Regularization

1 Ridge regression

1.1 Motivation and definition

As seen in the previous notes, the OLS 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 12. 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 ℓ_2-norm regularization is known as ridge regression in statistics and as Tikhonov regularization in the literature on inverse problems.

Definition 1.1 (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*

$$\beta_{\text{RR}} := \arg\min_{\beta} \|y - X^T\beta\|_2^2 + \lambda\|\beta\|_2^2, \tag{1}$$

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

$$\beta_{\text{RR}} = \left(XX^T + \lambda I\right)^{-1} Xy. \tag{2}$$

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

$$\beta_{\text{RR}} := \arg\min_{\beta} \left|\left| \begin{bmatrix} y \\ 0 \end{bmatrix} - \begin{bmatrix} X^T \\ \sqrt{\lambda} I \end{bmatrix} \beta \right|\right|_2^2. \tag{3}$$

Applying the formula for the closed-form solution of the OLS estimator yields

$$\beta_{\text{RR}} = \left(\begin{bmatrix} X & \sqrt{\lambda} I \end{bmatrix} \begin{bmatrix} X & \sqrt{\lambda} I \end{bmatrix}^T \right)^{-1} \begin{bmatrix} X & \sqrt{\lambda} I \end{bmatrix} \begin{bmatrix} y \\ 0 \end{bmatrix} \tag{4}$$

$$= \left(XX^T + \lambda I\right)^{-1} Xy. \tag{5}$$

□

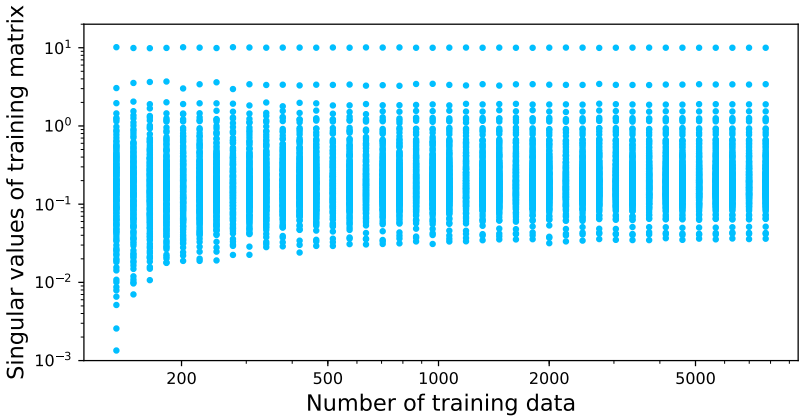

Figure 1: Singular values of the training matrix in Example 2.2 of the notes on linear regression for different numbers of training data.

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 3, 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 3 also shows the coefficients of the model applied to the data described in Example 2.2 of the notes on linear regression 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 4 shows that ridge regression outperforms least-squares regression on the temperature dataset 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 are larger for small values of n, where regularization is more useful.

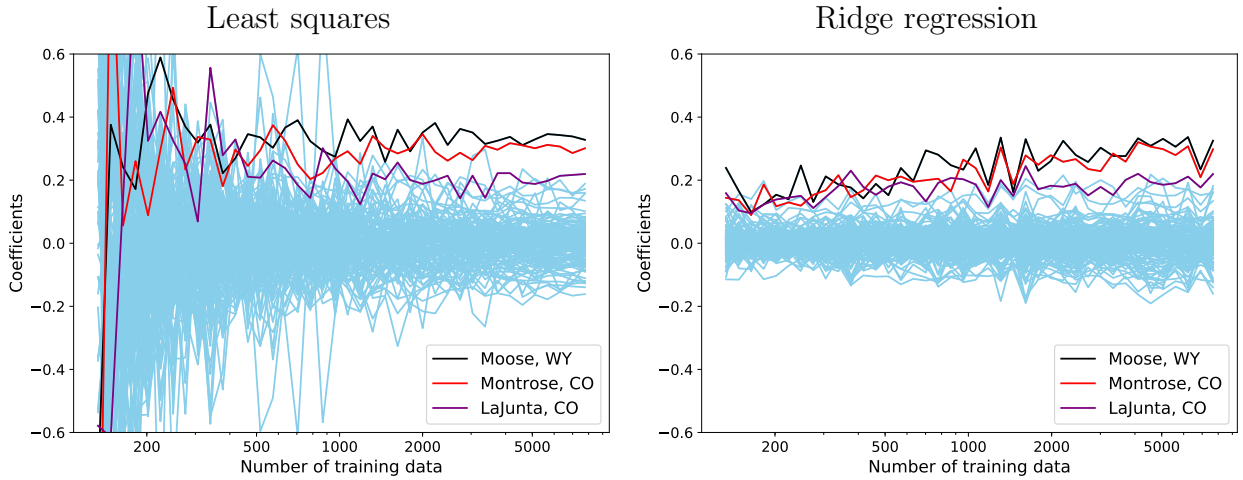

Figure 2: Coefficients of the least-squares (left) and ridge-regression (right) estimators computed from the data described in Example 2.2 of the notes on linear regression for different values of training data. All coefficients are depicted in light blue except the three that have the largest magnitudes for large n, which correspond to the stations of Moose in Wyoming, and Montrose and La Junta in Colorado.

1.2 Analysis under additive-noise model

In order to analyze the ridge-regression estimator, we consider data generated by a linear model as in Section 4 of the notes on linear regression. The training data are equal to the n-dimensional vector

$$\tilde{y}_{\text{train}} := X^T \beta_{\text{true}} + \tilde{z}_{\text{train}}, \tag{6}$$

where $X \in \mathbb{R}^{p \times n}$ contains n p-dimensional feature vectors. The noise $\tilde{z}_{\text{train}}$ is modeled as an n-dimensional iid Gaussian vector with zero mean and variance σ^2.

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 the decomposition of the OLS estimator, we obtain

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

Figure 5 shows the different components for a simple example with two features. The following theorem provides the distribution of the ridge-regression coefficient estimate for the additive model.

