Sparse regression

1 Linear regression

In statistics, the problem of regression is that of learning a function that allows to estimate a certain quantity of interest, the *response* or *dependent variable*, from several observed variables, known as *covariates*, *features* or *independent variables*. For example, we might be interested in estimating the price of a house based on its extension, the number of rooms, the year it was built, etc. The function that models the relation between the response and the predictors is learnt from training data and can then be used to predict the response for new examples.

In linear regression, we assume that the response is well modeled as a linear combination of the predictors. The model is parametrized by an intercept $\beta_0 \in \mathbb{R}$ and a vector of weights $\beta \in \mathbb{R}^p$, where p is the number of predictors. Let us assume that we have n data points consisting of a value of the response y_i and the corresponding values of the predictors X_{i1}, X_{i2}, ..., X_{ip}, $1 \leq i \leq n$. The linear model is given by

$$y_i \approx \beta_0 + \sum_{j=1}^{p} \beta_j X_{ij}, \quad 1 \leq i \leq n, \tag{1}$$

$$\begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_n \end{bmatrix} \approx \begin{bmatrix} 1 \\ 1 \\ \dots \\ 1 \end{bmatrix} \beta_0 + \begin{bmatrix} X_{11} & X_{12} & \cdots & X_{1p} \\ X_{21} & X_{22} & \cdots & X_{2p} \\ & & \cdots & \\ X_{n1} & X_{n2} & \cdots & X_{np} \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \dots \\ \beta_p \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ \dots \\ 1 \end{bmatrix} \beta_0 + X\beta. \tag{2}$$

We have already encountered linear models that arise in inverse problems such as compressed sensing or super-resolution. A major difference between statistics and those applications is that in statistics the ultimate aim is to predict y accurately for new values of the predictors, *not* to estimate β. The role of β is merely to quantify the linear relation between the response and the predictors. In contrast, when solving an inverse problems the main objective is to determine β, which has a physical interpretation: an image of a 2D slice of a body in MRI, the spectrum of multisinusoidal signal in spectral super-resolution, reflection coefficients of strata in seismography, etc.

1.1 Least-squares estimation

To calibrate the linear regression model, we estimate the weight vector from the training data. By far the most popular method to compute the estimate is to minimize the ℓ_2 norm

of the fitting error on the training set. In more detail, the weight estimate β_{ls} is the solution of the least-squares problem

$$\text{minimize} \quad \left|\left| y - X\tilde{\beta} \right|\right|_2 . \tag{3}$$

The least-squares cost function is convenient from a computational view, since it is convex and can be minimized efficiently (in fact, as we will see in a moment it has a closed-form solution). In addition, as detailed in Proposition 1.3 below, it has a reasonable probabilistic interpretation.

The following proposition, proved in Section 1.1, shows that the solution to the least-squares problem has a closed form solution.

Proposition 1.1 (Least-squares solution)**.** *For $p \geq n$, if X is full rank then the solution to the least-squares problem* (3) *is*

$$\beta_{\text{ls}} := \left(X^T X\right)^{-1} X^T y. \tag{4}$$

A corollary to this result provides a geometric interpretation for the least-squares estimate of y: it is obtained by projecting the response onto the column space of the matrix formed by the predictors.

Corollary 1.2. *For $p \geq n$, if X is full rank then $X\beta_{\text{ls}}$ is the projection of y onto the column space of X.*

Proof. Let $X = U\Sigma V^T$ be the singular-value decomposition of X. Since X is full rank and $p \geq n$ we have $U^T U = I$, $V^T V = I$ and Σ is a square invertible matrix, which implies

$$\begin{aligned} X\beta_{\text{ls}} &= X\left(X^T X\right)^{-1} X^T y & (5)\\ &= U\Sigma V^T \left(V\Sigma U^T U \Sigma V^T\right) V\Sigma U^T y & (6)\\ &= UU^T y. & (7) \end{aligned}$$

□

Proposition 1.3 (Least-squares solution as maximum-likelihood estimate)**.** *If we model y as*

$$y = X\beta + z \tag{8}$$

where $X \in \mathbb{R}^{n\times p}$, $p \geq n$, $\beta \in \mathbb{R}^p$ and $y \in \mathbb{R}^n$ are fixed and the entries of $z \in \mathbb{R}^n$ are iid Gaussian random variables with mean zero and the same variance, then the maximum-likelihood estimate of β given y is equal to β_{ls}.

The proposition is proved in Section A.2 of the appendix.

1.2 Preprocessing

The careful reader might notice that we have not explained how to fit the intercept β_0. Before fitting a linear regression model, we typically perform the following preprocessing steps.

1. Centering each predictor column X_i subtracting its mean

$$\mu_j := \sum_{i=1}^{n} X_{ij} \tag{9}$$

   so that each column of X has mean zero.

2. Normalizing each predictor column X_j by dividing by

$$\sigma_j := \sqrt{\sum_{i=1}^{n} (X_{ij} - \mu_j)^2} \tag{10}$$

   so that each column of X has ℓ_2 norm equal to one. The objective is to make the estimate invariant to the units used to measure each predictor.

3. Centering the response vector y by subtracting its mean

$$\mu_y := \sum_{i=1}^{n} y_i. \tag{11}$$

By the following lemma, the intercept of the linear model once we have centered the data is equal to zero.

Lemma 1.4 (Intercept)**.** *If the mean of y and of each of the columns of X is zero, then the intercept in the least-squares solution is also zero.*

The lemma is proved in Section A.3 of the appendix.

Once have solved the least-squares problem using the centered and normalized data to obtain β_{ls}, we can use the model to estimate the response corresponding to a new vector of predictors $x \in \mathbb{R}^p$ by computing

$$f(x) := \mu_y + \sum_{i=1}^{p} \beta_{\text{ls},i} \, \frac{x_i - \mu_i}{\sigma_i}. \tag{12}$$

1.3 Overfitting

Imagine that a friend tells you:

I found a cool way to predict the temperature in New York: It's just a linear combination of the temperature in every other state. I fit the model on data from the last month and a half and it's perfect!

Your friend is not lying, but the problem is that she is using a number of data points to fit the linear model that is roughly the same as the number of parameters. If $n = p$ we can find a β such that $y = X\beta$ exactly, even if y and X have nothing to do with each other! This is called overfitting and is usually caused by using a model that is too flexible with respect to the number of data that are available.

To evaluate whether a model suffers from overfitting we separate the data into a training set and a test set. The training set is used to fit the model and the test set is used to evaluate the error. A model that overfits the training set will have a very low error when evaluated on the training examples, but will not generalize well to the test examples.

Figure 1 shows the result of evaluating the training error and the test error of a linear model with $p = 50$ parameters fitted from n training examples. The training and test data are generated by fixing a vector of weights β and then computing

$$y_{\text{train}} = X_{\text{train}}\,\beta + z_{\text{train}}, \tag{13}$$

$$y_{\text{test}} = X_{\text{test}}\,\beta, \tag{14}$$

where the entries of X_{train}, X_{test}, z_{train} and β are sampled independently at random from a Gaussian distribution with zero mean and unit variance. The training and test errors are defined as

$$\text{error}_{\text{train}} = \frac{||X_{\text{train}}\,\beta_{\text{ls}} - y_{\text{train}}||_2}{||y_{\text{train}}||_2}, \tag{15}$$

$$\text{error}_{\text{test}} = \frac{||X_{\text{test}}\,\beta_{\text{ls}} - y_{\text{test}}||_2}{||y_{\text{test}}||_2}. \tag{16}$$

Note that even the true β does not achieve zero training error because of the presence of the noise, but the test error is actually zero if we manage to estimate β exactly.

The training error of the linear model grows with n. This makes sense as the model has to fit more data using the same number of parameters. When n is close to p (50), the fitted model is much better than the true model at replicating the training data (the error of the true model is shown in green). This is a sign of overfitting: the model is adapting to the noise and not learning the true linear structure. Indeed, in that regime the test error is extremely high. At larger n, the training error rises to the level achieved by the true linear model and the test error decreases, indicating that we are learning the underlying model.

