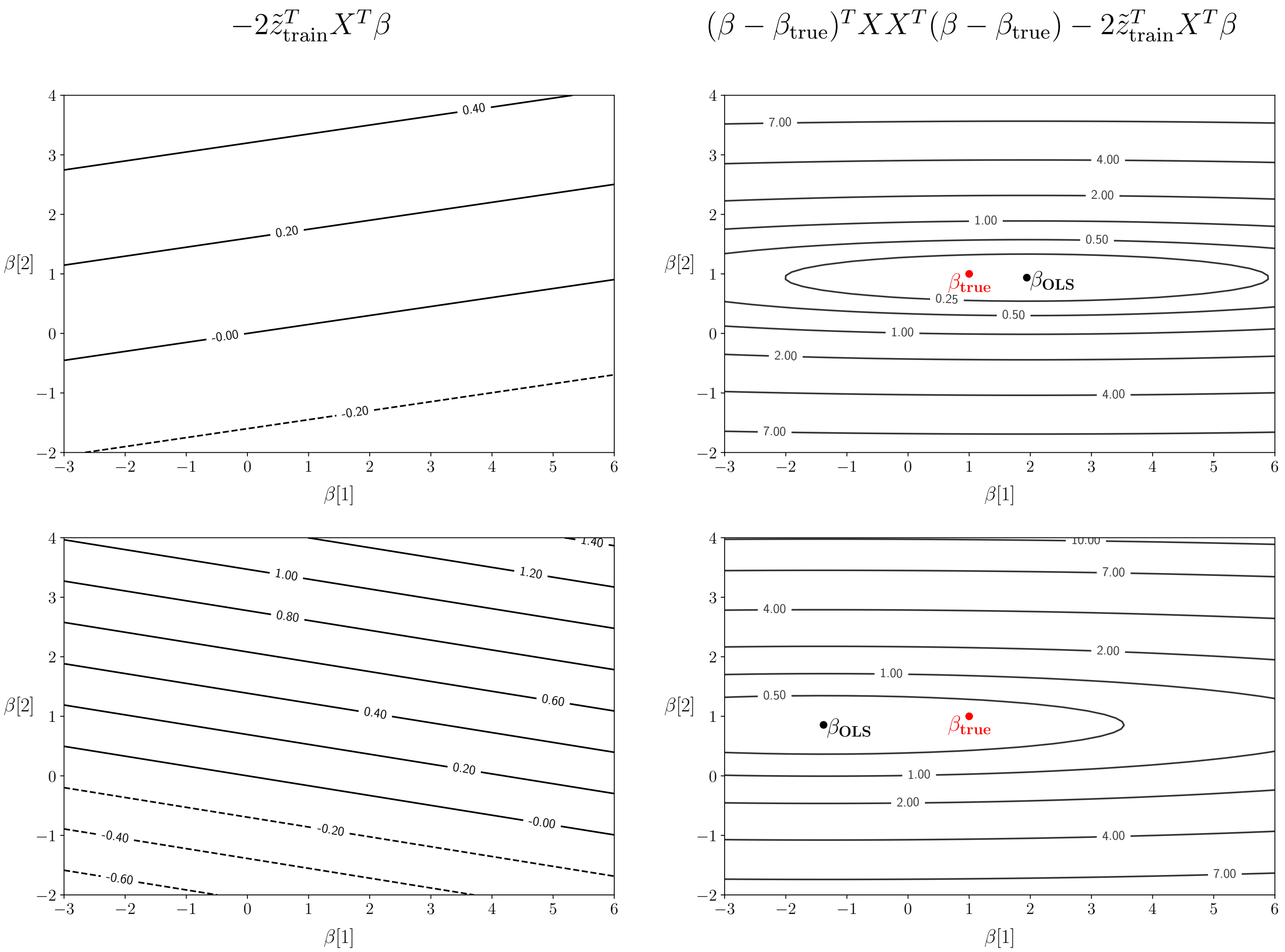
The maximum-likelihood estimator equals

$$\begin{aligned} \beta_{\text{ML}} &= \arg\max_{\beta} \mathcal{L}_{y_{\text{train}}}(\beta) && (70) \\ &= \arg\max_{\beta} \log \mathcal{L}_{y_{\text{train}}}(\beta) && (71) \\ &= \arg\min_{\beta} \left|\left|y_{\text{train}} - X^T \beta\right|\right|_2^2. && (72) \end{aligned}$$

□

The OLS cost function can be decomposed into a deterministic quadratic form centered at β_{true} and a random linear function that depends on the noise,

$$\begin{aligned} \arg\min_{\beta} \|\tilde{y}_{\text{train}} - X^T\beta\|_2^2 &= \arg\min_{\beta} \|\tilde{z}_{\text{train}} - X^T(\beta - \beta_{\text{true}})\|_2^2 && (73) \\ &= \arg\min_{\beta} (\beta - \beta_{\text{true}})^T X X^T (\beta - \beta_{\text{true}}) - 2\tilde{z}_{\text{train}}^T X^T (\beta - \beta_{\text{true}}) + \tilde{z}_{\text{train}}^T \tilde{z}_{\text{train}} \\ &= \arg\min_{\beta} (\beta - \beta_{\text{true}})^T X X^T (\beta - \beta_{\text{true}}) - 2\tilde{z}_{\text{train}}^T X^T \beta. && (74) \end{aligned}$$



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

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

Figure 6: The left column show two realizations of the random linear component of the least-squares cost function (see Eq. (74)) for the example in Figure 5. The right column shows the corresponding cost function, which is a quadratic centered at a point that does not coincide with β_{true} due to the linear term. The minimum of the quadratic is denoted by β_{OLS}.

Figure 5 shows the quadratic component for a simple example with two features. Let $X = USV^T$ be the SVD of the feature matrix. The contour lines of the quadratic form are ellipsoids, defined by the equation

$$\begin{aligned}(\beta - \beta_{\text{true}})^T X X^T (\beta - \beta_{\text{true}}) &= (\beta - \beta_{\text{true}})^T U S^2 U^T (\beta - \beta_{\text{true}}) && (75)\\ &= \sum_{i=1}^{p} s_i^2 (u_i^T (\beta - \beta_{\text{true}}))^2 = c^2 && (76)\end{aligned}$$

for a constant c. The axis of the ellipsoid are the left singular vectors of X. The curvature in those directions is proportional to the square of the singular values, as shown in Figure 5. Due to the random linear component, the minimum of the least-squares cost function is not at β_{true}. Figure 5 shows this for a simple example. The following theorem shows that the minimum of the cost function is a Gaussian random vector centered at β_{true}.

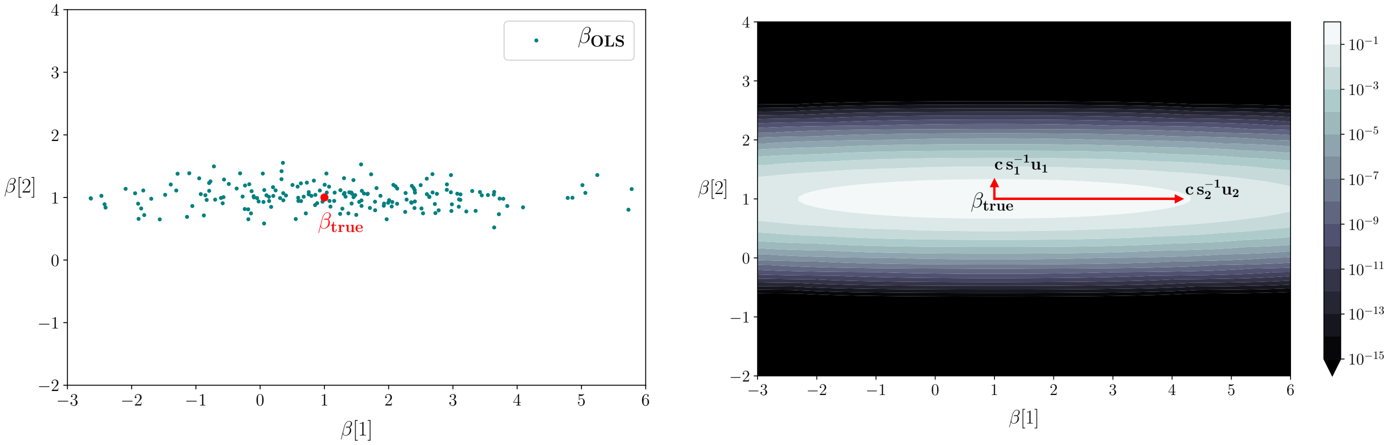


Figure 7: The left image is a scatterplot of OLS estimates corresponding to different noise realizations for the example in Figure 6. The right image is a heatmap of the distribution of the OLS estimate, which is centered at β_{true} and has covariance matrix $\sigma^2 US^{-2}U^T$, as established in Theorem 4.3.

Theorem 4.3. *If the training data follow the additive model in Eq. (68) and X is full rank, the OLS coefficient*

$$\tilde{\beta}_{\text{OLS}} := \arg\min_{\beta} \left|\left|\tilde{y}_{\text{train}} - X^T\beta\right|\right|_2, \tag{77}$$

is a Gaussian random vector with mean β_{true} and covariance matrix $\sigma^2 US^{-2}U^T$, where $X = USV^T$ is the SVD of the feature matrix.

Proof. We have

$$\begin{aligned}
\beta_{\text{OLS}} &= (XX^T)^{-1}X\tilde{y}_{\text{train}} &(78)\\
&= (XX^T)^{-1}XX^T\beta_{\text{true}} + (XX^T)^{-1}X\tilde{z}_{\text{train}} &(79)\\
&= \beta_{\text{true}} + (XX^T)^{-1}X\tilde{z}_{\text{train}} &(80)\\
&= \beta_{\text{true}} + US^{-1}V^T\tilde{z}_{\text{train}}. &(81)
\end{aligned}$$

The result then follows from Theorem 8.6 in the PCA lecture notes. □

Figure 7 shows a scatterplot of OLS estimators corresponding to different noise realizations, as well as the distribution of the OLS estimator. The contour lines of the distribution are ellipsoidal with axes aligned with the left singular vectors of the feature matrix. The variance along those axes is proportional to the inverse of the squared singular values. If there are singular values that are very small, the variance in the direction of the corresponding singular vector can be very large, as is the case along the horizontal axis of Figure 7.

In practice, we cannot verify the coefficient error for most datasets, because there is no *true* underlying linear model. Instead, the models are evaluated in terms of feature prediction. In the next two sections we analyze the prediction error of the OLS estimator on training and test data.

4.1 Training error

The training error achieved by OLS has an intuitive geometric interpretation: it is the projection of the noise vector onto the subspace spanned by the feature vectors.