Theorem 1.3 (Ridge-regression coefficient estimate)**.** *If the training data follow the additive model in Eq. (6), then the ridge regression coefficient estimate is a Gaussian random vector with mean*

$$\beta_{\text{bias}} := \sum_{j=1}^{p} \frac{s_j^2 \langle u_j, \beta_{\text{true}} \rangle}{s_j^2 + \lambda} u_j \tag{7}$$

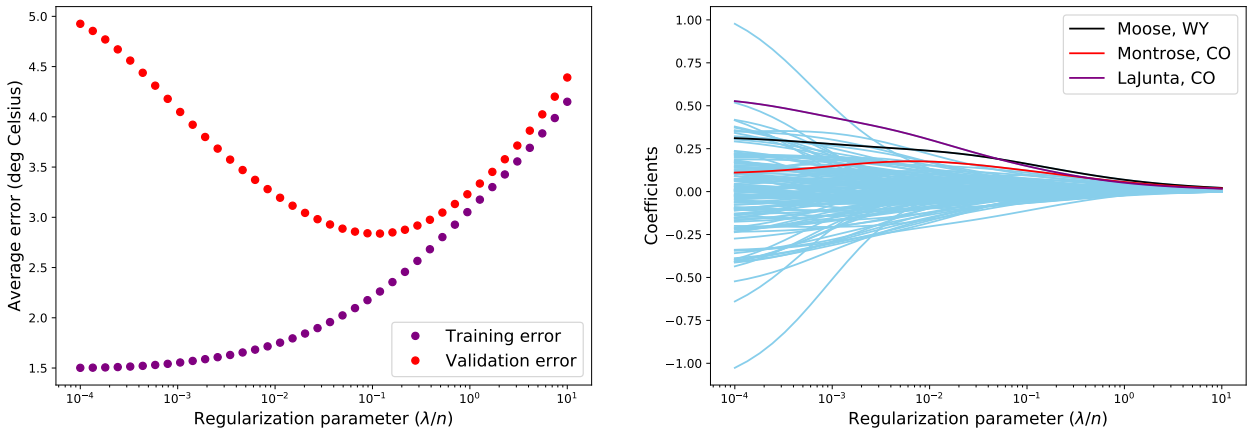

Figure 3: The left graph shows the training and validation errors of the ridge-regression estimator applied to the data described in Example 2.2 of the notes on linear regression for different values of the regularization parameter λ. The number of training data is fixed to $n = 202$ training data. The 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.

and covariance matrix

$$\Sigma_{\text{RR}} := \sigma^2 U \operatorname{diag}_{j=1}^{p} \left(\frac{s_j^2}{(s_j^2 + \lambda)^2} \right) U^T, \tag{8}$$

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

Proof. By Theorem 1.2 the solution equals

$$\begin{aligned}
\tilde{\beta}_{\text{RR}} &= \left(XX^T + \lambda I\right)^{-1} X \left(X^T \beta_{\text{true}} + \tilde{z}_{\text{train}}\right) && (9)\\
&= \left(US^2U^T + \lambda UU^T\right)^{-1} \left(US^2U^T \beta_{\text{true}} + USV^T \tilde{z}_{\text{train}}\right) && (10)\\
&= \left(U(S^2 + \lambda I)U^T\right)^{-1} \left(US^2U^T \beta_{\text{true}} + USV^T \tilde{z}_{\text{train}}\right) && (11)\\
&= U(S^2 + \lambda I)^{-1} U^T \left(US^2U^T \beta_{\text{true}} + USV^T \tilde{z}_{\text{train}}\right) && (12)\\
&= U(S^2 + \lambda I)^{-1} S^2 U^T \beta_{\text{true}} + U\left(S^2 + \lambda I\right)^{-1} SV^T \tilde{z}_{\text{train}}, && (13)
\end{aligned}$$

because V is an orthogonal matrix. □

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,

$$(\beta - \beta_{\text{true}})^T XX^T (\beta - \beta_{\text{true}}) + \lambda \beta^T \beta. \tag{14}$$

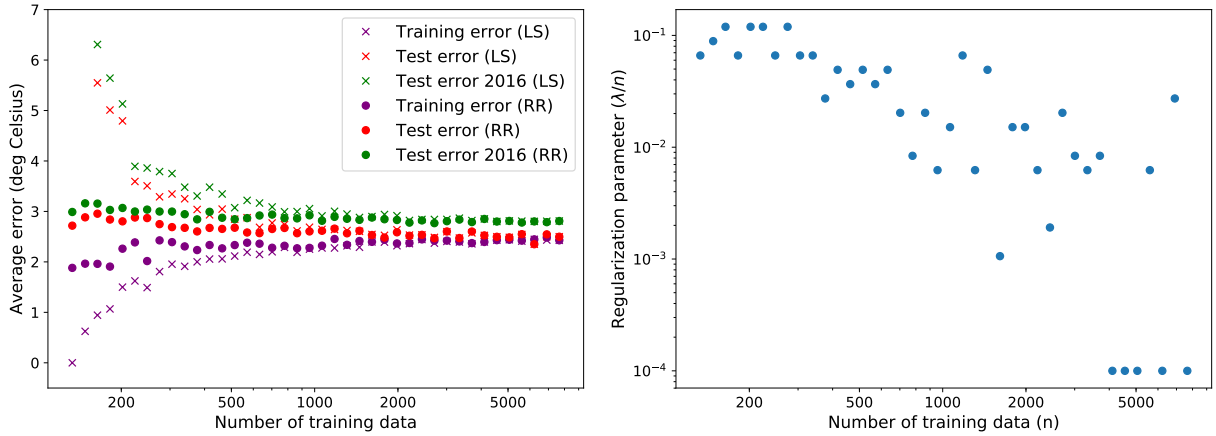

Figure 4: Performance of the ridge-regression estimator on the temperature data. The left image shows the RMSE achieved by the model on the training and test sets, and on the 2016 data, for different number of training data and compares it to the RMSE of least-squares regression. The right graph shows the values of λ selected from a validation dataset of size 100 for each number of training data.