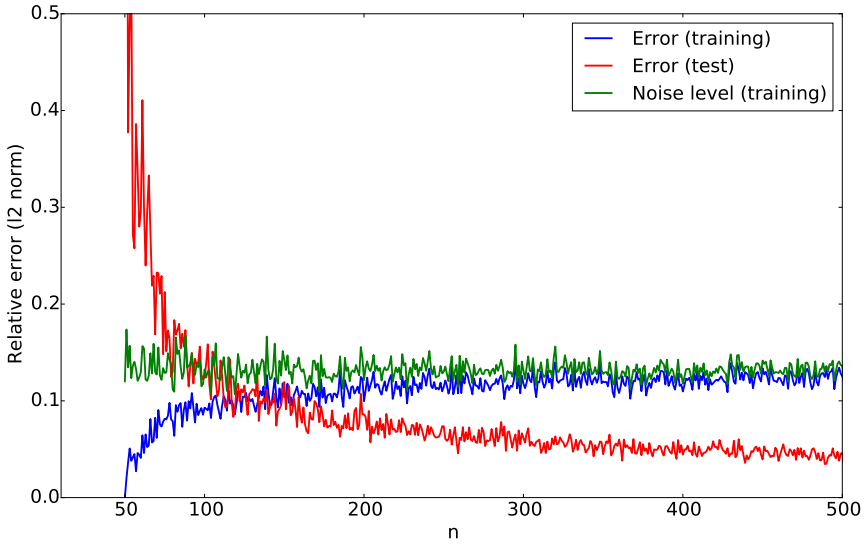


Figure 1: Relative ℓ_2-norm error in estimating the response achieved using least-squares regression for different values of n (the number of training data). The training error is plotted in blue, whereas the test error is plotted in red. The green line indicates the training error of the true model used to generate the data.

1.4 Theoretical analysis of linear regression

In this section we analyze the solution of the least-squares regression fit to understand its dependence with the number of training examples. The following theorem, proved in Section A.4 of the appendix, characterizes the error in estimating the weights under the assumption that the data are indeed generated by a linear model with additive noise.

Theorem 1.5. *Assume the data y are generated according to a linear model with additive noise,*

$$y = X\beta + z, \tag{17}$$

where $X \in \mathbb{R}^{n \times p}$ and $\beta \in \mathbb{R}^p$, and that the entries of $z \in \mathbb{R}^n$ are drawn independently at random from a Gaussian distribution with zero mean and variance σ_z^2. The least-squares estimate

$$\beta_{\text{ls}} := \arg\min_{\tilde{\beta}} \left|\left| y - X\tilde{\beta} \right|\right|_2 \tag{18}$$

satisfies

$$\frac{p\,\sigma_z^2\,(1-\epsilon)}{\sigma_{\max}^2} \leq ||\beta - \beta_{\text{ls}}||_2^2 \leq \frac{p\,\sigma_z^2\,(1+\epsilon)}{\sigma_{\min}^2} \tag{19}$$

with probability $1 - 2\exp\left(-\frac{p\epsilon^2}{8}\right)$. $\sigma_{\min}$ *and* $\sigma_{\max}$ *denote the smallest and largest singular value of* X *respectively.*

The bounds in the theorem are in terms of the singular values of the predictor matrix $X \in \mathbb{R}^{n \times p}$. To provide some intuition as to the scaling of these singular values when we fix p and increase n, let us assume that the entries of X are drawn independently at random from a standard Gaussian distribution. Then by Proposition 3.4 in Lecture Notes 5 both $\sigma_{\min}$ and $\sigma_{\max}$ are close to $\sqrt{n}$ with high probability as long as $n > C\,p$ for some constant C. This implies that if the variance of the noise z equals one,

$$||\beta - \beta_{\text{ls}}||_2 \approx \sqrt{\frac{p}{n}}. \tag{20}$$

If each of the entries of β has constant magnitude the ℓ_2 norm of β is approximately equal to $\sqrt{p}$. In this case, the theoretical analysis predicts that the normalized error

$$\frac{||\beta - \beta_{\text{ls}}||_2}{||\beta||_2} \approx \frac{1}{\sqrt{n}}. \tag{21}$$

Figure 2 shows the result of a simulation where the entries of X, β and z are all generated by sampling independently from standard Gaussian random variables. The relative error scales precisely as $1/\sqrt{n}$.

For a fixed number of predictors, this analysis indicates that the least-squares solution converges to the true weights as the number of data n grows. In statistics jargon, the estimator is *consistent*. The result suggests two scenarios in which the least-squares estimator may not yield a good estimate: when p is of the same order as n and when some of the predictors are highly correlated, as some of the singular values of X may be very small.

1.5 Global warming

In this section we describe the application of linear regression to climate data. In particular, we analyze temperature data taken in a weather station in Oxford over 150 years.[1] Our objective is not to perform prediction, but rather to determine whether temperatures have risen or decreased during the last 150 years in Oxford.

In order to separate the temperature into different components that account for seasonal effects we use a simple linear with three predictors and an intercept

$$y \approx \beta_0 + \beta_1 \cos\left(\frac{2\pi t}{12}\right) + \beta_2 \cos\left(\frac{2\pi t}{12}\right) + \beta_3\, t \tag{22}$$

[1]The data is available at `http://www.metoffice.gov.uk/pub/data/weather/uk/climate/stationdata/oxforddata.txt`.

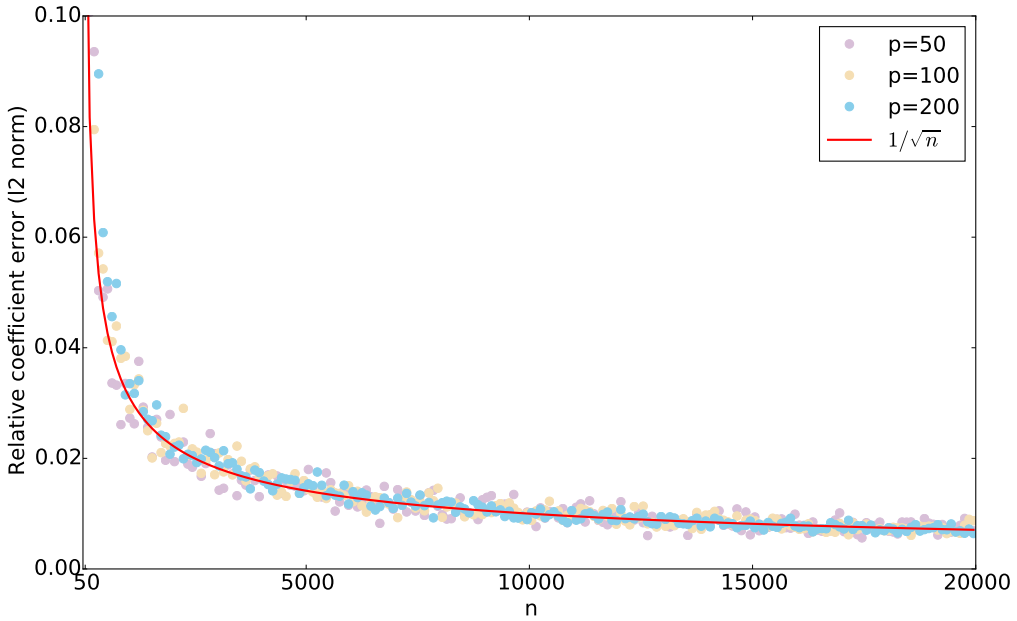


Figure 2: Relative ℓ_2-norm error of the least-squares estimate as n grows. The entries of X, β and z are all generated by sampling independently from standard Gaussian random variables. The simulation is consistent with 21.

where t denotes the time in months. The corresponding matrix of predictors is

$$X := \begin{bmatrix} 1 & \cos\left(\frac{2\pi t_1}{12}\right) & \sin\left(\frac{2\pi t_1}{12}\right) & t_1 \\ 1 & \cos\left(\frac{2\pi t_2}{12}\right) & \sin\left(\frac{2\pi t_2}{12}\right) & t_2 \\ \dots & \dots & \dots & \dots \\ 1 & \cos\left(\frac{2\pi t_n}{12}\right) & \sin\left(\frac{2\pi t_n}{12}\right) & t_n \end{bmatrix}. \tag{23}$$

The intercept β_0 represents the mean temperature, β_1 and β_2 account for periodic yearly fluctuations and β_3 is the overall trend. If β_3 is positive then the model indicates that temperatures are increasing, if it is negative then it indicates that temperatures are decreasing.

The results of fitting the linear model are shown in Figures 3 and 4. The fitted model indicates that both the maximum and minimum temperatures have an increasing trend of about 0.8 degrees Celsius (around 1.4 degrees Fahrenheit).