Lemma 4.4. *If the training data follow the additive model in Eq. (68) and X is full rank, the training error of the OLS estimate $X^T \tilde{\beta}_{\text{OLS}}$ is the projection of the noise onto the orthogonal complement of the row space of X.*

Proof. By Lemma 3.3

$$\begin{aligned}
\tilde{y}_{\text{train}} - X^T \tilde{\beta}_{\text{OLS}} &= \tilde{y}_{\text{train}} - \mathcal{P}_{\text{row}(X)}\, \tilde{y}_{\text{train}} && (82)\\
&= X^T \beta_{\text{true}} + \tilde{z}_{\text{train}} - \mathcal{P}_{\text{row}(X)} \left(X^T \beta_{\text{true}} + \tilde{z}_{\text{train}}\right) && (83)\\
&= X^T \beta_{\text{true}} + \tilde{z}_{\text{train}} - X^T \beta_{\text{true}} - \mathcal{P}_{\text{row}(X)}\, \tilde{z}_{\text{train}} && (84)\\
&= \mathcal{P}_{\text{row}(X)^\perp}\, \tilde{z}_{\text{train}}. && (85)
\end{aligned}$$

□

We define the average training square error as

$$\widetilde{E}^2_{\text{train}} := \frac{1}{n} \left|\left|\tilde{y}_{\text{train}} - X^T \tilde{\beta}_{\text{OLS}}\right|\right|_2^2. \tag{86}$$

This quantity captures the average error incurred by the OLS estimate on the training data. By Lemma 4.4, if the noise is Gaussian, then the training error is the projection of an n-dimensional iid Gaussian random vector onto the subspace orthogonal to the span of the feature vectors. The iid assumption means that the Gaussian distribution is isotropic. The dimension of this subspace equals $n - p$, so the fraction of the variance in the Gaussian vector that lands on it should be approximately equal to $1 - p/n$. The following theorem establishes that this is indeed the case.

Theorem 4.5. *If the training data follow the additive model in Eq. (68) and X is full rank, then the mean of the average training error defined in Eq. (86) equals*

$$\mathrm{E}\left(\widetilde{E}^2_{\text{train}}\right) = \sigma^2 \left(1 - \frac{p}{n}\right), \tag{87}$$

and its variance equals

$$\mathrm{Var}(\widetilde{E}^2_{\text{train}}) = \frac{2\sigma^4 (n-p)}{n^2}. \tag{88}$$

Proof. By Lemma 4.4

$$\begin{aligned}
n\widetilde{E}^2_{\text{train}} &= \left|\left|\mathcal{P}_{\text{row}(X)^\perp}\, \tilde{z}_{\text{train}}\right|\right|_2^2 && (89)\\
&= \tilde{z}_{\text{train}}^T V_\perp V_\perp^T V_\perp V_\perp^T \tilde{z}_{\text{train}} && (90)\\
&= \left|\left|V_\perp^T \tilde{z}_{\text{train}}\right|\right|_2^2, && (91)
\end{aligned}$$

where the columns of $V_\perp$ are an orthonormal basis for $\text{row}(X)^\perp$. By Theorem 8.6 in the notes on PCA $V_\perp^T \tilde{z}_{\text{train}}$ is a Gaussian vector of dimension $n-p$ with covariance matrix

$$\begin{aligned}
\Sigma_{V_\perp^T \tilde{z}_{\text{train}}} &= V_\perp^T \Sigma_{\tilde{z}_{\text{train}}} V_\perp && (92)\\
&= V_\perp^T \sigma^2 I V_\perp && (93)\\
&= \sigma^2 I. && (94)
\end{aligned}$$

The error is therefore equal to the square ℓ_2 norm of an iid Gaussian random vector. Let $\tilde{w}$ be a d-dimensional zero-mean Gaussian random vector $\tilde{w}$ with unit variance. The expected value of its square ℓ_2 norm is

$$\begin{aligned}
\mathrm{E}\left(\|\tilde{w}\|_2^2\right) &= \mathrm{E}\left(\sum_{i=1}^d \tilde{w}[i]^2\right) && (95)\\
&= \sum_{i=1}^d \mathrm{E}\left(\tilde{w}[i]^2\right) && (96)\\
&= d. && (97)
\end{aligned}$$

The mean square equals

$$\begin{aligned}
\mathrm{E}\left[\left(\|\tilde{w}\|_2^2\right)^2\right] &= \mathrm{E}\left[\left(\sum_{i=1}^d \tilde{w}[i]^2\right)^2\right] && (98)\\
&= \sum_{i=1}^d \sum_{j=1}^d \mathrm{E}\left(\tilde{w}[i]^2 \tilde{w}[j]^2\right) && (99)\\
&= \sum_{i=1}^d \mathrm{E}\left(\tilde{w}[i]^4\right) + 2\sum_{i=1}^{d-1} \sum_{j=i+1}^{d} \mathrm{E}\left(\tilde{w}[i]^2\right) \mathrm{E}\left(\tilde{w}[j]^2\right) && (100)\\
&= 3d + d(d-1) \quad \text{(the 4th moment of a standard Gaussian equals 3)} && (101)\\
&= d(d+2), && (102)
\end{aligned}$$

so the variance equals

$$\begin{aligned}
\mathrm{Var}\left(\|\tilde{w}\|_2^2\right) &= \mathrm{E}\left[\left(\|\tilde{w}\|_2^2\right)^2\right] - \mathrm{E}^2\left(\|\tilde{w}\|_2^2\right) && (103)\\
&= 2d. && (104)
\end{aligned}$$

As d grows, the relative deviation of the squared norm of the Gaussian vector from its mean decreases proportionally to $\sqrt{2/d}$, as shown in Figure 8. Geometrically, the probability density concentrates close to the surface of a sphere with radius $\sqrt{d}$. By definition of the training error, we have

$$\begin{aligned}
\widetilde{E}_{\text{train}}^2 &= \frac{1}{n} \left\| V_\perp^T \tilde{z}_{\text{train}} \right\|_2^2 && (105)\\
&= \frac{\sigma^2}{n} \|\tilde{w}\|_2^2, && (106)
\end{aligned}$$

so the result follows from setting $d := n - p$ in Eqs. (97) and (104). □

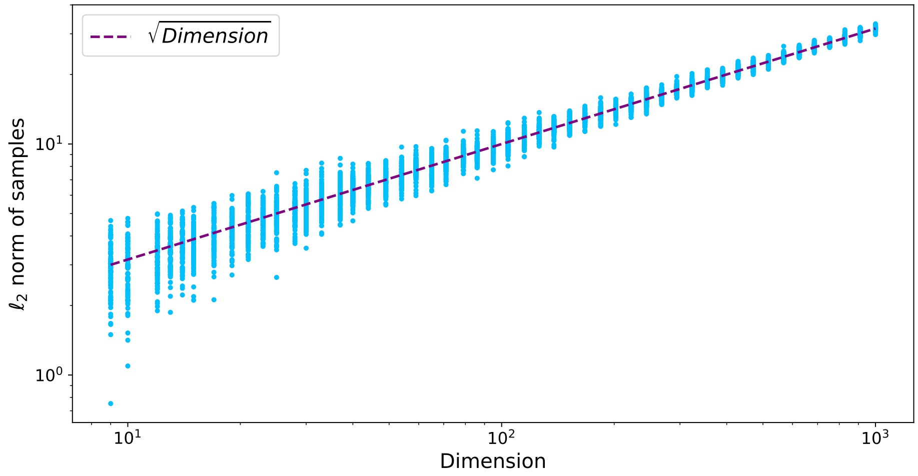


Figure 8: The graphs shows the ℓ_2 norm of 100 independent samples from standard Gaussian random vectors in different dimensions. The norms of the samples concentrate around the square root of the dimension.

The variance of the square error scales with $1/n$, which implies that the error concentrates around its mean with high probability as the number of examples in the training set grows. This can be made precise using Chebyshev's inequality, which is a direct consequence of Markov's inequality.

Lemma 4.6 (Chebyshev's inequality)**.** *Let $\tilde{r}$ be a random variable with finite variance. For any positive constant $c > 0$,*

$$P((\tilde{a} - \mathrm{E}(\tilde{a}))^2 \geq c) \leq \frac{\mathrm{Var}(\tilde{a})}{c}. \tag{107}$$

Proof. Apply Markov's inequality (Theorem in the notes on PCA) to the random variable $(\tilde{a} - \mathrm{E}(\tilde{a}))^2$. □

Corollary 4.7. *If the training data follow the additive model in Eq.* (68) *and X is full rank, then for any $\epsilon > 0$ we have*

$$P\left(\left(\widetilde{E}^2_{\text{train}} - \sigma^2\left(1 - \frac{p}{n}\right)\right) > \epsilon\right) < \frac{2\sigma^4}{n\epsilon^2}. \tag{108}$$

Proof. By Theorem 4.5 $\mathrm{Var}(\widetilde{E}^2_{\text{train}}) \leq \frac{2\sigma^4}{n}$. □

The bound in this corollary can be improved significantly by using sharper concentration bounds that exploit higher-order moments. In any case, the important conclusion is that the training error concentrates around $\sigma^2\left(1 - \frac{p}{n}\right)$. For large n, the error equals the variance of the noisy component σ^2, which is the error achieved by the true coefficients β_{true}. When $n \approx p$, however, the error can be much smaller. This is bad news. If an estimator achieves an error of less than σ it must be *overfitting* the training noise, which will result in a higher generalization error on held-out data. This suggests that the OLS estimator overfits the training data when the number of examples is small with respect to the number of features. Figure 9 shows that this is indeed the case for the dataset in Example 2.5. In fact, the training error is proportional to $\left(1 - \frac{p}{n}\right)$ as predicted by our theoretical result.

4.2 Test error

The test error of an estimator quantifies its performance on held-out data, which have not been used to fit the model. We model the test data as

$$\tilde{y}_{\text{test}} := \tilde{x}_{\text{test}}^T \beta_{\text{true}} + \tilde{z}_{\text{test}}. \tag{109}$$

The linear coefficients are the same as in the training set, but the features and noise are different. The features are modeled as a p-dimensional random vector $\tilde{x}_{\text{test}}$ with zero mean (the features are assumed to be centered) and the noise $\tilde{z}_{\text{test}}$ is a zero-mean Gaussian random variable with the same variance σ^2 as the training noise. The training and test noise are assumed to be independent from each other and from the features. Our goal is to characterize the test error

$$\begin{aligned} \widetilde{E}_{\text{test}} &:= \tilde{y}_{\text{test}} - \tilde{x}_{\text{test}}^T \tilde{\beta}_{\text{OLS}} & (110) \\ &= \tilde{z}_{\text{test}} + \tilde{x}_{\text{test}}^T \left(\beta_{\text{true}} - \tilde{\beta}_{\text{OLS}} \right), & (111) \end{aligned}$$

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

Theorem 4.8 (Test mean square error)**.** *If the training data follow the additive model in Eq. (68), X is full rank, and the test data follow the model in Eq. (109), then the mean square of the test error equals*

$$\mathrm{E}(\widetilde{E}_{\text{test}}^2) = \sigma^2 \left(1 + \sum_{i=1}^{p} \frac{\mathrm{Var}(u_i^T \tilde{x}_{\text{test}})}{s_i^2} \right), \tag{112}$$

where $\Sigma_{\tilde{x}_{\text{test}}}$ is the covariance matrix of the feature vector, $s_1, \ldots, s_p$ are the singular values of X and $v_1, \ldots, v_p$ are the right singular vectors.