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

$$\beta_{\text{true}} - \mathrm{E}\left(\tilde{\beta}_{\text{RR}}\right) = \beta_{\text{true}} - U(S^2 + \lambda I)^{-1} S^2 U^T \beta_{\text{true}} \tag{15}$$

$$= \sum_{j=1}^{p} \langle u_j, \beta_{\text{true}} \rangle \, u_j - \sum_{j=1}^{p} \frac{s_j^2 \, \langle u_j, \beta_{\text{true}} \rangle}{s_j^2 + \lambda} u_j \tag{16}$$

$$= \sum_{j=1}^{p} \frac{\lambda \, \langle u_j, \beta_{\text{true}} \rangle}{s_j^2 + \lambda} u_j. \tag{17}$$

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 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 in Section 4.3 of the notes on linear regression, 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 7 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

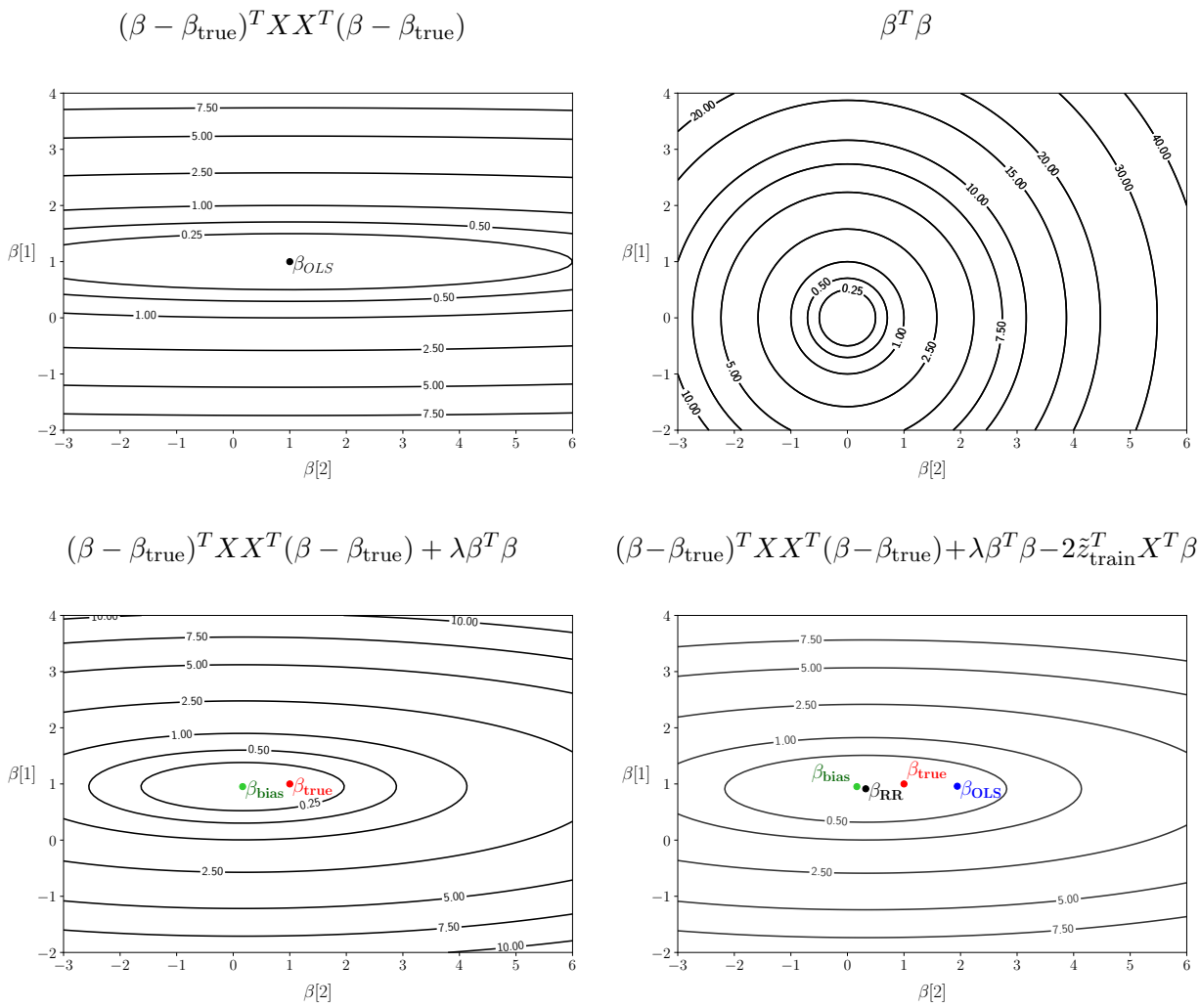

Figure 5: Visualization of the different components of the ridge-regression cost function for a simple 2D example. The regularization parameter is set to $\lambda := 0.05$. The top row shows the two deterministic quadratic forms cost function: the least square component (left) and the regularization 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}.