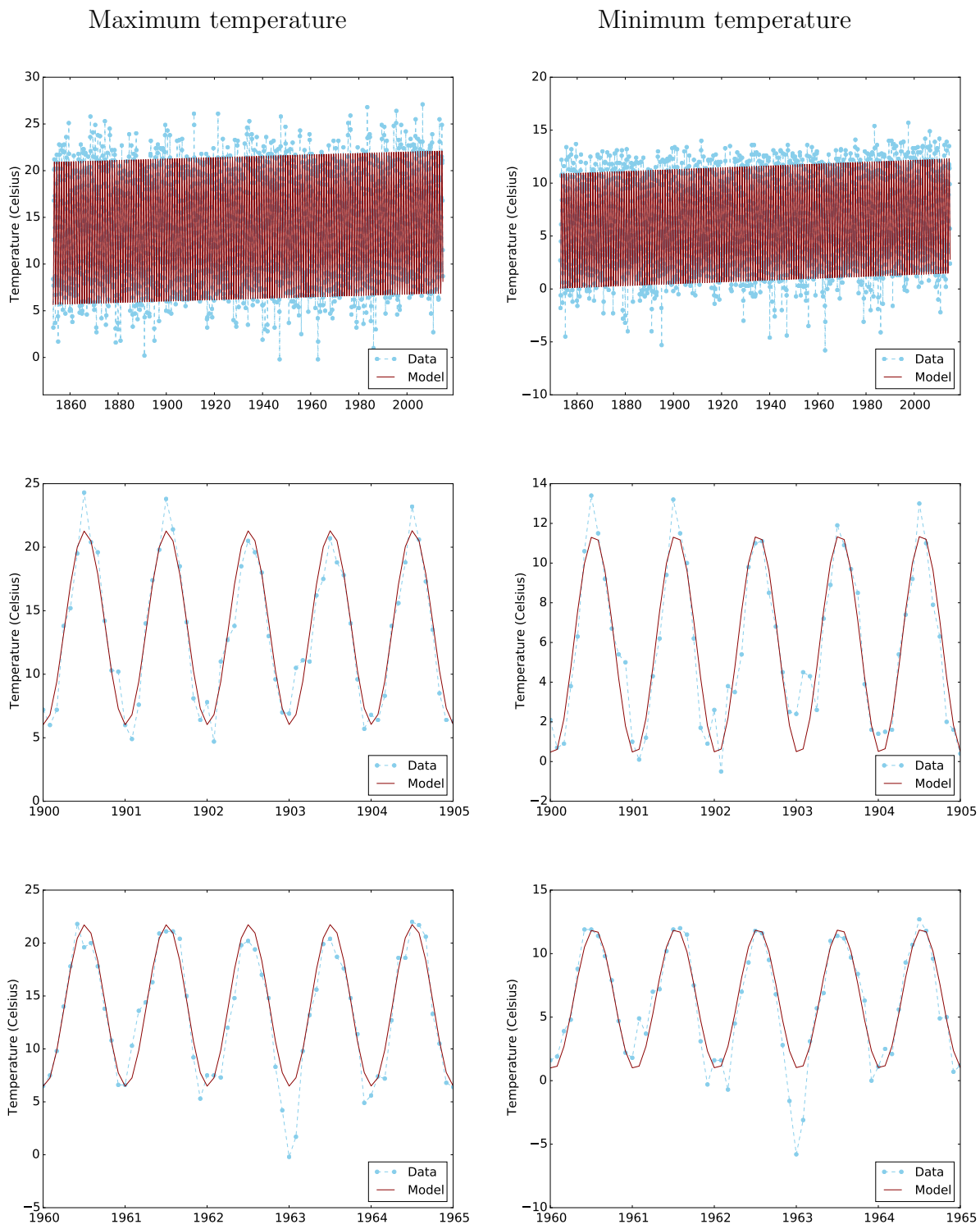
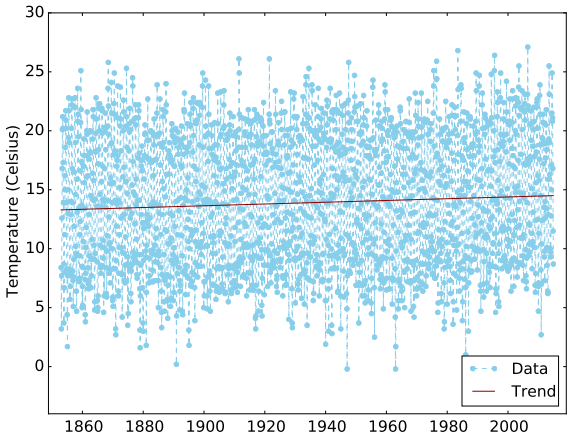


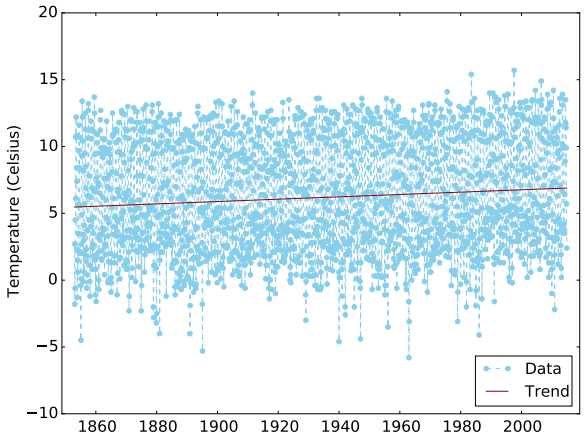
Figure 3: Temperature data together with the linear model described by (22) for both maximum and minimum temperatures.

Maximum temperature



+ 0.75 °C / 100 years

Minimum temperature



+ 0.88 °C / 100 years

Figure 4: Temperature trend obtained by fitting the model described by (22) for both maximum and minimum temperatures.

1.6 Logistic regression

The problem of classification in statistics and machine learning is very related to regression. The only difference is that in classification the response is binary: it is equal to either 0 or 1. Of course, the labels 0 and 1 are arbitrary, they represent two distinct classes into which we aim to classify our data. For example, the predictors might be pictures and the two classes *cats* and *dogs*.

In linear regression, we use a linear model to predict the response from the predictors. In logistic regression, we use a linear model to predict *how likely* it is for the response to equal 0 or 1. This requires mapping the output of the linear model to the $[0, 1]$ interval, which is achieved by applying the logistic function or sigmoid,

$$g(t) := \frac{1}{1 + \exp -t}, \tag{24}$$

to a linear combination of the predictors. In more detail, we model the probability that $y_i = 1$ by

$$\mathrm{P}(y_i = 1 | X_{i1}, X_{i2}, \ldots, X_{ip}) \approx g\left(\beta_0 + \sum_{j=1}^{p} \beta_j X_{ij}\right) \tag{25}$$

$$= \frac{1}{1 + \exp\left(-\beta_0 - \sum_{j=1}^{p} \beta_j X_{ij}\right)}. \tag{26}$$

In Proposition 1.3 we established that least-squares fitting computes a maximum-likelihood estimate in the case of linear models with additive Gaussian noise. The following proposition derives a cost function to calibrate a logistic-regression model by maximizing the likelihood under the assumption that the response is a Bernouilli random variable parametrized by the linear model.

Proposition 1.6 (Logistic-regression cost function)**.** *If we assume that the response values $y_1, y_2, \ldots, y_n$ in the training data are independent samples from Bernouilli random variables $\check{y}_1, \check{y}_2, \ldots, \check{y}_n$ such that*

$$\mathrm{P}\left(\check{y}_i = 1 | X_{i1}, X_{i2}, \ldots, X_{ip}\right) := g\left(\beta_0 + \sum_{j=1}^{p} \beta_j X_{ij}\right), \tag{27}$$

$$\mathrm{P}\left(\check{y}_i = 0 | X_{i1}, X_{i2}, \ldots, X_{ip}\right) := 1 - g\left(\beta_0 + \sum_{j=1}^{p} \beta_j X_{ij}\right), \tag{28}$$

then the maximum-likelihood estimate of the intercept and the weights β_0 and β_0 are obtained by maximizing the function

$$\log \mathcal{L}\left(\tilde{\beta}_0, \tilde{\beta}\right) := \sum_{i=1}^{n} y_i \log g\left(\tilde{\beta}_0 + \sum_{j=1}^{p} \tilde{\beta}_j X_{ij}\right) + (1 - y_i) \log\left(1 - g\left(\tilde{\beta}_0 + \sum_{j=1}^{p} \tilde{\beta}_j X_{ij}\right)\right). \tag{29}$$

Proof. Due to the independence assumption, the joint probability mass function (pmf) of the random vector $\check{y}$ equals

$$p_{\check{y}}(y) := \prod_{i=1}^{n} g\left(\tilde{\beta}_0 + \sum_{j=1}^{p} \tilde{\beta}_j X_{ij}\right)^{y_i} \left(1 - g\left(\tilde{\beta}_0 + \sum_{j=1}^{p} \tilde{\beta}_j X_{ij}\right)\right)^{1-y_i}. \tag{30}$$

The likelihood is defined as the joint pmf parametrized by the weight vectors,

$$\mathcal{L}\left(\tilde{\beta}_0, \tilde{\beta}\right) := \prod_{i=1}^{n} g\left(\tilde{\beta}_0 + \sum_{j=1}^{p} \tilde{\beta}_j X_{ij}\right)^{y_i} \left(1 - g\left(\tilde{\beta}_0 + \sum_{j=1}^{p} \tilde{\beta}_j X_{ij}\right)\right)^{1-y_i}. \tag{31}$$

Taking the logarithm of this nonnegative function completes the proof. □

The log-likelihood function is strictly concave, so the logistic-regression estimate is well defined. Although the cost function is derived by assuming that the data follow a certain probabilistic model, logistic regression is widely deployed in situations where the probabilistic assumptions do not hold. The model will achieve high prediction accuracy on any dataset where the predictors are linearly separable, as long as sufficiently data is available.