Proof. By assumption, the two components of the test error in Eq. (111) are independent, so the variance of their sum is the sum of their variances:

$$\mathrm{Var}\left(\tilde{y}_{\text{test}} - \tilde{x}_{\text{test}}^T \tilde{\beta}_{\text{OLS}} \right) = \sigma^2 + \mathrm{Var}\left(\tilde{x}_{\text{test}}^T \left(\beta_{\text{true}} - \tilde{\beta}_{\text{OLS}} \right) \right) \tag{113}$$

Since everything is zero mean, this also holds for the mean square. Let USV^T be the SVD of X. The coefficient error equals

$$\beta_{\text{OLS}} - \beta_{\text{true}} = \sum_{i=1}^{p} \frac{v_i^T \tilde{z}_{\text{train}}}{s_i} u_i, \tag{114}$$

by Theorem 4.3. This implies

$$\begin{aligned} \mathrm{E}\left[\left(\tilde{x}_{\text{test}}^T \left(\beta_{\text{true}} - \tilde{\beta}_{\text{OLS}} \right) \right)^2 \right] &= \mathrm{E}\left[\left(\sum_{i=1}^{p} \frac{v_i^T \tilde{z}_{\text{train}} \, u_i^T \tilde{x}_{\text{test}}}{s_i} \right)^2 \right] & (115) \\ &= \sum_{i=1}^{p} \frac{\mathrm{E}\left[(v_i^T \tilde{z}_{\text{train}})^2 \right] \mathrm{E}\left[(u_i^T \tilde{x}_{\text{test}})^2 \right]}{s_i^2}, & (116) \end{aligned}$$

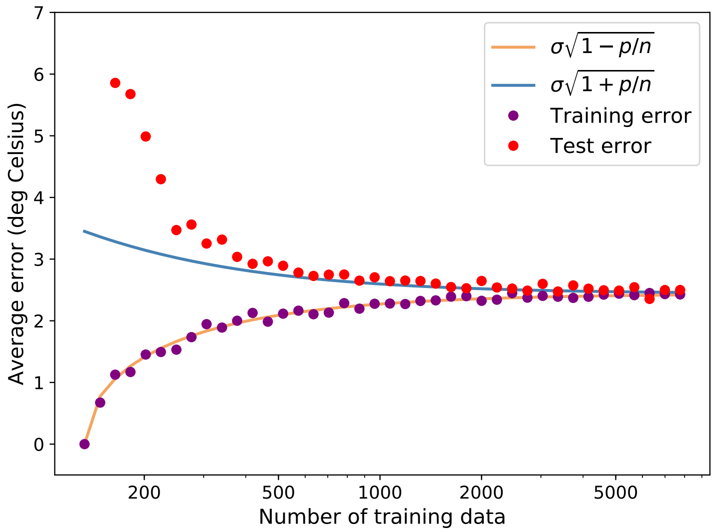


Figure 9: Comparison of the theoretical approximation for the training and test error of the OLS with the actual errors on the temperature data described in Example 2.5. The parameter σ is fixed based on the asymptotic value of the error.

where the second equality holds because when we expand the square, the cross terms cancel due to the independence assumptions and linearity of expectation. For $i \neq j$

$$\begin{aligned}
\operatorname{E}\left(\frac{v_i^T \tilde{z}_{\text{train}}\, u_i^T \tilde{x}_{\text{test}}}{s_i}\frac{v_j^T \tilde{z}_{\text{train}}\, u_j^T \tilde{x}_{\text{test}}}{s_j}\right) &= \frac{\operatorname{E}\left(u_i^T \tilde{x}_{\text{test}} u_j^T \tilde{x}_{\text{test}}\right)}{s_i s_j} v_i^T \operatorname{E}\left(\tilde{z}_{\text{train}} \tilde{z}_{\text{train}}^T\right) v_j && (117)\\
&= \frac{\operatorname{E}\left(u_i^T \tilde{x}_{\text{test}} u_j^T \tilde{x}_{\text{test}}\right)}{s_i s_j} v_i^T v_j && (118)\\
&= 0. && (119)
\end{aligned}$$

By linearity of expectation, we conclude

$$\begin{aligned}
\operatorname{E}\left[\left(\tilde{x}_{\text{test}}^T\left(\beta_{\text{true}} - \tilde{\beta}_{\text{OLS}}\right)\right)^2\right] &= \sum_{i=1}^{p} \frac{v_i^T \operatorname{E}(\tilde{z}_{\text{train}} \tilde{z}_{\text{train}}^T) v_i u_i^T \operatorname{E}(\tilde{x}_{\text{test}} \tilde{x}_{\text{test}}^T) u_i}{s_i^2} && (120)\\
&= \sigma^2 \sum_{i=1}^{p} \frac{u_i^T \Sigma_{\tilde{x}_{\text{test}}} u_i}{s_i^2}, && (121)
\end{aligned}$$

because the covariance matrix of the training noise $\tilde{z}_{\text{train}}$ equals $\sigma^2 I$. □

The square of the ith singular value of the training covariance matrix is proportional to the sample variance of the training data in the direction of the ith singular value u_i,

$$\begin{aligned}
\frac{s_i^2}{n} &= \frac{u_i X X^T u_i}{n} && (122)\\
&= u_i^T \Sigma_{\mathcal{X}} u_i && (123)\\
&= \operatorname{var}\left(\mathcal{P}_{u_i} \mathcal{X}\right). && (124)
\end{aligned}$$

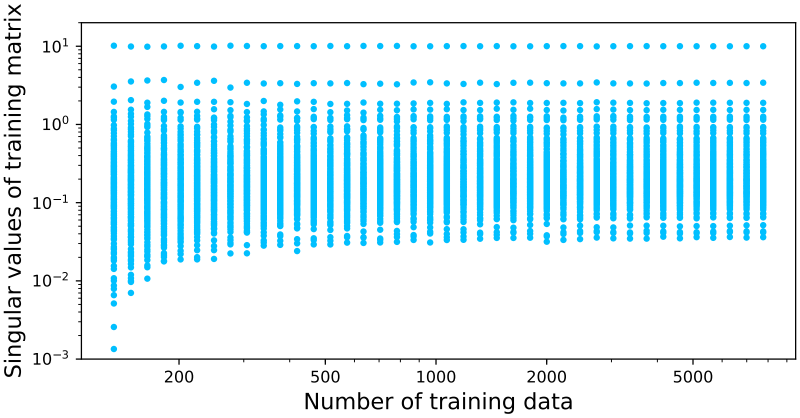


Figure 10: Singular values of the training matrix in Example 2.5 for different numbers of training data.

If this sample variance is a good approximation to the variance of the test data in that direction then

$$\mathrm{E}(\widetilde{E}_{\text{test}}^2) \approx \sigma^2 \left(1 + \frac{p}{n}\right). \tag{125}$$

However, if the training data is not large enough, the sample covariance matrix may not provide a good estimate of the feature variance in every direction. In that case, there may be terms in the test error where s_i is very small, due to correlations between the features, but the true directional variance is not. Figure 10 shows that some of the singular values of the training matrix in the temperature prediction are indeed minuscule. Unless the test variance in that direction cancel them out, this results in a large test error.

Intuitively, estimating the contribution of low-variance components of the feature vector to the linear coefficients requires amplifying them. This also amplifies the training noise in those directions. When estimating the response, this amplification is neutralized by the corresponding small directional variance of the test features as long as it occurs in the right directions (which is the case if the sample covariance matrix is a good approximation to the test covariance matrix). Otherwise, it will result in a high response error. This typically occurs when the number of training data is small with respect to the number of features. The effect is apparent in Figure 2 for small values of n. In the next sections, we describe techniques to alleviate this issue.

5 Ridge regression

As we saw in the previous section, the least-squares estimator can suffer from significant noise amplification when the number of training data are small. This results in coefficients with very large amplitudes, which overfit the noise in the training set, as illustrated by the left image in Figure 21. A popular approach to avoid this problem is to add an extra term to the least-squares cost function, which penalizes the norm of the coefficient vector . The goal is to promote solutions that yield a good fit to the data using linear coefficients that are not too large. Modifying cost functions to favor structured solutions is called *regularization*. Least-squares regression combined with

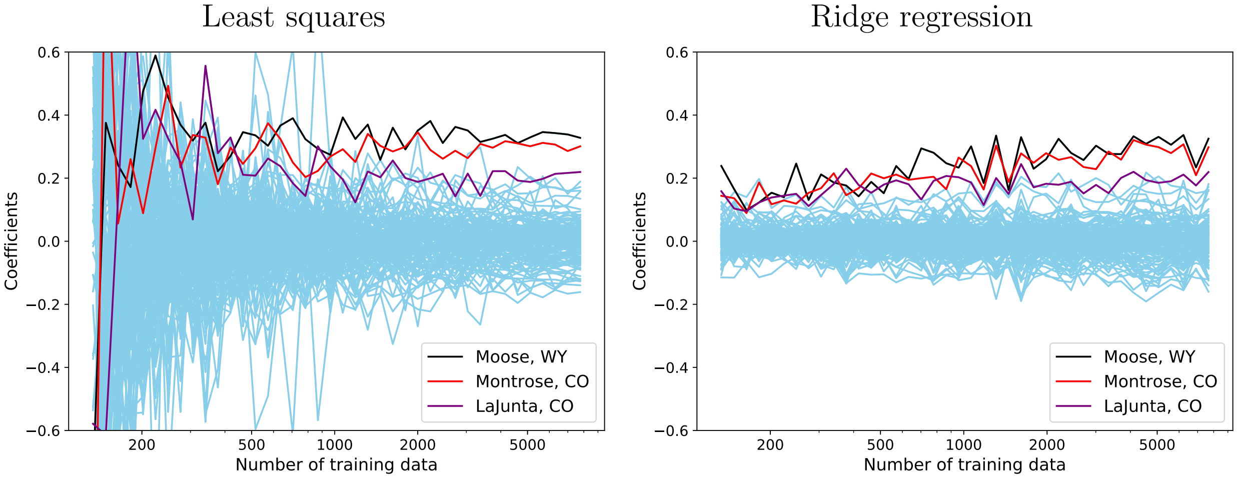


Figure 11: Coefficients of the least-squares (left) and ridge-regression (right) estimators computed from the data described in Example 2.5 for different values of training data. All coefficients are depicted in light blue except the three that have the largest magnitudes for large n, which correspond to the stations of Moose in Wyoming, and Montrose and La Junta in Colorado.