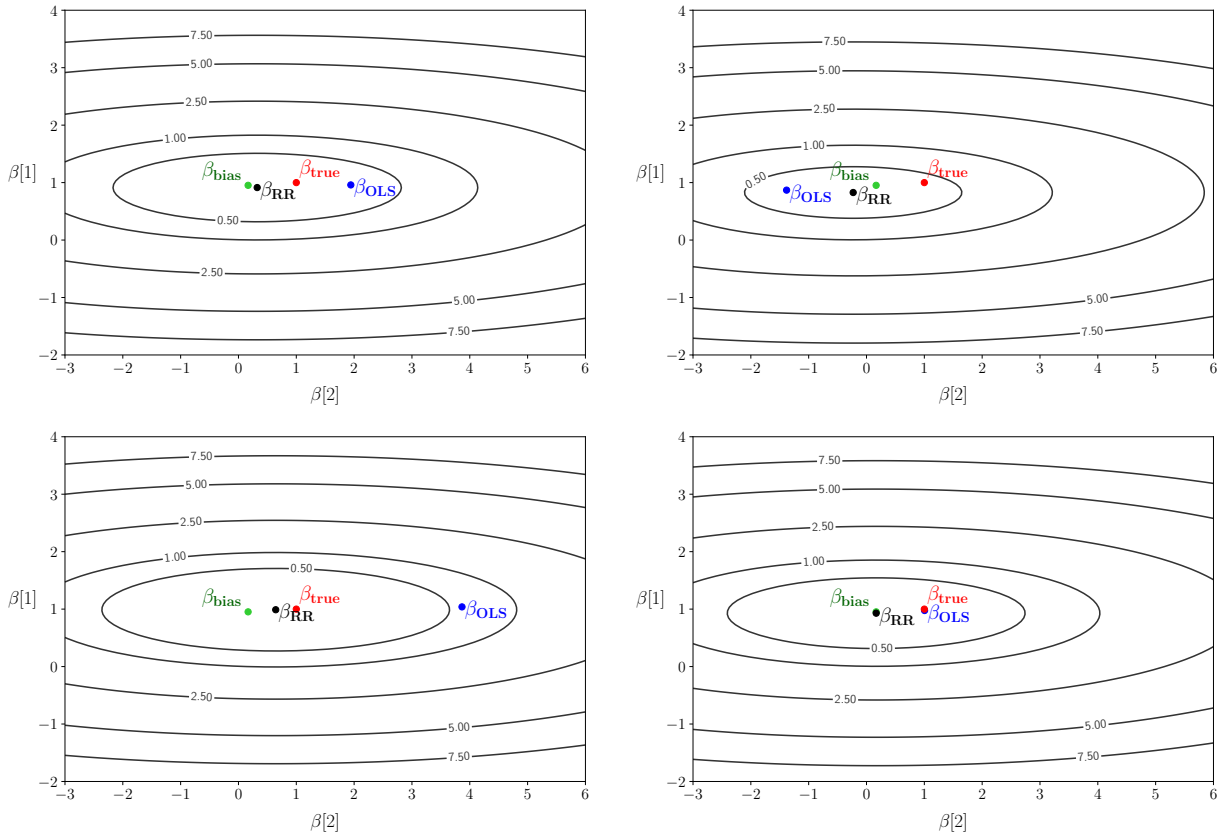

Figure 6: Different realizations of the ridge-regression cost function corresponding to different realizations of the noise (the true coefficients and the feature matrix remain the same) for the example in Figure 5. The regularization parameter is set to $\lambda := 0.05$.

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.

2 Regularization via early stopping

2.1 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[1]. 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

[1]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.