2 Sparse regression

2.1 Model selection

In Section 1.4 we establish that linear regression allows to learn a linear model when the number of available examples n is large with respect to the number of predictors p. However, in many modern applications, the number of predictors can be extremely large. An example is computational genomics, where the predictors may correspond to gene-expression measurements from thousands of genes, whereas n is the number of patients which might only be in the hundreds.

It is obviously impossible to fit a linear model when $p > n$, or even when $p \approx n$ without overfitting (depending on the noise level), but it may still be possible to fit a sparse linear model that only depends on a subset of $s < p$ predictors. Selecting the relevant predictors to include in the model is called *model selection* in statistics.

2.2 Best subset selection and forward stepwise regression

A possible way to select a small number of relevant predictors from a training set is to fix the order of the sparse model $s < p$ and then evaluate the least-squares fit of all possible s-sparse models in order to select the one that provides the best fit. This is called the *best-subset selection* method. Unfortunately it is computationally intractable even for small values of s and p. For instance, there are more than 10^{13} possible models if $s = 10$ and $p = 100$.

An alternative to an exhaustive evaluation of all possible sparse models is to select the predictors greedily in the spirit of signal-processing methods such as orthogonal matching pursuit. In *forward stepwise regression* we select the predictor that is most correlated with the response and then project the rest of predictors onto its orthogonal complement. Iterating this procedure allows to learn an s-sparse model in s steps.

Algorithm 2.1 (Forward stepwise regression)**.** *Given a matrix of predictors $X \in \mathbb{R}^{n \times p}$ and a response $y \in \mathbb{R}^n$, we initialize the residual and the subset of relevant predictors $\mathcal{S}$ by setting,*

$$j_0 := \arg\max_j |\langle y, X_j \rangle| \tag{32}$$

$$\mathcal{S}_0 := \{j_0\} \tag{33}$$

$$\beta_{\text{ls}} := \arg\min_{\tilde{\beta}} \left|\left| y - X_{\mathcal{S}_0} \tilde{\beta} \right|\right|_2 \tag{34}$$

$$r^{(0)} := y - X_{\mathcal{S}_0} \beta_{\text{ls}}. \tag{35}$$

Then for $k = 2, 3, \ldots, s$ we compute

$$j_k := \arg\max_{j \notin \mathcal{S}_{j-1}} \left| \left\langle y, \mathcal{P}_{\mathrm{col}\left(X_{\mathcal{S}_{j-1}}\right)^{\perp}} X_j \right\rangle \right| \tag{36}$$

$$\mathcal{S}_j := \mathcal{S}_{j-1} \cup \{j_k\} \tag{37}$$

$$\beta_{\mathrm{ls}} := \arg\min_{\tilde{\beta}} \left\| y - X_{\mathcal{S}_j} \tilde{\beta} \right\|_2 \tag{38}$$

$$r^{(k)} := r^{(k-1)} - X_{\mathcal{S}_j} \beta_{\mathrm{ls}}. \tag{39}$$

The algorithm is very similar to orthogonal matching pursuit. The only difference is the orthogonalization step in which we project the remaining predictors onto the orthogonal complement of the span of the predictors that have been selected already.

2.3 The lasso

Fitting a sparse model that uses a subset of the available predictors is equivalent to learning a weight vector β that only contains a small number of nonzeros. As we saw in the case of sparse signal representations and underdetermined inverse problems, penalizing the ℓ_1 norm is an efficient way of promoting sparsity. In statistics, ℓ_1-norm regularized least squares is known as the *lasso*,

$$\text{minimize} \quad \frac{1}{2n} \left\| y - \tilde{\beta}_0 - X\tilde{\beta} \right\|_2^2 + \lambda \left\| \tilde{\beta} \right\|_1 \tag{40}$$

$$\tag{41}$$

$\lambda > 0$ is a regularization parameter that controls the tradeoff between the fit to the data and the ℓ_1 norm of the weights. Figure 5 shows the values of the coefficients obtained by the lasso for a linear regression problem with 50 predictors where the response only depends on 3 of them. If λ is very large, all coefficients are equal to zero. If λ is very small, then the lasso estimate is equal to the least-squares estimate. For values in between, the lasso yields a sparse model containing the coefficients corresponding to the relevant predictors.

A different formulation for the lasso, which is the one that appeared in the original paper [4], incorporates a constraint on the ℓ_1 norm of the weight vector, instead of an additive term.

$$\text{minimize} \quad \left\| y - X\tilde{\beta} - \tilde{\beta}_0 \right\|_2^2 \tag{42}$$

$$\text{subject to} \quad \left\| \tilde{\beta} \right\|_1 \leq \tau \tag{43}$$

The two formulations are equivalent, but the relation between λ and τ depends on X and y.

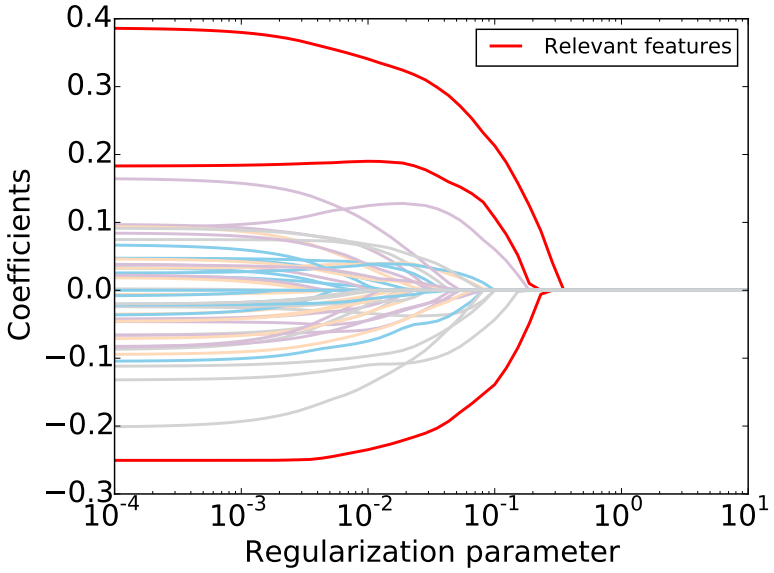


Figure 5: Magnitude of the lasso coefficients for different value of the regularization parameter. The number of predictors is 50, but the response only depends on 3 of them, which are marked in red.

To compare the performance of the lasso, forward stepwise regression and least-squares regression on a dataset that follows a sparse linear model we generate simulated data by computing

$$y_{\text{train}} = X_{\text{train}}\,\beta + z_{\text{train}}, \tag{44}$$

$$y_{\text{test}} = X_{\text{test}}\,\beta, \tag{45}$$

where the entries of X_{train}, X_{test} and z_{train} and β are sampled independently at random from a Gaussian distribution with zero mean and unit variance. 10 entries of β are also sampled iid from a standard normal, but the rest are set to zero. As a result the response only depends on 10 predictors out of a total of 50. The training and test errors were computed as in (15) and (16).

Figure 6 shows the results for different values of n (to be clear we compute the estimates once at each value of n). As expected, the least-squares estimator overfits the training data and performs very badly on the test set for small values of n. In contrast, the lasso and forward stepwise regression do not overfit and achieve much smaller errors on the test set, even when n is equal to p.

In our implementation of forward stepwise regression we set the number of predictors in the sparse model to the true number. On real data we would need to also estimate the order of the model. The greedy method performs very well in some instances because it manages to

select the correct sparse model. However, in other cases, mistakes in selecting the relevant predictors produce high fit errors, even on the training set. In contrast, the lasso achieves accurate fits for every value of n.

2.4 Theoretical analysis of the lasso

Assume that we have data that follow a sparse linear model of the form

$$y = X\beta + z \tag{46}$$

where the weight vector β is sparse, so that the response y only depends on $s < p$ predictors. By Theorem 1.5, if the noise has variance σ^2, least-squares regression yields an estimate of the weight vector that satisfies

$$||\beta - \beta_{\text{ls}}||_2 \approx \sigma_z \sqrt{\frac{p}{n}}. \tag{47}$$

as long as the matrix of predictors X is well conditioned and has entries with constant amplitudes. When p is close to n this indicates that least-squares regression does not yield a good estimate.

In order to characterize the error achieved by the lasso, we introduce the restricted-eigenvalue property (REP), which is similar to the restricted-isometry property (RIP) that we studied in Lecture Notes 5.