ℓ_2-norm regularization is known as ridge regression in statistics and as Tikhonov regularization in the literature on inverse problems.

Definition 5.1 (Ridge regression)**.** *For any $X \in \mathbb{R}^{p\times n}$ and $y \in \mathbb{R}^n$ the ridge-regression estimator is the minimizer of the optimization problem*

$$\beta_{\text{RR}} := \arg\min_{\beta} \|y - X^T\beta\|_2^2 + \lambda\|\beta\|_2^2, \tag{126}$$

where $\lambda > 0$ is a fixed regularization parameter.

As in the case of least-squares regression, the ridge-regression estimator has a closed form solution.

Theorem 5.2 (Ridge-regression estimate)**.** *For any $X \in \mathbb{R}^{p\times n}$ and $y \in \mathbb{R}^n$ we have*

$$\beta_{\text{RR}} = \left(XX^T + \lambda I\right)^{-1} Xy. \tag{127}$$

Proof. The cost function can be reformulated to equal a modified least-squares problem

$$\beta_{\text{RR}} := \arg\min_{\beta} \left|\left| \begin{bmatrix} y \\ 0 \end{bmatrix} - \begin{bmatrix} X^T \\ \sqrt{\lambda} I \end{bmatrix} \beta \right|\right|_2^2. \tag{128}$$

By Theorem 2.4 the solution equals

$$\begin{aligned} \beta_{\text{RR}} &= \left(\begin{bmatrix} X & \sqrt{\lambda} I \end{bmatrix} \begin{bmatrix} X & \sqrt{\lambda} I \end{bmatrix}^T \right)^{-1} \begin{bmatrix} X & \sqrt{\lambda} I \end{bmatrix} \begin{bmatrix} y \\ 0 \end{bmatrix} && (129) \\ &= \left(XX^T + \lambda I\right)^{-1} Xy. && (130) \end{aligned}$$

□

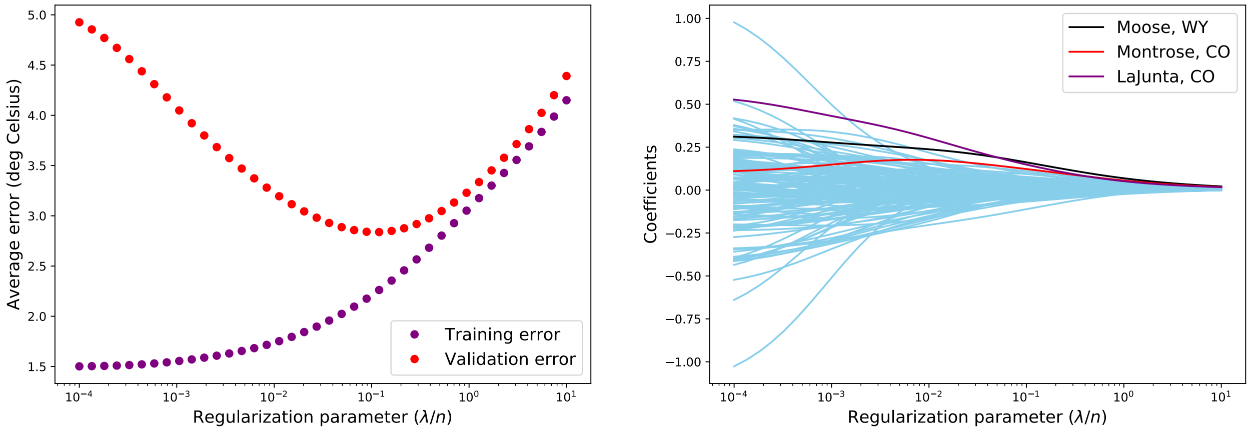


Figure 12: The left graph shows the training and validation errors of the ridge-regression estimator applied to the data described in Example 2.5 for different values of the regularization parameter λ. The number of training data is fixed to $n = 202$ training data. The right figure shows the values of the model coefficients for the different λ values. All coefficients are depicted in light blue except the three that have the largest magnitudes for large n, which correspond to the stations of Moose in Wyoming, and Montrose and La Junta in Colorado.

Notice that when $\lambda \to 0$, β_{RR} converges to the least-squares estimator. When $\lambda \to \infty$, β_{RR} converges to zero.

The regularization parameter λ governs the trade-off between the term that promotes a good model fit on the training set and the term that controls the magnitudes of the coefficients. Ideally we would like to set the value of λ that achieves the best test error. However, we do not have access to the test set when training the regression model (and even if we did, one should never use test data for anything else other than evaluating test error!). We cannot use the training data to determine λ, since $\lambda = 0$ obviously achieves the minimum error on the training data. Instead, we use *validation* data, separate from the training and test data, to evaluate the error of the model for different values of λ and select the best value. This approach for setting model hyper parameters is commonly known as cross validation.

As shown in Figure 12, in the regime where the least-squares estimator overfits the training data, the ridge-regression estimator typically also overfits for small values of λ, which is reflected in a high validation error. Increasing λ improves the validation error, up until a point where the error increases again, because the least-squares term loses too much weight with respect to the regularization term. Figure 12 also shows the coefficients of the model applied to the data described in Example 2.5 for different values of λ. When λ is small, many coefficients are large, which makes it possible to overfit the training noise through cancellations. For larger λ their magnitudes decrease, eventually becoming too small to produce an accurate fit.

Figure 13 shows that ridge regression outperforms least-squares regression on the dataset of Example 2.5 for small values of n, and has essentially the same performance for larger values, when the least-squares estimator does not overfit the training data (this is expected as the estimators are equivalent for small λ values). The figure also shows that λ values selected by cross validation

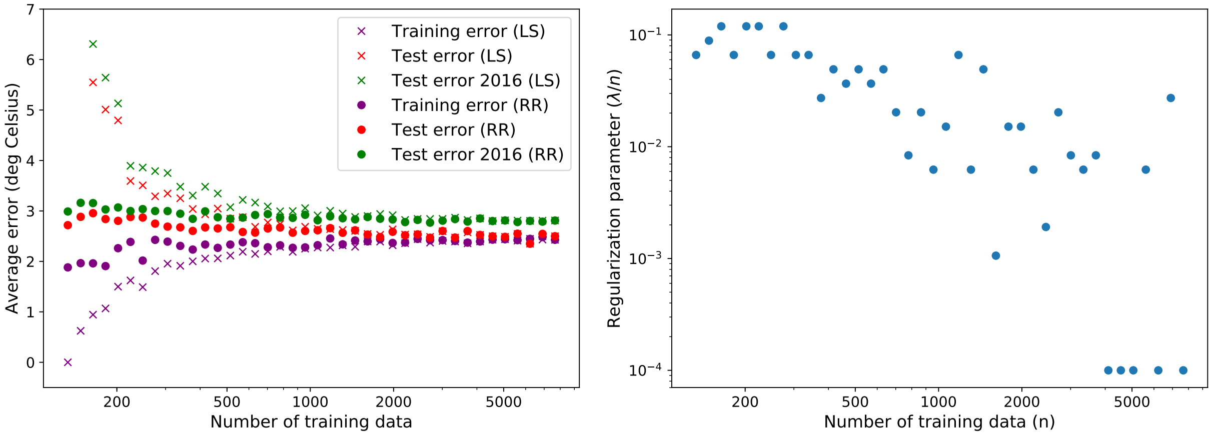


Figure 13: Performance of the ridge-regression estimator on the temperature data described in Example 2.5. The left image shows the RMSE achieved by the model on the training and test sets, and on the 2016 data, for different number of training data and compares it to the RMSE of least-squares regression. The right graph shows the values of λ selected from a validation dataset of size 100 for each number of training data.

are larger for small values of n, where regularization is more useful.

In order to analyze the ridge-regression estimator, we consider data generated by a linear model as in Eq. (68). In that case, the ridge-regression cost function can be decomposed into the sum of two deterministic quadratic forms centered at β_{true} and at the origin, and a random linear function that depends on the noise. By the same argument used to derive Eq. (74)

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

Figure 14 shows the different components for a simple example with two features. The proof of the following theorem derives the distribution of the ridge-regression coefficient estimate by analyzing these components.

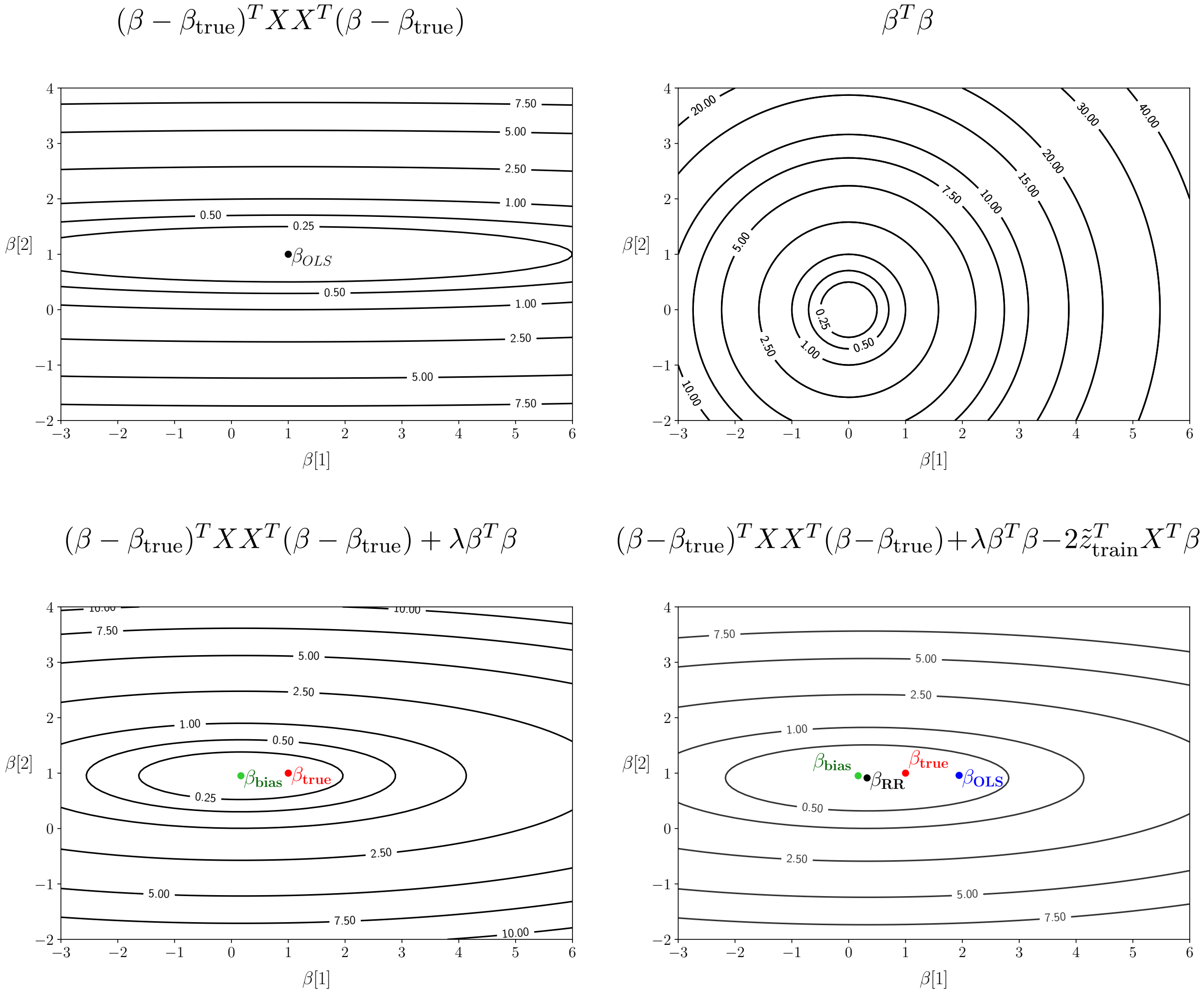
Theorem 5.3 (Ridge-regression coefficient estimate)**.** *If the training data follow the additive model in Eq.* (68)*, then the ridge regression coefficient estimate is a Gaussian random vector with mean*