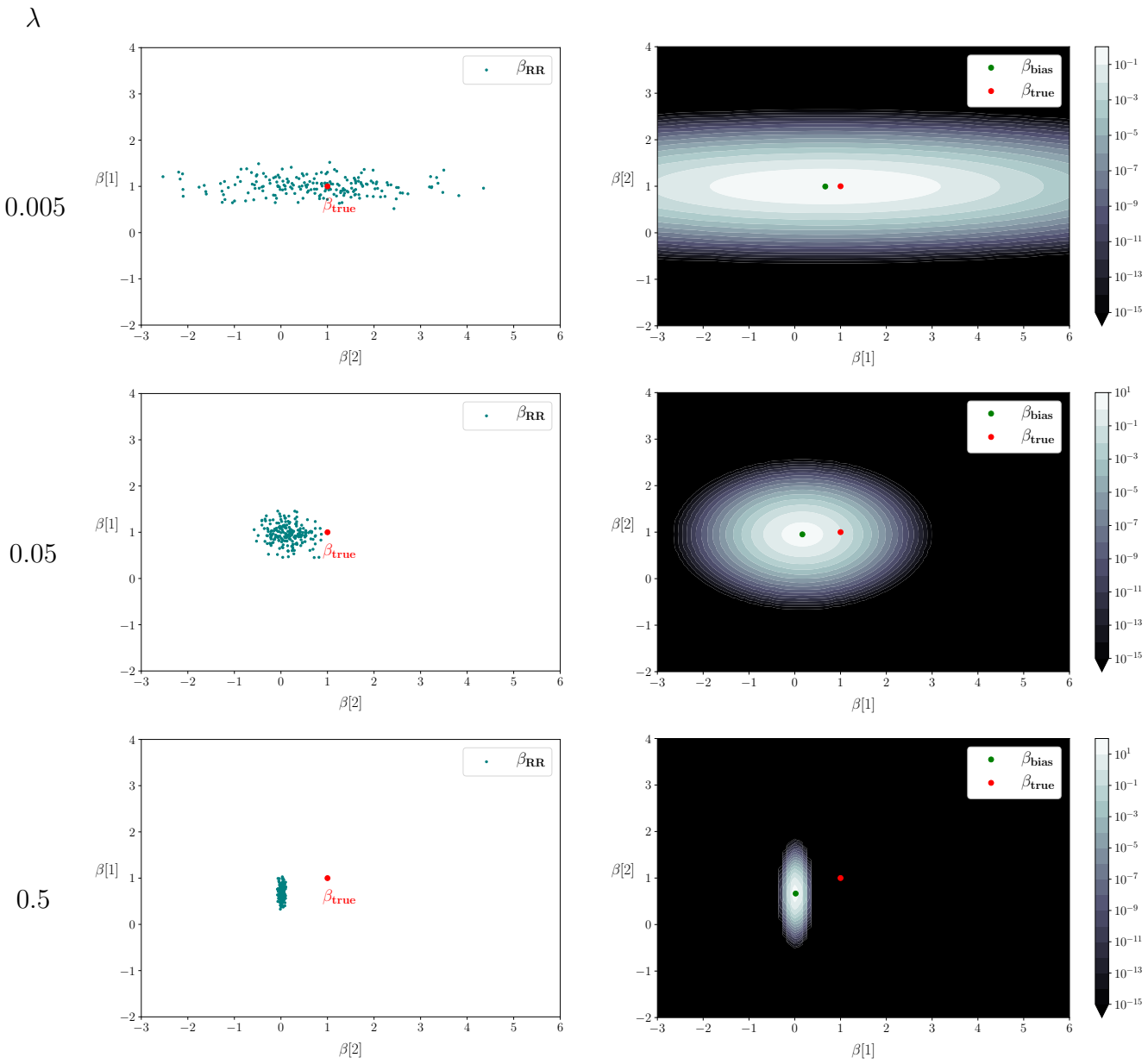

Figure 7: The left image is a scatterplot of the ridge-regression estimate corresponding to different noise realizations of the example in Figure 6. The right image is a heatmap of the distribution of the estimate, which follows a Gaussian distribution with the mean and covariance matrix derived in Theorem 1.3. Each row corresponds to a different choice of the regularization parameter λ, illustrating the corresponding bias-variance tradeoff.

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

$$\nabla f(\beta) = XX^T\beta - Xy. \tag{18}$$

The gradient-descent updates are

$$\beta^{(k+1)} := \beta^{(k)} + \alpha_k X \left(y - X^T\beta^{(k)}\right) \tag{19}$$

$$= \beta^{(k)} + \alpha_k \sum_{i=1}^{n} \left(y[i] - \langle x_i, \beta^{(k)} \rangle\right) x_i, \tag{20}$$

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 2.1. *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. (19) as

$$\beta^{(k+1)} = \left(I - \alpha XX^T\right) \beta^{(k)} + \alpha Xy, \tag{21}$$

which yields

$$\beta^{(k+1)} = \left(I - \alpha XX^T\right)^{k+1} \beta^{(0)} + \sum_{i=0}^{k} \left(I - \alpha XX^T\right)^i \alpha Xy. \tag{22}$$

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

$$\beta^{(k+1)} = \left(UU^T - \alpha US^2U^T\right)^{k+1} \beta^{(0)} + \alpha \sum_{i=0}^{k} \left(UU^T - \alpha US^2U^T\right)^i USV^T y \tag{23}$$

$$= U\left(I - \alpha S^2\right)^{k+1} U^T\beta^{(0)} + \alpha U \sum_{i=0}^{k} \left(I - \alpha S^2\right)^i SV^T y \tag{24}$$

$$= U \operatorname{diag}_{j=1}^{p} \left(\left(1 - \alpha s_j^2\right)^{k+1}\right) U^T\beta^{(0)} + \alpha U \operatorname{diag}_{j=1}^{p} \left(\sum_{i=0}^{k} \left(1 - \alpha s_j^2\right)^i\right) SV^T y. \tag{25}$$

By the geometric-sum formula we conclude:

$$\beta^{(k+1)} = U \operatorname{diag}_{j=1}^{p} \left(\left(1 - \alpha s_j^2\right)^{k+1}\right) U^T\beta^{(0)} + \alpha U \operatorname{diag}_{j=1}^{p} \left(\frac{1 - \left(1 - \alpha s_j^2\right)^{k+1}}{\alpha s_j^2}\right) SV^T y. \tag{26}$$

□

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

Corollary 2.2. *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

$$\lim_{k\to\infty} \beta^{(k)} = \lim_{k\to\infty} U \operatorname{diag}_{j=1}^{p} \left(\left(1 - \alpha s_j^2\right)^k\right) U^T \beta^{(0)} + U \operatorname{diag}_{j=1}^{p} \left(\frac{1 - \left(1 - \alpha s_j^2\right)^k}{s_j}\right) V^T y \tag{27}$$

$$= US^{-1}V^T y, \tag{28}$$

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 2.3. *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*

$$\frac{\left|\left|y_{\text{OLS}} - y^{(k)}\right|\right|_2}{||y||_2} \leq \left(1 - \frac{s_p^2}{s_1^2}\right)^k, \tag{29}$$

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

Proof. By Theorem 2.1, if $\beta^{(0)}$ is the zero vector,

$$y^{(k)} := X^T \beta^{(k)} \tag{30}$$

$$= VSU^T U \operatorname{diag}_{j=1}^{p} \left(\frac{1 - \left(1 - \alpha s_j^2\right)^k}{s_j}\right) V^T y \tag{31}$$

$$= VV^T y - V \operatorname{diag}_{j=1}^{p} \left(\left(1 - \alpha s_j^2\right)^k\right) V^T y. \tag{32}$$

The operator norm $||M||$ of a matrix M is equal to its largest singular value, so for any vector w $||Mw|| \leq ||M||\,||w||_2$. Since $y_{\text{OLS}} = VV^T y$ (see Section 4.2 of the notes on linear regression), this implies

$$\left|\left|y_{\text{OLS}} - y^{(k)}\right|\right|_2 = \left|\left|V \operatorname{diag}_{j=1}^{p} \left(\left(1 - \alpha s_j^2\right)^k\right) V^T y\right|\right|_2 \tag{33}$$

$$\leq ||V||\, \left|\left|\operatorname{diag}_{j=1}^{p} \left(\left(1 - \alpha s_j^2\right)^k\right)\right|\right|\, \left|\left|V^T y\right|\right|_2 \tag{34}$$