Definition 2.2 (Restricted-eigenvalue property)**.** *A matrix $M \in \mathbb{R}^{n \times p}$ satisfies the restricted-eigenvalue property with parameter s if there exists $\gamma > 0$ such that for any $v \in \mathbb{R}^p$ if*

$$||v_{T^c}||_1 \leq ||v_T||_1 \tag{48}$$

for any subset T with cardinality s then

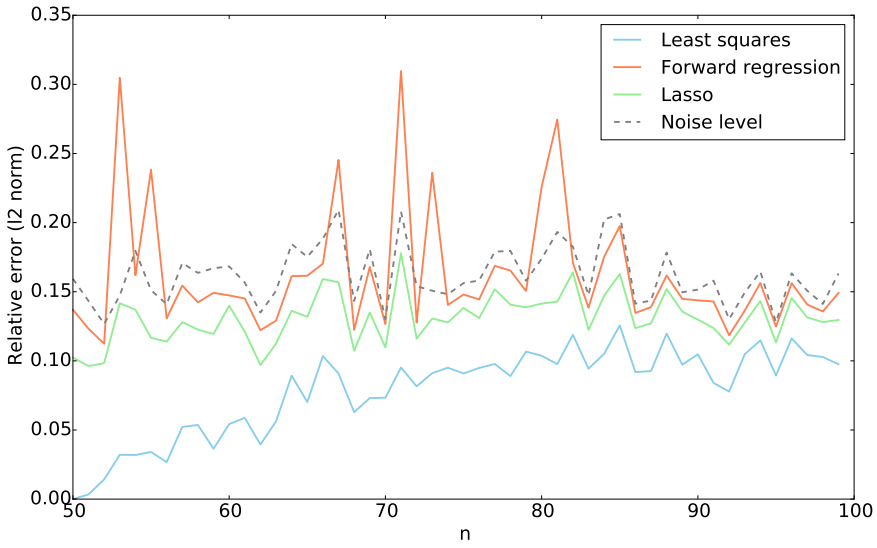
$$\frac{1}{n} ||M\ v||_2^2 \geq \gamma\, ||v||_2^2 . \tag{49}$$

Just like the RIP, the REP states that the matrix preserves the norm of a certain class of vectors. In this case, the vectors are not necessarily sparse, but rather have ℓ_1-norm concentrated on a sparse subset of the entries, which can be interpreted as a robustified notion of sparsity. The property may hold even if $p > n$, i.e. when we have more predictors than examples. The following theorem provides guarantees for the lasso under the REP.

Theorem 2.3. *Assume that the data y are generated according to a linear model with additive noise,*

$$y = X\beta + z, \tag{50}$$

Training



Test

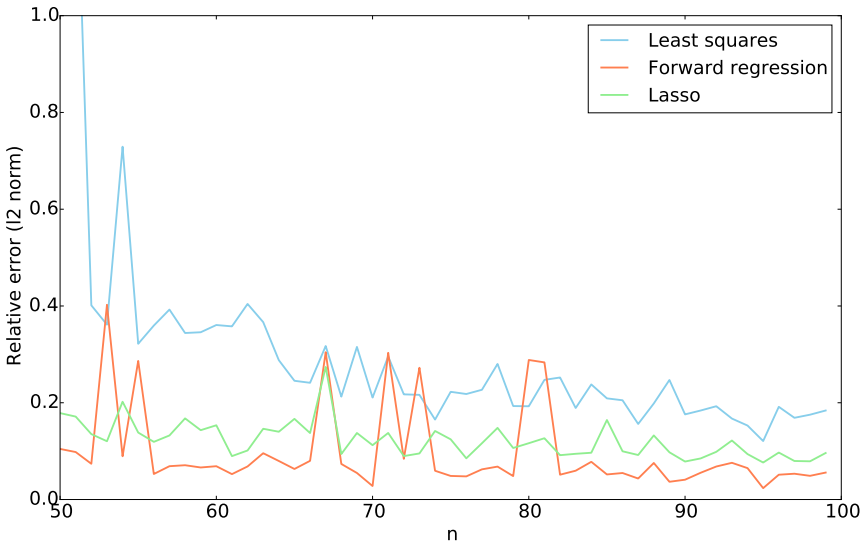


Figure 6: Comparison of the training and test error of the lasso, forward stepwise regression and least-squares regression for simulated data where the number of predictors is equal to 50 but only 10 are used to generate the response.

where $X \in \mathbb{R}^{n \times p}$ and $\beta \in \mathbb{R}^p$, and that the entries of $z \in \mathbb{R}^n$ are drawn independently at random from a Gaussian distribution with zero mean and variance σ_z^2. If β has s nonzero entries and X satisfies the restricted-eigenvalue property, the solution β_{lasso} to

$$\text{minimize} \quad \left|\left| y - X\tilde{\beta} \right|\right|_2^2 \tag{51}$$

$$\text{subject to} \quad \left|\left| \tilde{\beta} \right|\right|_1 \leq \tau \tag{52}$$

if we set $\tau := ||\beta||_1$ satisfies

$$||\beta - \beta_{\text{lasso}}||_2 \leq \frac{\sigma_z \sqrt{32\ \alpha\ s \log p}}{\gamma\ n} \max_i ||X_i||_2 \tag{53}$$

with probability $1 - 2\exp\left(-\left(\alpha - 1\right)\log p\right)$ for any $\alpha > 2$.

The result establishes that the lasso achieves an error that scales as $\sigma_z\sqrt{s/n}$, which is the same rate achieved by least squares if the true sparse model is known!

In this section we have focused on the estimation of the weight vector. This is important for model selection, as the sparsity pattern and the amplitude of the weights reveals the sparse model used to predict the response. However, in statistics the main aim is often to predict the response. For this purpose, in principle, conditions such as the REP should not be necessary. For results on the prediction error of the lasso we refer the interested reader to Chapter 11 of [3]. In Section 3 we discuss the performance of the lasso and related methods when the predictor matrix does not satisfy the REP.

2.5 Sparse logistic regression

An advantage of the lasso over greedy methods to learn sparse linear models is that it can be easily applied to logistic regression. All we need to do is add an ℓ_1-norm regularization term to the cost function derived in Section 1.6. In detail, to learn a sparse logistic-regression model we minimize the function

$$-\sum_{i=1}^{n} y_i \log g\left(\tilde{\beta}_0 + \sum_{j=1}^{p} \tilde{\beta}_j X_{ij}\right) - (1 - y_i) \log\left(1 - g\left(\tilde{\beta}_0 + \sum_{j=1}^{p} \tilde{\beta}_j X_{ij}\right)\right) + \lambda \left|\left| \tilde{\beta} \right|\right|_1.$$

This version of the lasso can be used to obtain a sparse logistic model for prediction of binary responses. To illustrate this, we consider a medical dataset[2]. The response indicates whether 271 patients suffer from arrhythmia or not. The predictors contain information about each patient, such as age, sex, height and weight, as well as features extracted from

[2]The data can be found at `https://archive.ics.uci.edu/ml/datasets/Arrhythmia`

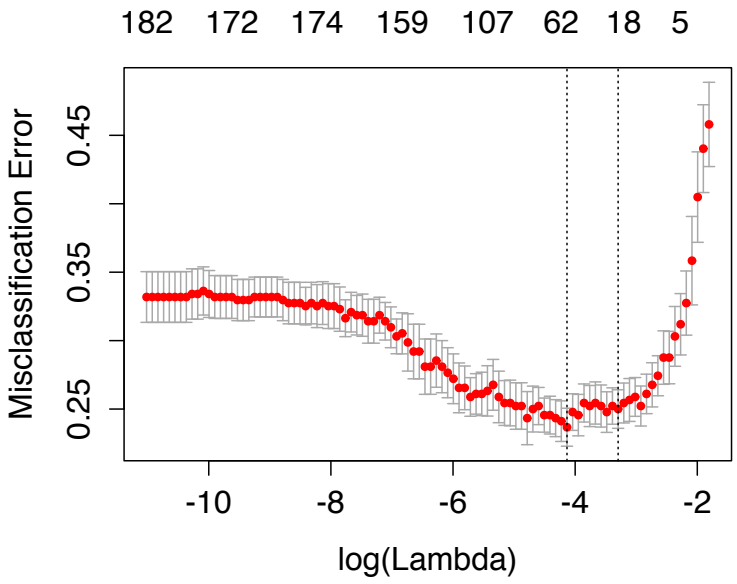


Figure 7: Distribution of misclassification errors achieved after repeatedly fitting the model using a random subset of 90% of the examples for different values of the regularization parameter λ. The number of predictors included in the sparse model is indicated above the graph.

electrocardiogram recordings. The total number of predictors is 182. We use the glmnet package in R [2] to fit the sparse logistic regression model.

Figure 7 shows the distribution of the test error achieved after repeatedly fitting the model using a random subset of 90% of the examples; a procedure known as *cross-validation* in statistics. The number of predictors included in the sparse model is indicated above the graph. The best results are obtained by a model containing 62 predictors, but a model containing 18 achieves very similar accuracy (both are marked with a dotted line).