$$\beta_{\text{bias}} := \sum_{j=1}^{p} \frac{s_j^2 \langle u_j, \beta_{\text{true}} \rangle}{s_j^2 + \lambda} u_j \tag{131}$$

and covariance matrix

$$\Sigma_{\text{RR}} := \sigma^2 U \operatorname{diag}_{j=1}^{p} \left(\frac{s_j^2}{(s_j^2 + \lambda)^2} \right) U^T, \tag{132}$$

where $\operatorname{diag}_{j=1}^{p}(d_i)$ *denotes a diagonal matrix with entries* $d_1, \ldots, d_p$*.*



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

$\beta^T \beta$

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

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

Figure 14: Visualization of the different components of the ridge-regression cost function for the example in Figure 5. The regularization parameter is set to $\lambda := 0.05$. The top row shows the two deterministic quadratic forms cost function: the least square component (left) and the regularization component (right). The bottom left plot shows the combination of both quadratic components. The resulting quadratic is centered at a point β_{bias}, which is the expected value of the ridge-regression coefficient estimate. Finally, the bottom right plot shows a realization of the ridge-regression cost function obtained by adding the deterministic quadratic components with the random linear component that depends on the training response. The minimum of the resulting cost function is denoted by β_{RR}. For comparison, we also include the minimum of the OLS cost function β_{OLS}.

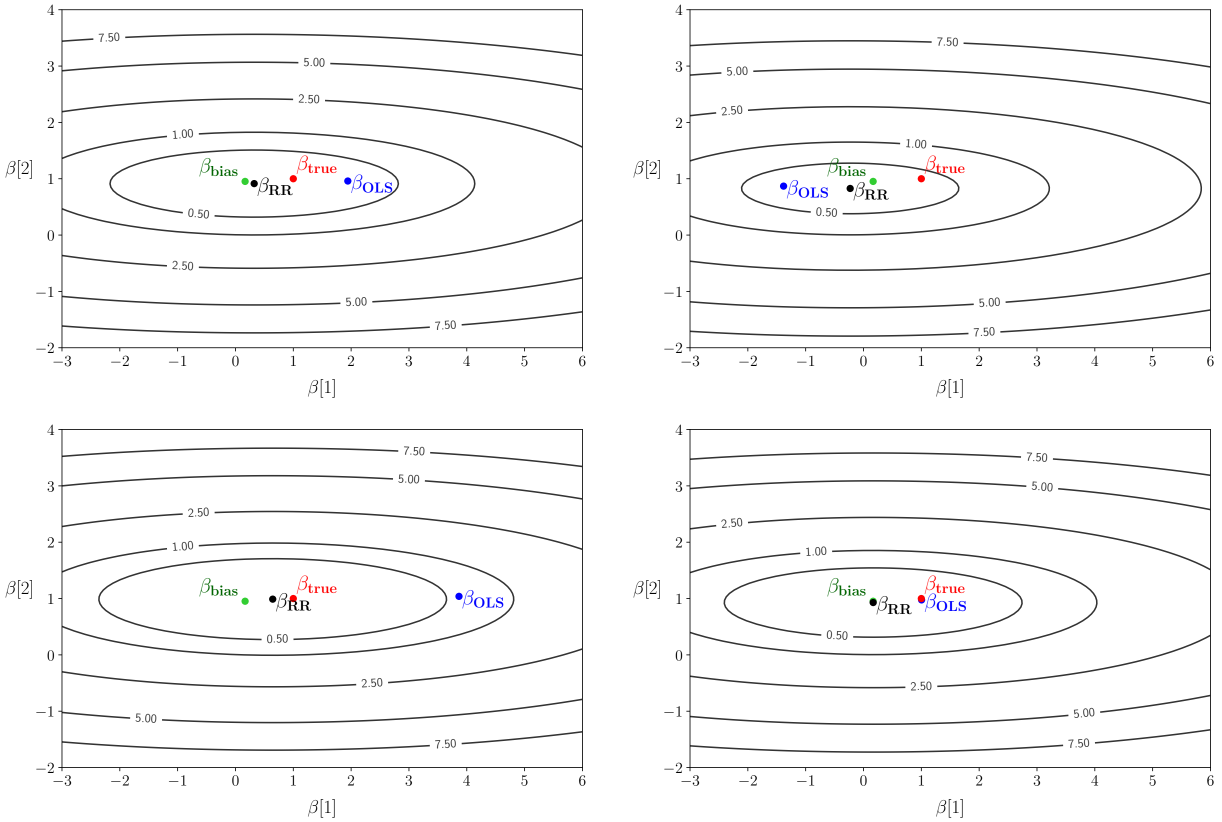


Figure 15: Different realizations of the ridge-regression cost function corresponding to different realizations of the noise (the true coefficients and the feature matrix remain the same) for the example in Figure 14. The regularization parameter is set to $\lambda := 0.05$.

Proof. By Theorem 5.2 the solution equals

$$\begin{aligned}
\tilde{\beta}_{\text{RR}} &= \left(XX^T + \lambda I\right)^{-1} X \left(X^T \beta_{\text{true}} + \tilde{z}_{\text{train}}\right) && (133)\\
&= \left(US^2U^T + \lambda UU^T\right)^{-1} \left(US^2U^T \beta_{\text{true}} + USV^T \tilde{z}_{\text{train}}\right) && (134)\\
&= \left(U(S^2 + \lambda I)U^T\right)^{-1} \left(US^2U^T \beta_{\text{true}} + USV^T \tilde{z}_{\text{train}}\right) && (135)\\
&= U(S^2 + \lambda I)^{-1}U^T \left(US^2U^T \beta_{\text{true}} + USV^T \tilde{z}_{\text{train}}\right) && (136)\\
&= U(S^2 + \lambda I)^{-1}S^2U^T \beta_{\text{true}} + U \left(S^2 + \lambda I\right)^{-1} SV^T \tilde{z}_{\text{train}}, && (137)
\end{aligned}$$

because V is an orthogonal matrix. The result then follows from Theorem 8.6 in the PCA lecture notes. □

In contrast to the OLS estimator, the ridge-regression estimator is not centered at the true coefficients. Instead, it is centered at β_{bias}, which is the center of the deterministic quadratic component in the cost function,

$$(\beta - \beta_{\text{true}})^T XX^T (\beta - \beta_{\text{true}}) + \lambda \beta^T \beta. \qquad (138)$$

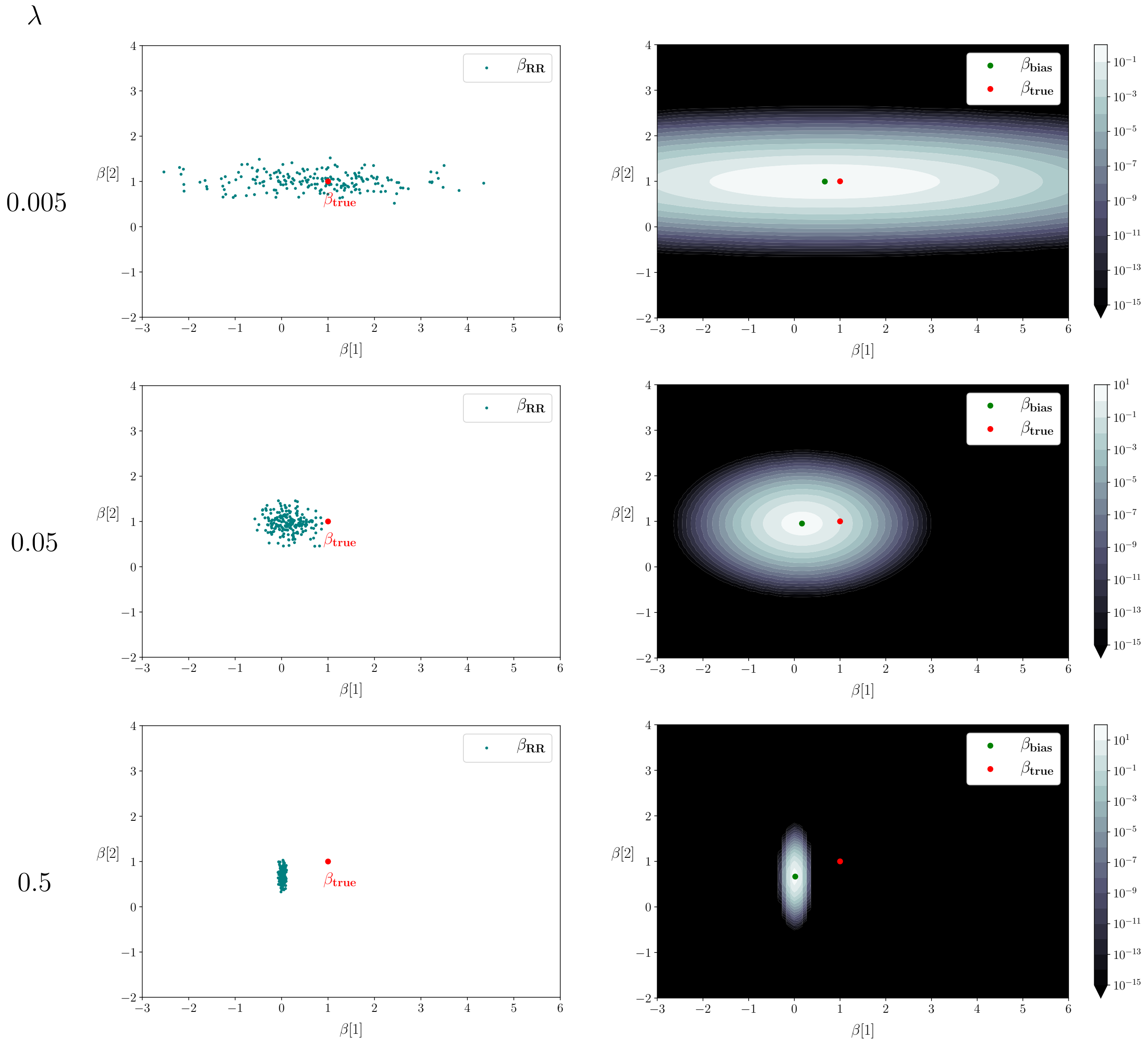


Figure 16: The left image is a scatterplot of the ridge-regression estimate corresponding to different noise realizations of the example in Figure 15. The right image is a heatmap of the distribution of the estimate, which follows a Gaussian distribution with the mean and covariance matrix derived in Theorem 5.3. Each row corresponds to a different choice of the regularization parameter λ, illustrating the corresponding bias-variance tradeoff.