$$\leq \left|1 - \frac{s_p^2}{s_1^2}\right|^k ||y||_2 \tag{35}$$

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

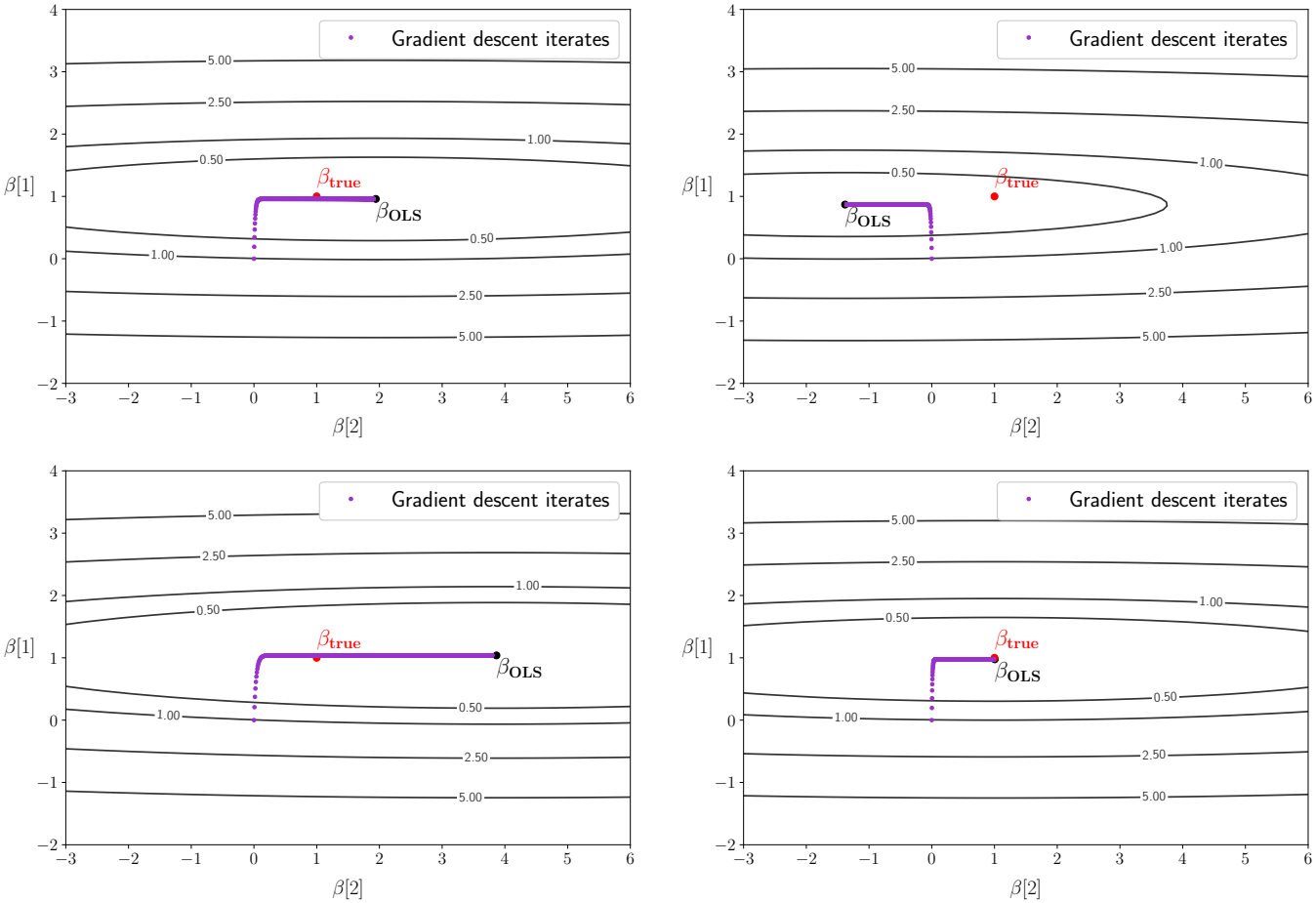

Figure 8: Iterates of gradient descent initialized at the origin with a fixed step size for the example in Figure 6. Each image corresponds to a different noise realization.

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 1). 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 2.4 (Gradient-descent coefficient estimate)**.** *If the training data follow the additive model in Eq. (6), then the coefficient estimate obtained by running gradient descent initialized at the origin until the kth iteration with a constant step size $\alpha > 0$ is a Gaussian random vector with*