3 Correlated predictors

In many situations, some of the predictors in a dataset may be highly correlated. As a result, the predictor matrix is ill conditioned, which is problematic for least squares regression, and also for the lasso. In this section, we discuss this issue and show how it can be tackled through regularization of the regression cost function.

3.1 Ridge regression

When the data in a linear regression problem is of the form $y = X\beta + z$, we can write the error of the least-squares estimator in terms of the singular-value decomposition of $X = U\Sigma V^T$,

$$||\beta - \beta_{\text{ls}}||_2 = \sqrt{\sum_{j=1}^{p} \left(\frac{U_j^T z}{\sigma_j} \right)^2}, \tag{54}$$

see (102) in Section 1.5. If a subset of the predictors are highly correlated, then some of the singular values will have very small values, which results in noise amplification. *Ridge regression* is an estimation technique that controls noise amplification by introducing an ℓ_2-norm penalty on the weight vector,

$$\text{minimize} \quad \left|\left| y - X\tilde{\beta} \right|\right|_2^2 + \lambda \left|\left| \tilde{\beta} \right|\right|_2^2, \tag{55}$$

where $\lambda > 0$ is a regularization parameter that controls the weight of the regularization term as in the lasso. In inverse problems, ℓ_2-norm regularization is often known as Tikhonov regularization.

The following proposition shows that, under the assumption that the data indeed follow a linear model, the error of the ridge-regression estimator can be decomposed into a term that depends on the signal and a term that depends on the noise.

Proposition 3.1 (Ridge-regression error)**.** *If $y = X\beta + z$ and $X \in \mathbb{R}^{n\times p}$, $n \geq p$, is full rank, then the solution of Problem* (55) *can be written as*

$$\beta_{\text{ridge}} = V \begin{bmatrix} \frac{\sigma_1^2}{\sigma_1^2+\lambda} & 0 & \cdots & 0 \\ 0 & \frac{\sigma_2^2}{\sigma_2^2+\lambda} & \cdots & 0 \\ & & \cdots & \\ 0 & 0 & \cdots & \frac{\sigma_p^2}{\sigma_p^2+\lambda} \end{bmatrix} V^T\beta + V \begin{bmatrix} \frac{\sigma_1}{\sigma_1^2+\lambda} & 0 & \cdots & 0 \\ 0 & \frac{\sigma_2}{\sigma_2^2+\lambda} & \cdots & 0 \\ & & \cdots & \\ 0 & 0 & \cdots & \frac{\sigma_p}{\sigma_p^2+\lambda} \end{bmatrix} U^T z. \tag{56}$$

We defer the proof to Section A.6 in the appendix.

Increasing the value of the regularization parameter λ allows to control the noise term when some of the predictors are highly correlated. However, this also increases the error term that depends on the original signal; if $z = 0$ then we don't recover the true weight vector β unless $\lambda = 0$. Calibrating the regularization parameter allows to adapt to the conditioning of the predictor matrix and the noise level in order to achieve a good tradeoff between both terms.

3.2 The elastic net

Figure 8 shows the coefficients of the lasso and ridge regression learnt from a dataset where the response follows a sparse regression model. The model includes 50 predictors but only 12 are used to generate the response. These 12 predictors are divided into two groups of 6 that are highly correlated[3].

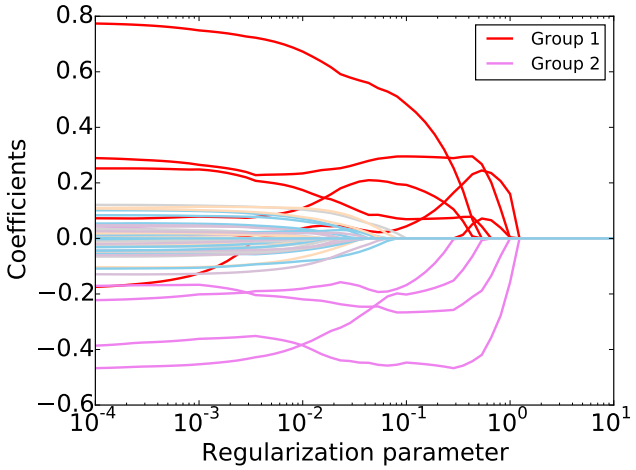
The model obtained using ridge regression assigns similar weights to the correlated variables. This is a desirably property since all of these predictors are equally predictive of the response value. However, the learnt model is not sparse for any value of the regularization parameter, which means that it selects all of the irrelevant predictors. In contrast, the lasso produces a sparse model, but the coefficients of the relevant predictors are very erratic. In fact, in the regime where the coefficients are sparse not all the relevant predictors are included in the model (two from the second group are missing).

The following lemma, proved in gives some intuition as to why the coefficient path for the lasso tends to be erratic when some predictors are highly correlated. When two predictors are exactly the same, then the lasso chooses arbitrarily between the two, instead of including both in the model with similar weights.

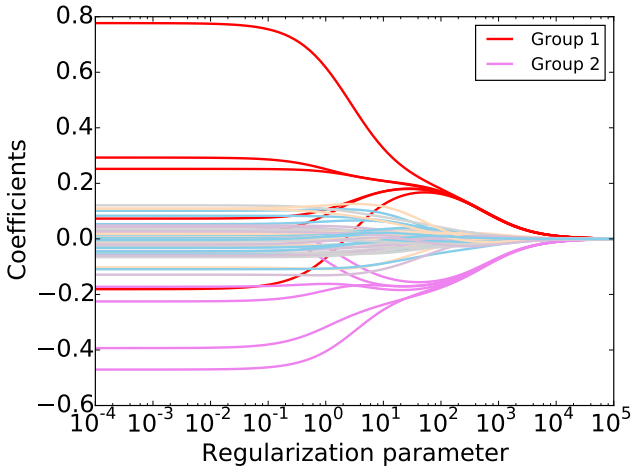
Lemma 3.2. *If two columns of the predictor matrix X are identical $X_i = X_j$, $i \neq j$, and*

[3]In more detail, the predictors in each group are sampled from a Gaussian distribution with zero mean and unit variance such that the covariance between each pair of predictors is equal to 0.95.

Lasso



Ridge regression



Elastic net ($\alpha = 0.5$)

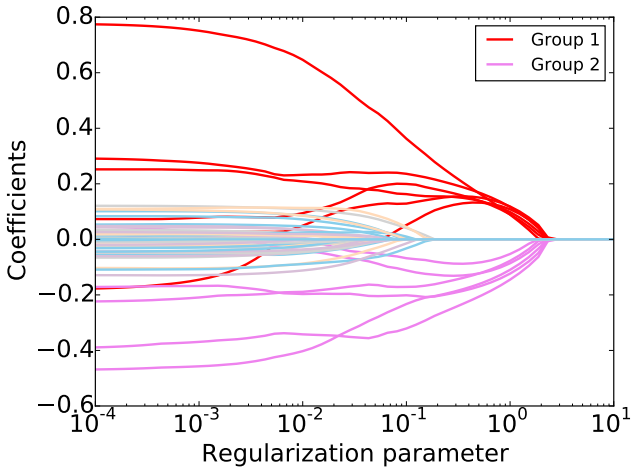


Figure 8: Coefficients of the lasso, ridge-regression and elastic-net estimate for a linear regression problem where the response only depends on two groups of 6 predictors each out of a total of 50 predictors. The predictors in each group are highly correlated.

β_{lasso} *is a solution of the lasso, then*

$$\beta(\alpha)_i := \alpha\,\beta_{\text{lasso},i} + (1-\alpha)\;\beta_{\text{lasso},j}, \tag{57}$$

$$\beta(\alpha)_j := (1-\alpha)\;\beta_{\text{lasso},i} + \alpha\,\beta_{\text{lasso},j}, \tag{58}$$

$$\beta(\alpha)_k := \beta_{\text{lasso},k}, \quad k \notin \{i,j\}, \tag{59}$$

is also a solution for any $0 < \alpha < 1$.

The following lemma, which we have borrowed from [6], provides some intuition as to why strictly convex regularization functions such as ridge regression tend to weigh highly correlated predictors in a similar way. This does not contradict the previous result because the lasso is not strictly convex. The result is proved in Section A.8 of the appendix.

Lemma 3.3 (Identical predictors)**.** *Let us consider a regularized least squares problem of the form*

$$\text{minimize} \quad \frac{1}{2n}\left|\left|y - X\tilde{\beta}\right|\right|_2^2 + \lambda\,\mathcal{R}\left(\tilde{\beta}\right) \tag{60}$$

where $\mathcal{R}$ *is an arbitrary regularizer, which is strictly convex and invariant to the ordering of its argument. If two columns of the predictor matrix* X *are identical* $X_i = X_j$ *then the corresponding coefficients in the solution* $\beta_{\mathcal{R}}$ *of the optimization problem are also identical:* $\beta_{\mathcal{R},i} = \beta_{\mathcal{R},j}$.