As a result, the estimator has a systematic error equal to

$$\beta_{\text{true}} - \mathrm{E}\left(\tilde{\beta}_{\text{RR}}\right) = \beta_{\text{true}} - U(S^2 + \lambda I)^{-1} S^2 U^T \beta_{\text{true}} \tag{139}$$

$$= \sum_{j=1}^{p} \langle u_j, \beta_{\text{true}} \rangle u_j - \sum_{j=1}^{p} \frac{s_j^2 \langle u_j, \beta_{\text{true}} \rangle}{s_j^2 + \lambda} u_j \tag{140}$$

$$= \sum_{j=1}^{p} \frac{\lambda \langle u_j, \beta_{\text{true}} \rangle}{s_j^2 + \lambda} u_j. \tag{141}$$

The expected error is called *bias* in statistics. The bias of ridge regression increases with λ, since the derivative of $(\lambda/(s_i+\lambda))^2$ with respect to λ equals $2\lambda s_i/(s_i+\lambda)^3$. As λ increases, the expected value of the estimate is shrunk towards zero. This may seem puzzling at first: why not just set λ to zero, and just use the OLS estimate which is unbiased? The reason is the *variance* of the estimate. Increasing λ decreases the variance of the estimator.

In OLS ($\lambda = 0$) the variance in the direction of each left singular vector of the feature matrix is proportional to σ^2/s_i^2, where s_i is the corresponding singular value. This produces severe noise amplification if any of the singular values are very small. As explained at the end of Section 4.2, this results in significant test error if the sample covariance matrix is not a good approximation of the true covariance matrix, which often occurs when the number of training data is small. The role of λ is to neutralize the contribution of the small singular values. If $\lambda \gg s_i^2$, then the variance in the direction of the corresponding singular vector is approximately equal to $\sigma^2 s_i^2/\lambda^2$, which is much smaller than σ^2/s_i^2. The ideal value of λ strikes a balance between increasing the bias and decreasing the variance. In statistics this is known as the bias-variance tradeoff. Figure 16 shows the distribution of the ridge-regression estimator for a simple example when the value of λ varies. When λ is very small, the estimate resembles the OLS estimate: it is almost centered at the true coefficients, but it varies wildly in the direction of the singular vectors associated with small singular values. As λ increases the variance decreases, but the center of the distribution strays farther and farther away from the true coefficients.

6 Gradient descent

Gradient descent is the simplest and most popular iterative optimization method. The idea is to make progress towards the minimum of a cost function by moving in the direction of steepest descent[3]. In this section we analyze the properties of a linear-regression estimate obtained by applying gradient descent to the least-squares cost function. For a response vector $y \in \mathbb{R}^n$ and a feature matrix $X := \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix} \in \mathbb{R}^{p \times n}$ the gradient of function equals

$$\nabla f(\beta) = X X^T \beta - X y. \tag{142}$$

[3] For a cost function f, the directional derivative in the direction of a unit-norm vector v at a point x equals $\langle \nabla f(x), v \rangle$. In the direction $-\nabla f(x)$ it equals $-\left|\left|\nabla f(x)\right|\right|_2$. This is the smallest possible derivative since $\langle \nabla f(x), v \rangle \geq -\left|\left|v\right|\right|_2 \left|\left|\nabla f(x)\right|\right|_2$ by the Cauchy-Schwarz inequality.

The gradient-descent updates are

$$\beta^{(k+1)} := \beta^{(k)} + \alpha_k X \left(y - X^T \beta^{(k)} \right) \tag{143}$$

$$= \beta^{(k)} + \alpha_k \sum_{i=1}^{n} \left(y[i] - \langle x_i, \beta^{(k)} \rangle \right) x_i, \tag{144}$$

where $\beta^{(k)} \in \mathbb{R}^p$ and $\alpha_k > 0$ are the coefficient estimate and the step size respectively at iteration k. Gradient descent iteratively corrects the coefficient vector. If an entry of the response vector $y[i]$ is larger than the linear estimate $\langle x_i, \beta^{(k)} \rangle$ we add a small multiple of $x^{(i)}$ in order to reduce the difference. If it is smaller we subtract it.

The following theorem provides a closed-form solution for the iterations of gradient descent in terms of the SVD of the feature matrix when the step size is constant.

Theorem 6.1. *Let $X^{p \times n}$, $n \geq p$, be full rank. The $k+1$th iteration of gradient descent with a constant step size $\alpha > 0$ applied to the least-squares cost function equals*

$$\beta^{(k+1)} = U \operatorname{diag}_{j=1}^{p} \left(\left(1 - \alpha s_j^2\right)^{k+1} \right) U^T \beta^{(0)} + U \operatorname{diag}_{j=1}^{p} \left(\frac{1 - \left(1 - \alpha s_j^2\right)^{k+1}}{s_j} \right) V^T y, \quad k = 1, 2, 3 \ldots,$$

where USV^T is the SVD of X, $\beta^{(0)} \in \mathbb{R}^p$ is the initial coefficient vector, and $\operatorname{diag}_{j=1}^{p}(d_i)$ denotes a diagonal matrix with entries $d_1, \ldots, d_p$.

Proof. We reformulate Eq. (143) as

$$\beta^{(k+1)} = \left(I - \alpha X X^T \right) \beta^{(k)} + \alpha X y, \tag{145}$$

which yields

$$\beta^{(k+1)} = \left(I - \alpha X X^T \right)^{k+1} \beta^{(0)} + \sum_{i=0}^{k} \left(I - \alpha X X^T \right)^i \alpha X y. \tag{146}$$

Since $p \leq n$ and X is full rank, we have $UU^T = U^T U = I$, so that

$$\beta^{(k+1)} = \left(UU^T - \alpha U S^2 U^T \right)^{k+1} \beta^{(0)} + \alpha \sum_{i=0}^{k} \left(UU^T - \alpha U S^2 U^T \right)^i U S V^T y \tag{147}$$

$$= U \left(I - \alpha S^2 \right)^{k+1} U^T \beta^{(0)} + \alpha U \sum_{i=0}^{k} \left(I - \alpha S^2 \right)^i S V^T y \tag{148}$$

$$= U \operatorname{diag}_{j=1}^{p} \left(\left(1 - \alpha s_j^2\right)^{k+1} \right) U^T \beta^{(0)} + \alpha U \operatorname{diag}_{j=1}^{p} \left(\sum_{i=0}^{k} \left(1 - \alpha s_j^2\right)^i \right) S V^T y. \tag{149}$$

By the geometric-sum formula we conclude:

$$\beta^{(k+1)} = U \operatorname{diag}_{j=1}^{p} \left(\left(1 - \alpha s_j^2\right)^{k+1} \right) U^T \beta^{(0)} + \alpha U \operatorname{diag}_{j=1}^{p} \left(\frac{1 - \left(1 - \alpha s_j^2\right)^{k+1}}{\alpha s_j^2} \right) S V^T y. \tag{150}$$

□

An immediate consequence is that gradient descent converges to the optimal solution if the step size is small enough.

Corollary 6.2. *Let $0 < \alpha < 2/s_1^2$, where s_1 is the largest singular value of X. If X is full rank, gradient descent with step size α converges to the minimum of the least-squares cost function.*

Proof. If $0 < \alpha < 2/s_1^2 \leq 2/s_j^2$ for $1 \leq j \leq p$ then $\left|1 - \alpha s_j^2\right| < 1$ so $\lim_{k\to\infty} \left(1 - \alpha s_j^2\right)^k = 0$. This implies

$$\lim_{k\to\infty} \beta^{(k)} = \lim_{k\to\infty} U \operatorname{diag}_{j=1}^{p} \left(\left(1 - \alpha s_j^2\right)^k\right) U^T \beta^{(0)} + U \operatorname{diag}_{j=1}^{p} \left(\frac{1 - \left(1 - \alpha s_j^2\right)^k}{s_j}\right) V^T y \tag{151}$$

$$= U S^{-1} V^T y, \tag{152}$$

which is the solution to the least-squares problem. □

The response estimate produced by gradient descent consequently converges to the OLS prediction. The rate of convergence is governed by the condition number of the feature matrix. To simplify the exposition, we assume that the coefficient estimate is initialized to equal the zero vector.

Corollary 6.3. *Let $y_{\text{OLS}} := X^T \beta_{\text{OLS}}$, where β_{OLS} is the solution to the least-squares problem, and $y^{(k)} := X\beta^{(k)}$, where $\beta^{(k)}$ is the kth iteration of gradient descent initialized with the zero vector. If the step size is set to $\alpha := 1/s_1^2$ then*

$$\frac{\left|\left|y_{\text{OLS}} - y^{(k)}\right|\right|_2}{\left|\left|y\right|\right|_2} \leq \left(1 - \frac{s_p^2}{s_1^2}\right)^k, \tag{153}$$

where s_1 is the largest singular value of X and s_p is the smallest.