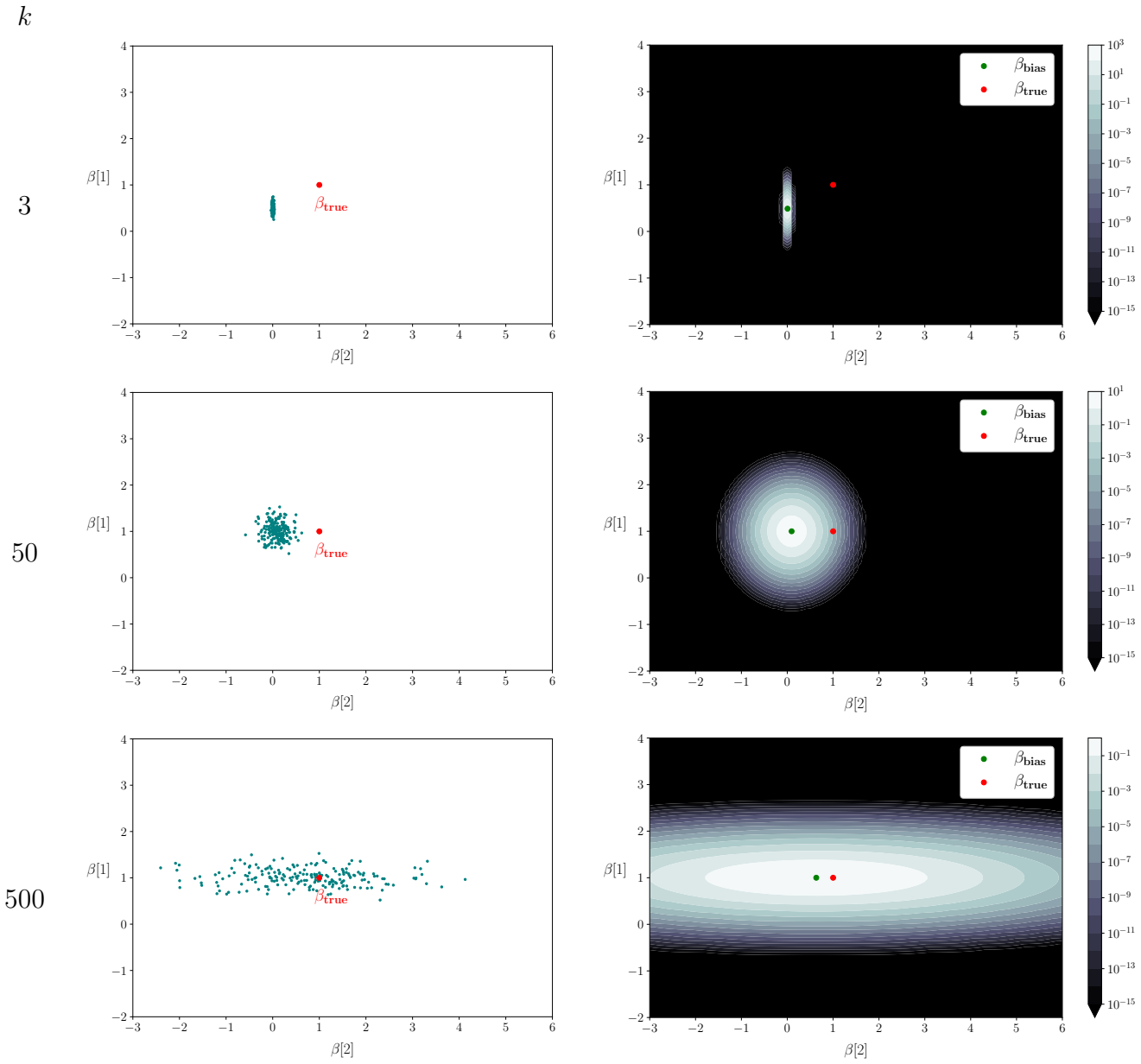

Figure 9: The left image is a scatterplot of the gradient-descent estimate corresponding to different noise realizations of the example in Figure 8. The right image is a heatmap of the distribution of the estimate, which follows a Gaussian distribution with the mean and covariance matrix derived in Theorem 2.4. Each row corresponds to a different choice of the number of iterations k, illustrating the corresponding bias-variance tradeoff.

mean

$$\beta_{\text{bias}} := \sum_{j=1}^{p} \left(1 - (1 - \alpha s_j^2)^k\right) \langle u_j, \beta_{\text{true}} \rangle \, u_j \tag{36}$$

and covariance matrix

$$\Sigma_{\text{GD}} := \sigma^2 U \, \text{diag}_{j=1}^{p} \left(\frac{(1 - (1 - \alpha s_j^2)^k)^2}{s_j^2} \right) U^T, \tag{37}$$

where $\text{diag}_{j=1}^{p}(d_i)$ *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 2.1

$$\tilde{\beta}^{(k)} = U \, \text{diag}_{j=1}^{p} \left(\frac{1 - \tau_j^k}{s_j} \right) V^T \left(X^T \beta_{\text{true}} + \tilde{z}_{\text{train}} \right) \tag{38}$$

$$= U \, \text{diag}_{j=1}^{p} \left(\frac{1 - \tau_j^k}{s_j} \right) V^T \left(V S U^T \beta_{\text{true}} + \tilde{z}_{\text{train}} \right) \tag{39}$$

$$= U \, \text{diag}_{j=1}^{p} \left(1 - \tau_j^k\right) U^T \beta_{\text{true}} + U \, \text{diag}_{j=1}^{p} \left(\frac{1 - \tau_j^k}{s_j} \right) V^T \tilde{z}_{\text{train}}. \tag{40}$$

□

2.2 Early stopping

As shown in Figure 8 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 $\sum_{j=1}^{p}(1 - \alpha s_j^2)^k \langle u_j, \beta_{\text{true}} \rangle \, u_j$ 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

$$\frac{\sigma^2 (1 - (1 - \alpha s_j^2)^k)^2}{s_j^2}. \tag{41}$$

For small k and small αs_j, we have $(1 - \alpha s_j^2)^k \approx 1 - k\alpha s_j^2$ (because for $x \approx 0$ we have $(1 - x)^k \approx 1 - kx$), so the variance of the corresponding component approximately equals $\alpha^2 k^2 \sigma^2 s_j^2$. Then, as k increases, the variance also increases, eventually approaching σ^2 / s_j^2, as in OLS. The ideal

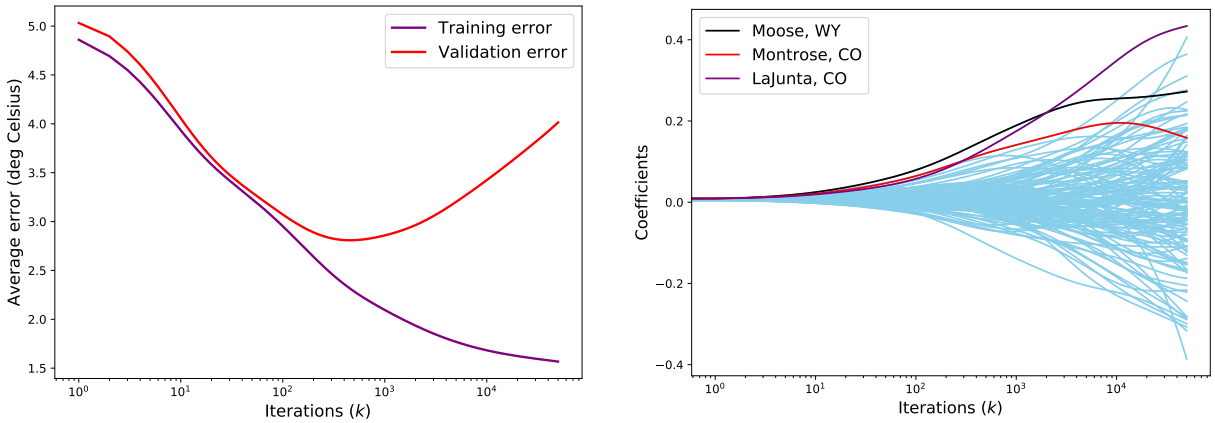

Figure 10: The left graph shows the training and validation errors of the gradient-descent estimator applied to the temperature-prediction task as the iterations progress. The number of training data is fixed to $n = 200$ training data. The right figure shows the values of the corresponding model coefficients. 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.

value of k should optimize the bias-variance tradeoff, as in ridge regression. Figure 9 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.