For sparse regression problems where some predictors are highly correlated ridge regression weighs correlated predictors similarly, as opposed to the lasso, but does not yield a sparse model. The *elastic net* combines the lasso and ridge-regression cost functions introducing an extra regularization parameter α,

$$\text{minimize} \quad \left|\left|y - X\tilde{\beta}\right|\right|_2^2 + \lambda\left(\frac{1-\alpha}{2}\left|\left|\tilde{\beta}\right|\right|_2^2 + \frac{\alpha}{2}\left|\left|\tilde{\beta}\right|\right|_1\right). \tag{61}$$

For $\alpha = 0$ the elastic net is equivalent to ridge regression, whereas for $\alpha = 1$ it's equivalent to the lasso. For intermediate values of α the cost function yields sparse linear models where the coefficients corresponding to highly correlated predictors have similar amplitudes, as shown in Figure 8.

Figure 9 plots the training and test error achieved by least squares, ridge regression, the lasso and the elastic net on a dataset where the response only depends on two groups of highly correlated predictors. The total number of predictors in the dataset is $p = 50$ and the number of training examples is $n = 100$. Least squares overfits the data, yielding the best error for the training set. Ridge regression has a significantly lower test error, but does not achieve the performance of the lasso because it does not yield a sparse model as can be seen in Figure 8. The elastic net achieves the lowest test error.

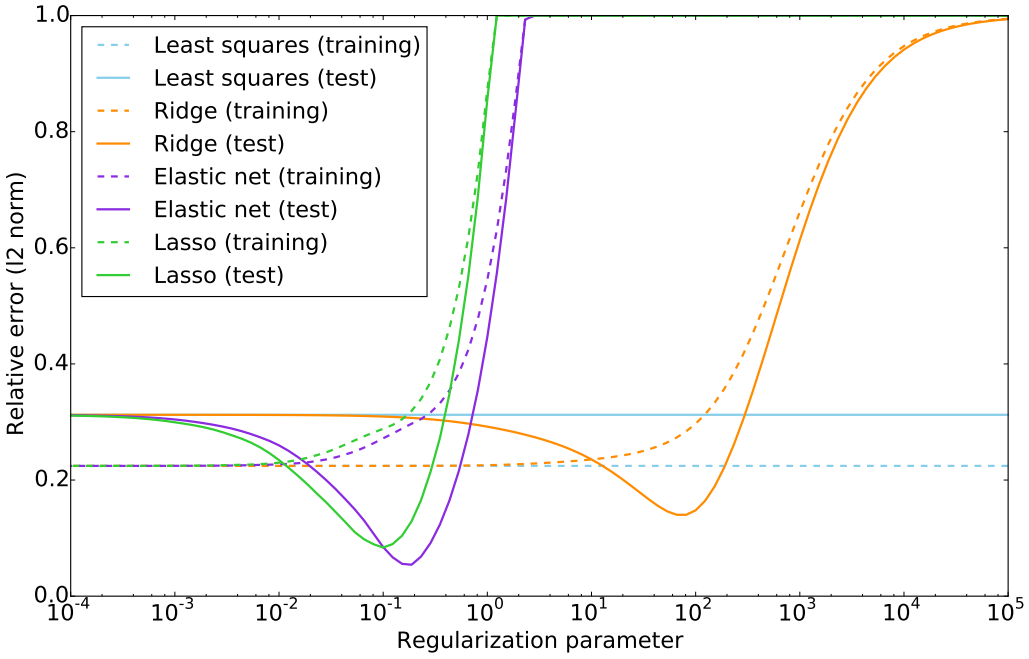


Figure 9: Training and test error achieved by least squares, ridge regression, the lasso and the elastic net on a dataset where the response only depends on two groups of highly correlated predictors. The coefficient paths are shown in Figure 8.

4 Group sparsity

4.1 Introduction

Group sparsity is a generalization of sparsity that allows us to design models that incorporate prior information about the data. If the entries of a vector are partitioned into several groups, then the vector is group sparse if only the entries corresponding to a small number of groups are nonzero, no matter how many entries are zero or nonzero *within the groups*. We have already exploited group sparsity in the context of denoising in Lecture Notes 4. There we used block thresholding to enforce the prior assumption that the STFT coefficients in a speech signal tend to have significant amplitude in contiguous areas. In this section, we will focus on the application of group-sparsity assumptions to regression models, where the groups are used to encode information about the structure of the predictors.

We consider a linear regression model where the predictors are partitioned into k groups $\mathcal{G}_1$, $\mathcal{G}_2$, $\ldots$, $\mathcal{G}_k$,

$$y \approx \beta_0 + X\beta = \beta_0 + \begin{bmatrix} X_{\mathcal{G}_1} & X_{\mathcal{G}_2} & \cdots & X_{\mathcal{G}_m} \end{bmatrix} \begin{bmatrix} \beta_{\mathcal{G}_1} \\ \beta_{\mathcal{G}_2} \\ \cdots \\ \beta_{\mathcal{G}_k} \end{bmatrix}. \tag{62}$$

A group-sparse regression model is a model in which only the predictors corresponding to a small number of groups are used to predict the response. For such models, the coefficient vector β has a group-sparse structure, since the entries corresponding to the rest of the groups are equal to zero. For example, if the predictors include the temperature, air pressure and other weather conditions at several locations, it might make sense to assume that the response will only depend on the predictors associated to a small number of locations. This implies that the β should be group sparse, where each group contains the predictors associated to a particular location.

4.2 Multi-task learning

Multi-task learning is a problem in machine learning and statistics which consists of learning models for several learning problems simultaneously, exploiting common structure. In the case of regression, an important example is when we want to estimate several responses Y_1, Y_2, $\ldots$, $Y_k \in \mathbb{R}^n$ that depend on the same predictors X_1, X_2, $\ldots$, $X_p \in \mathbb{R}^n$. To learn a linear

model for this problem we need to fit a matrix of coefficients $B \in \mathbb{R}^{p \times k}$,

$$Y = \begin{bmatrix} Y_1 & Y_2 & \dots & Y_k \end{bmatrix} \approx B_0 + XB \tag{63}$$

$$= B_0 + X \begin{bmatrix} B_1 & B_2 & \cdots & B_k \end{bmatrix}. \tag{64}$$

If we estimate B by solving a least-squares problem, then this is exactly equivalent to learning k linear regression separately, one for each response. However, a reasonable assumption in many cases is that the different responses depend on the *same predictors*, albeit with *with different coefficients*. This corresponds exactly to a group-sparse assumption on the coefficient matrix B: B should have a small number of nonzero rows.

4.3 Mixed ℓ_1/ℓ_2 norm

In order to promote group-sparse structure, a popular approach is to penalize the ℓ_1/ℓ_2 norm, which corresponds to the sum of the ℓ_2-norms of the entries in the different groups. The intuition is that minimizing the ℓ_1 norm induces sparsity, so minimizing the ℓ_1 norm of the ℓ_2-norms of the groups should induce sparsity at the group level.

Definition 4.1 (ℓ_1/ℓ_2 norm)**.** *The ℓ_1/ℓ_2 norm of a vector β with entries divided into k groups $\mathcal{G}_1$, $\mathcal{G}_2$, ..., $\mathcal{G}_k$ is defined as*

$$||\beta||_{1,2} := \sum_{i=1}^{k} ||\beta_{\mathcal{G}_i}||_2 \,. \tag{65}$$

In the case of multitask learning, where the groups correspond to the rows of a matrix, the ℓ_1/ℓ_2 norm of the matrix corresponds to the sum of the ℓ_2 norms of the rows,

$$||B||_{1,2} := \sum_{i=1}^{k} ||B_{:i}||_2 \,, \tag{66}$$

where $B_{:i}$ denotes the ith row of B.

Let us give a simple example that shows why penalizing the ℓ_1/ℓ_2 norm induces group sparsity. Let us define two groups $\mathcal{G}_1 := \{1, 2\}$ and $\mathcal{G}_1 := \{3\}$. The corresponding ℓ_1/ℓ_2 norm is given by

$$\left|\left|\begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix}\right|\right|_{1,2} = \sqrt{\beta_1^2 + \beta_2^2} + |\beta_3| \tag{67}$$

Our aim is to fit a regression model. Let us imagine that most of the response can be explained by the first predictor, so that

$$y \approx X \begin{bmatrix} \beta_1 \\ 0 \\ 0 \end{bmatrix}, \tag{68}$$

but the fit can be improved in two ways, by setting either β_2 or β_3 to a certain value α. The question is which of the two options is *cheaper* in terms of ℓ_1/ℓ_2 norm, since this is the option that will be chosen if we use the ℓ_1/ℓ_2 norm to regularize the fit. The answer is that modifying β_2 has much less impact on the ℓ_1/ℓ_2 norm

$$\left\| \begin{bmatrix} \beta_1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \alpha \\ 0 \end{bmatrix} \right\|_{1,2} = \sqrt{\beta_1^2 + \alpha^2}, \tag{69}$$

$$\left\| \begin{bmatrix} \beta_1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \alpha \end{bmatrix} \right\|_{1,2} = \sqrt{\beta_1^2 + \alpha^2 + 2\left|\beta_1\right|\alpha}, \tag{70}$$

especially if α is small or β_1 is large. The ℓ_1/ℓ_2 norm induces a group-sparse structure by making it less costly to include entries that belong to groups which already have nonzero entries, with respect to groups where all the entries are equal to zero.