Proof. By Theorem 6.1, if $\beta^{(0)}$ is the zero vector,

$$y^{(k)} := X^T \beta^{(k)} \tag{154}$$

$$= V S U^T U \operatorname{diag}_{j=1}^{p} \left(\frac{1 - \left(1 - \alpha s_j^2\right)^k}{s_j}\right) V^T y \tag{155}$$

$$= V V^T y - V \operatorname{diag}_{j=1}^{p} \left(\left(1 - \alpha s_j^2\right)^k\right) V^T y. \tag{156}$$

The operator norm $||M||$ of a matrix M is equal to its largest singular value. By Theorem 3.2, for any vector w $||Mw|| \leq ||M|| \, ||w||_2$. Since $y_{\text{OLS}} = V V^T y$ by Lemma 3.3, this implies

$$\left|\left|y_{\text{OLS}} - y^{(k)}\right|\right|_2 = \left|\left|V \operatorname{diag}_{j=1}^{p} \left(\left(1 - \alpha s_j^2\right)^k\right) V^T y\right|\right|_2 \tag{157}$$

$$\leq ||V|| \left|\left|\operatorname{diag}_{j=1}^{p} \left(\left(1 - \alpha s_j^2\right)^k\right)\right|\right| \left|\left|V^T y\right|\right|_2 \tag{158}$$

$$\leq \left|1 - \frac{s_p^2}{s_1^2}\right|^k ||y||_2 \tag{159}$$

because $\left(1 - \alpha s_p^2\right)^k$ is the largest singular value of the diagonal matrix, and V has orthonormal columns. □

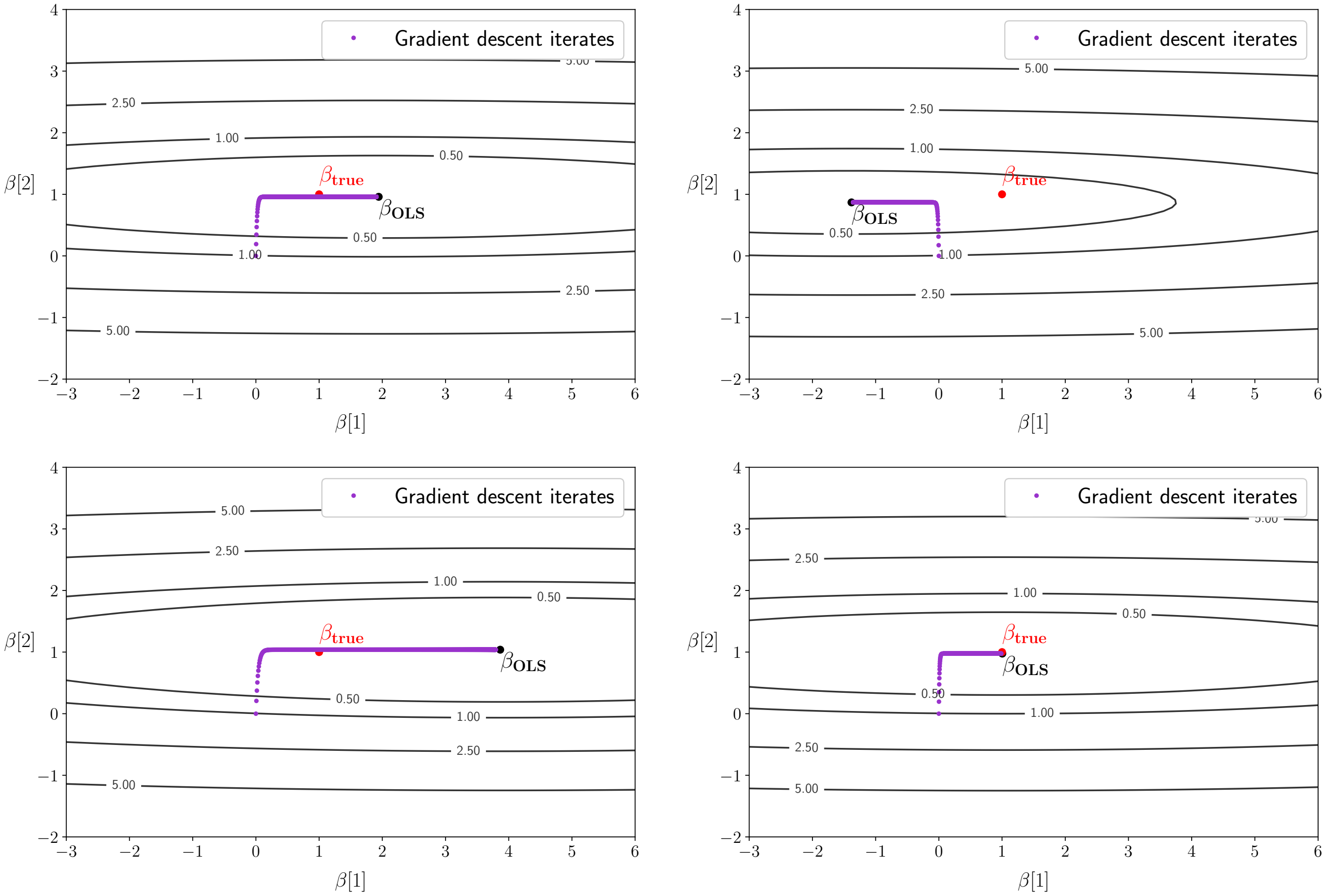


Figure 17: Iterates of gradient descent initialized at the origin with a fixed step size for the example in Figure 6. Each image corresponds to a different noise realization.

If the feature matrix is well conditioned, convergence is fast, but if there are singular values that are much smaller than the rest, gradient descent can take very long to converge. Large condition numbers are common in practical applications: the feature matrix in the temperature-prediction example has condition number around 10^3 (see Figure 10). If one cares about finding the least-squares solution fast, the method of choice should instead be conjugate gradients method, an optimization technique designed to achieve fast convergence. However, what we really care about is achieving a good estimate. It may therefore be of interest to evaluate the estimate produced by gradient descent for a fixed value of k, before convergence occurs. This technique is known as early stopping in the machine-learning literature. The following theorem provides a characterization of the estimate obtained via early stopping for data generated according to an additive generative model.

Theorem 6.4 (Gradient-descent coefficient estimate)**.** *If the training data follow the additive model in Eq.* (68)*, then the coefficient estimate obtained by running gradient descent initialized at the origin until the* k*th iteration with a constant step size* $\alpha > 0$ *is a Gaussian random vector with mean*

$$\beta_{\text{bias}} := \sum_{j=1}^{p} \left(1 - (1 - \alpha s_j^2)^k\right) \langle u_j, \beta_{\text{true}} \rangle \, u_j \tag{160}$$

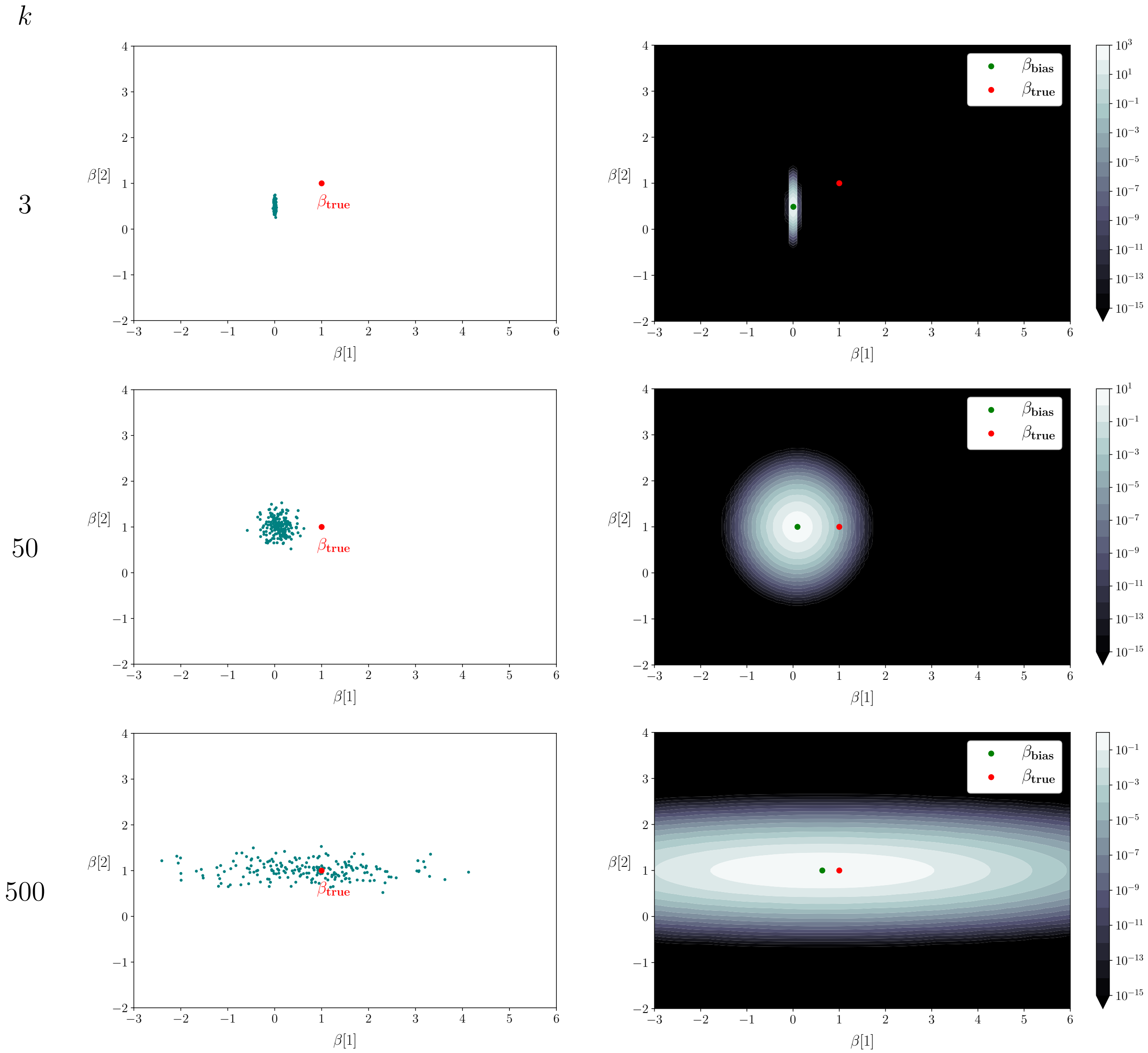


Figure 18: The left image is a scatterplot of the gradient-descent estimate corresponding to different noise realizations of the example in Figure 17. The right image is a heatmap of the distribution of the estimate, which follows a Gaussian distribution with the mean and covariance matrix derived in Theorem 6.4. Each row corresponds to a different choice of the number of iterations k, illustrating the corresponding bias-variance tradeoff.

and covariance matrix

$$\Sigma_{\mathrm{GD}} := \sigma^2 U \operatorname{diag}_{j=1}^{p} \left(\frac{(1-(1-\alpha s_j^2)^k)^2}{s_j^2} \right) U^T, \tag{161}$$

where $\operatorname{diag}_{j=1}^{p}(d_i)$ *denotes a diagonal matrix with entries* $d_1, \ldots, d_p$.