Example 2.5 (Temperature prediction via gradient descent with early stopping)**.** We apply gradient descent to minimize the least-squares cost function for the data in Example 2.2 of the notes on linear regression. 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 10 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 11 shows the number of iterations selected for different numbers of training data based on validation error. Figure 12 shows the corresponding coefficients and compares them the OLS coefficients. The effect achieved by early stopping is reminiscent of ridge regression. Figure 13 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. △

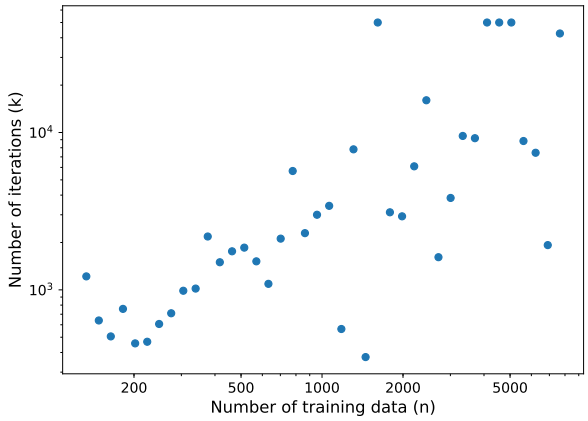

Figure 11: Results of selecting the number of iterations via cross-validation for the experiment described in Example 2.5. The image shows the number of iterations at which the gradient-descent estimator achieves minimum validation error for different numbers of training data. The maximum number of iterations was limited to 10^5.

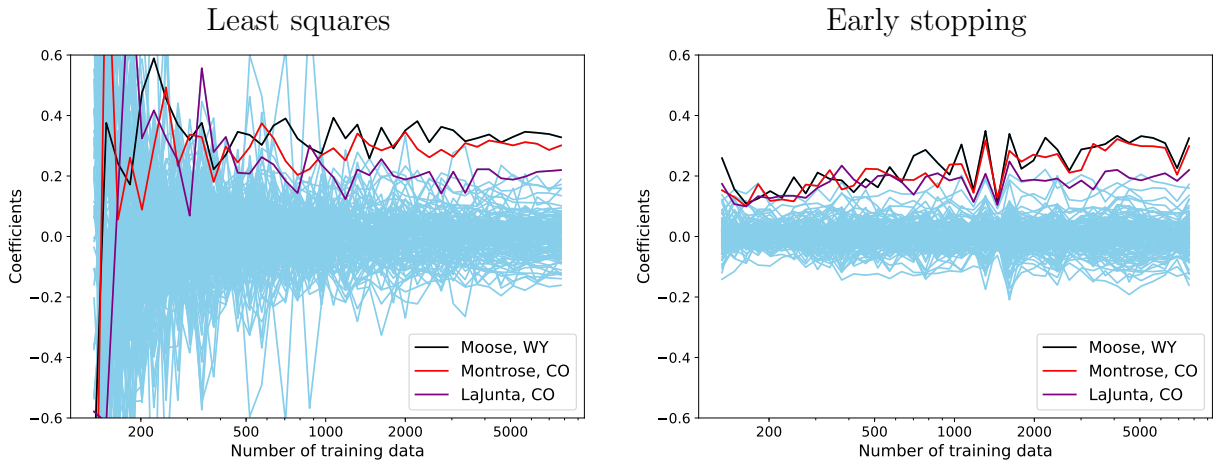

Figure 12: Coefficients of the least-squares (left) and gradient-descent (right) estimators for the experiment described in Example 2.5 for different values of training data. All coefficients are depicted in light blue except the three that have the largest magnitudes for large n, which correspond to the stations of Moose in Wyoming, and Montrose and La Junta in Colorado.

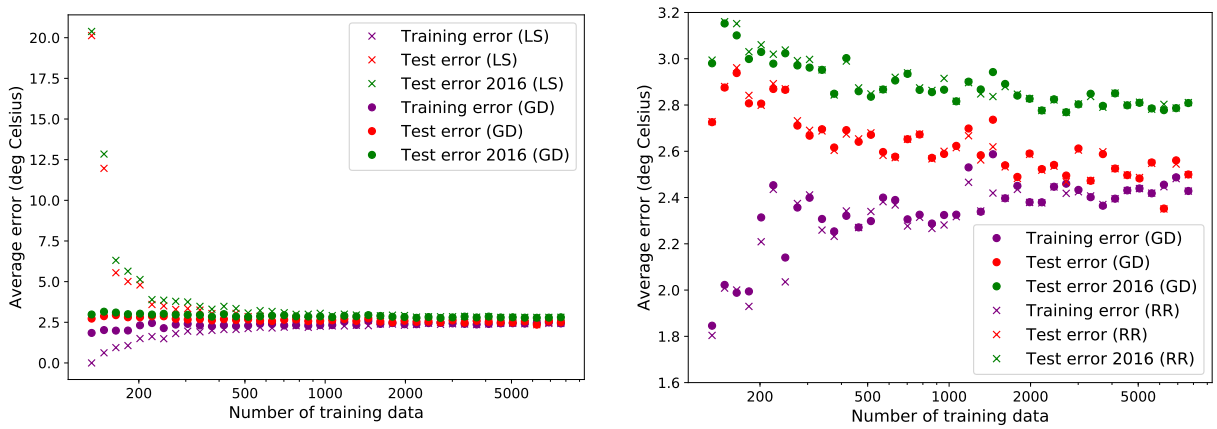

Figure 13: Performance of the gradient-descent estimator for the experiment described in Example 2.5. The left image compares the method to the least-squares estimator on the training and test sets, and on the 2016 data, for different number of training data. The right image shows the same comparison to the ridge-regression estimator.