Proof. To ease notation, let $\tau_j := 1 - \alpha s_j^2$. By Theorem 6.1

$$\begin{aligned}
\tilde{\beta}^{(k)} &= U \operatorname{diag}_{j=1}^{p} \left(\frac{1-\tau_j^k}{s_j} \right) V^T \left(X^T \beta_{\mathrm{true}} + \tilde{z}_{\mathrm{train}} \right) && (162)\\
&= U \operatorname{diag}_{j=1}^{p} \left(\frac{1-\tau_j^k}{s_j} \right) V^T \left(V S U^T \beta_{\mathrm{true}} + \tilde{z}_{\mathrm{train}} \right) && (163)\\
&= U \operatorname{diag}_{j=1}^{p} \left(1-\tau_j^k \right) U^T \beta_{\mathrm{true}} + U \operatorname{diag}_{j=1}^{p} \left(\frac{1-\tau_j^k}{s_j} \right) V^T \tilde{z}_{\mathrm{train}}. && (164)
\end{aligned}$$

The result then follows from Theorem 8.6 in the PCA lecture notes. □

As shown in Figure 17 the first iterates of gradient descent make fast progress along the directions of left singular vectors of the feature matrix corresponding to large singular values. Afterwards, the iterates move along the directions corresponding to the smaller singular values, until they converge to the OLS estimate. As a result, if we stop at iteration k, the expected value of the iterate is not centered at β_{true}; it is closer to the point at which gradient descent is initialized (the origin, in our analysis and examples). This produces a bias equal to $\sum_{j=1}^{p}(1-\alpha s_j^2)^k \langle u_j, \beta_{\mathrm{true}} \rangle u_j$ in the estimate, which decreases as k increases. As in the case of ridge regression, the reduction in bias is counterbalanced by an increase of the variance. Because the algorithm mostly makes progress in the direction of the singular vectors corresponding to the largest singular values, there is not as much variance in the direction of those corresponding to the small singular values. This is good news, because that is the source of most of the variance in the OLS estimate. At iteration k, the variance in the direction of the jth left singular vector equals

$$\frac{\sigma^2(1-(1-\alpha s_j^2)^k)^2}{s_j^2}. \tag{165}$$

For small k and small αs_j, we have $(1-\alpha s_j^2)^k \approx 1 - k\alpha s_j^2$ (because for $x \approx 0$ $(1-x)^k \approx 1-kx$), so the variance of the corresponding component approximately equals $\alpha^2 k^2 \sigma^2$. Then, as k increases, the variance also increases, eventually approaching σ^2/s_j^2, as in OLS. The ideal value of k should optimize the bias-variance tradeoff, as in ridge regression. Figure 18 shows the distribution of the gradient-descent estimator for a simple example when k varies. For large k, the estimate resembles the OLS estimate: it is almost centered at the true coefficients, but it varies wildly in the direction of the singular vectors associated with small singular values. As k decreases the variance along those directions also decreases, but the center of the distribution strays farther and farther away from the true coefficients.

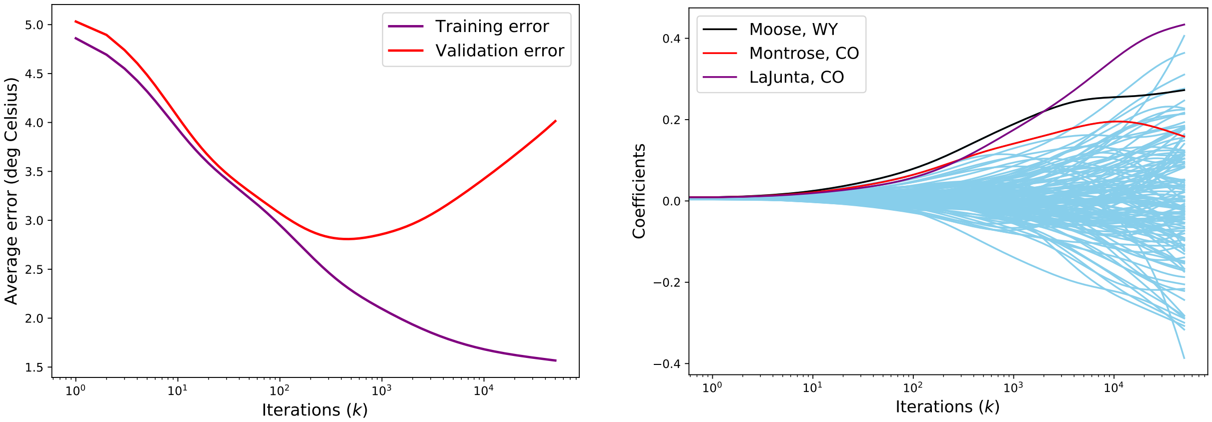


Figure 19: The left graph shows the training and validation errors of the gradient-descent estimator applied to the temperature-prediction task as the iterations progress. The number of training data is fixed to $n = 200$ training data. The right figure shows the values of the corresponding model coefficients. All coefficients are depicted in light blue except the three that have the largest magnitudes for large n, which correspond to the stations of Moose in Wyoming, and Montrose and La Junta in Colorado.

Example 6.5 (Temperature prediction via gradient descent with early stopping)**.** We apply gradient descent to minimize the least-squares cost function for the data in Example 2.5. The coefficients are initialized to be zero. The number of iterations of gradient descent are chosen by minimizing the error over a separate validation set. In addition, we test the model on data from 2016. The left image in Figure 19 shows training and validation errors of the gradient-descent estimator for $n = 200$ training data as the iterations progress. Both initially decrease, but at one point the validation error starts increasing due to overfitting. The right image shows that the coefficients amplitudes increase until they reach the value of the least-squares estimator. The minimum validation error is reached when the coefficients are still not too large. Figure 20 shows the number of iterations selected for different numbers of training data based on validation error. Figure 21 shows the corresponding coefficients and compares them the OLS coefficients. The effect achieved by early stopping is reminiscent of ridge regression. Figure 22 compares the error obtained by the estimator on training and test data compared to least squares and ridge regression. The method avoids the overfitting issues of least squares when the number of training data is small, and achieves very similar results to ridge regression. △

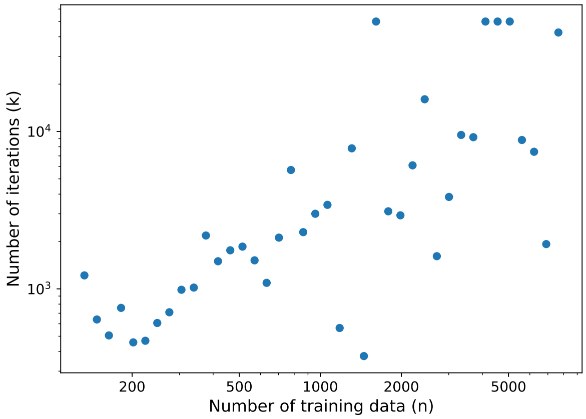


Figure 20: Results of selecting the number of iterations via cross-validation for the experiment described in Example 6.5. The image shows the number of iterations at which the gradient-descent estimator achieves minimum validation error for different numbers of training data. The maximum number of iterations was limited to 10^5.

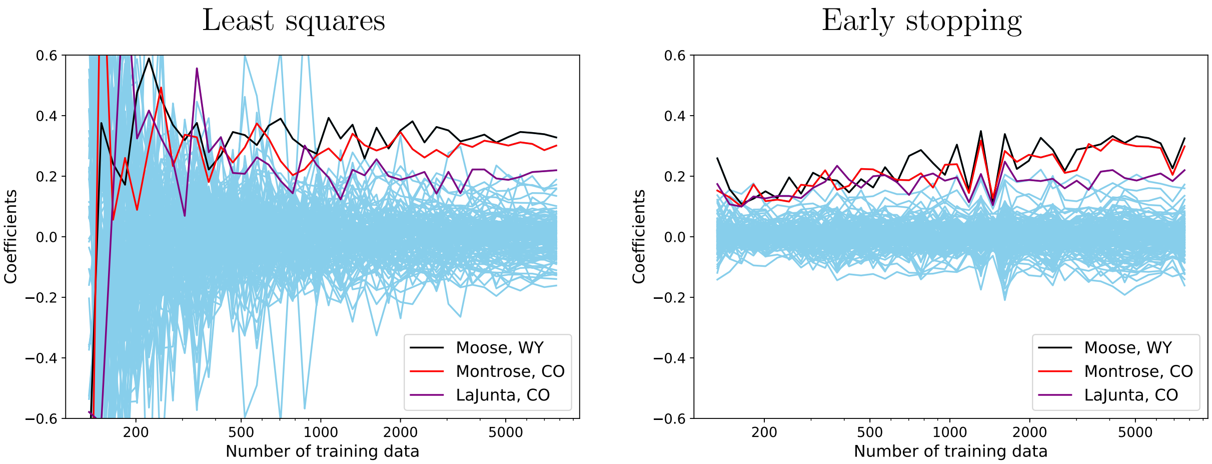


Figure 21: Coefficients of the least-squares (left) and gradient-descent (right) estimators for the experiment described in Example 6.5 for different values of training data. All coefficients are depicted in light blue except the three that have the largest magnitudes for large n, which correspond to the stations of Moose in Wyoming, and Montrose and La Junta in Colorado.

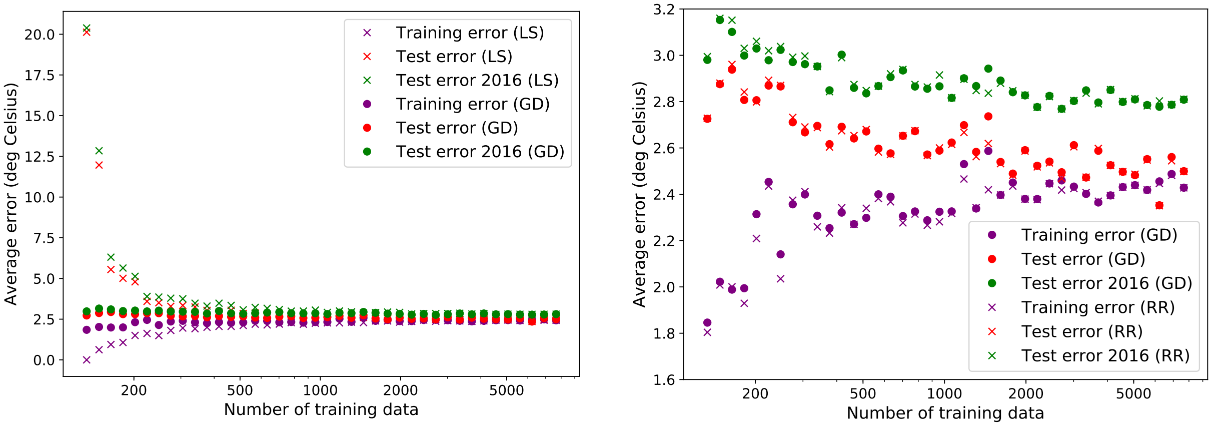


Figure 22: Performance of the gradient-descent estimator for the experiment described in Example 6.5. The left image compares the method to the least-squares estimator on the training and test sets, and on the 2016 data, for different number of training data. The right image shows the same comparison to the ridge-regression estimator.