The following lemma derives the subgradient of the ℓ_1/ℓ_2 norm. We defer the proof to Section A.9 in the appendix.

Lemma 4.2 (Subgradient of the ℓ_1/ℓ_2 norm)**.** *A vector $g \in \mathbb{R}^p$ is a subgradient of the ℓ_1/ℓ_2 norm at $\beta \in \mathbb{R}^p$ if and only if*

$$g_{\mathcal{G}_i} = \frac{\beta_{\mathcal{G}_i}}{||\beta_{\mathcal{G}_i}||_2} \quad \text{for } \beta_{\mathcal{G}_i} \neq 0, \tag{71}$$

$$||g_{\mathcal{G}_i}||_2 \leq 1 \quad \text{for } \beta_{\mathcal{G}_i} = 0. \tag{72}$$

4.4 Group and multitask lasso

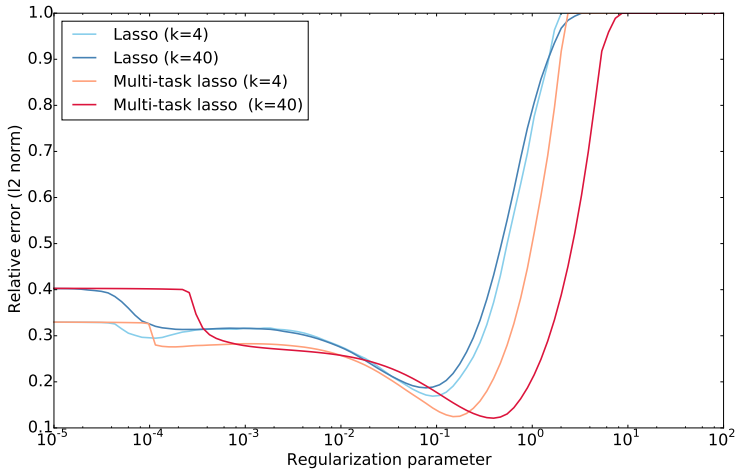
The group lasso [5] combines a least-squares term with an ℓ_1/ℓ_2-norm regularization term to fit a group-sparse linear regression model,

$$\text{minimize} \quad \left|\left|Y - \tilde{\beta}_0 - X\tilde{\beta}\right|\right|_{\text{F}}^2 + \lambda \left|\left|\tilde{\beta}\right|\right|_{1,2}, \tag{73}$$

where $\lambda > 0$ is a regularization parameter. Applying the exact same idea to the multi-task regression problem yields the multi-task lasso

$$\text{minimize} \quad \left|\left|Y - \tilde{B}_0 - X\tilde{B}\right|\right|_{\text{F}}^2 + \lambda \left|\left|\tilde{B}\right|\right|_{1,2}, \tag{74}$$

$s = 4$



$s = 30$

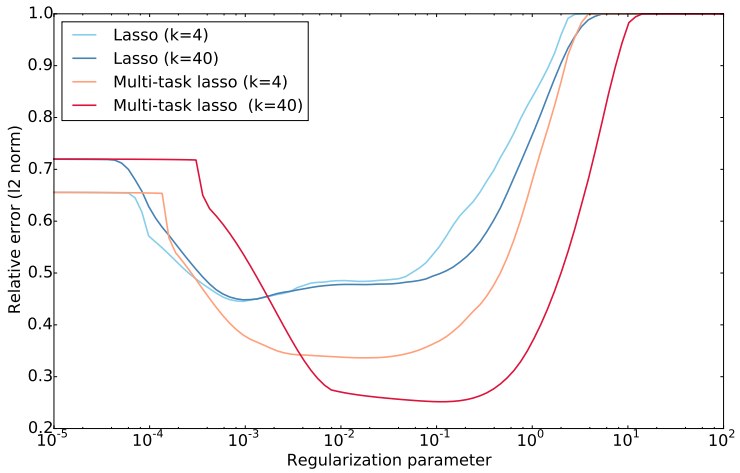


Figure 10: Errors achieved by the lasso and the multitask lasso on a multitask regression problem where the same s predictors (out of a total of $p = 100$ predictors) are used to produce the response in k sparse linear regression models. The training data is equal to $n = 50$.

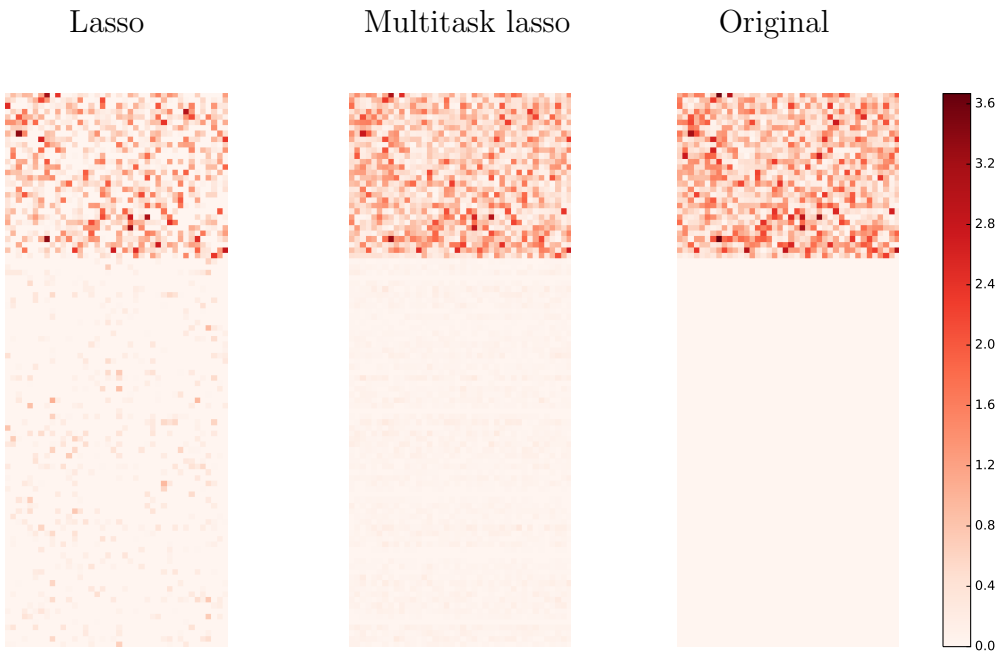


Figure 11: Coefficients for the lasso and the multitask lasso when $s = 30$ and $k = 40$ in the experiment described in Figure 10. The first 30 rows contain the relevant features.

where the Frobenius norm $||\cdot||_{\mathrm{F}}$ is equivalent to the ℓ_2 norm of the matrix once it is vectorized.

Figure 10 shows a comparison between the errors achieved by the lasso and the multitask lasso on a multitask regression problem where the same s predictors (out of a total of $p = 100$ predictors) are used to produce the response in k sparse linear regression models. The training data is equal to $n = 50$. The lasso fits each of the models separately, whereas the multitask lasso produces a joint fit. This allows to learn the model more effectively and achieve a lower error, particularly when the number of predictors is relatively large ($s = 30$). Figure 11 shows the actual coefficients fit by the lasso and the multitask lasso when $s = 30$ and $k = 40$. The multitask lasso is able to promote a group sparse structure which results in the correct identification of the relevant predictors. In contrast, the lasso fits sparse models that do not necessarily contain the same predictors, making it easier to include irrelevant predictors that seem relevant for a particular response because of the noise.

4.5 Proximal-gradient algorithm

In order to apply the group or the multitask lasso we need to solve a least-squares problem with an ℓ_1/ℓ_2-norm. In this section we adapt the proximal-gradient algorithm described in Lectures Notes 3 to this setting. The first step is to derive the proximal operator of this

norm.

Proposition 4.3 (Proximal operator of the ℓ_1/ℓ_2 norm)**.** *The solution to the optimization problem*

$$\text{minimize} \quad \frac{1}{2}\left|\left|\beta - \tilde{\beta}\right|\right|_2^2 + \alpha \left|\left|\tilde{\beta}\right|\right|_{1,2}, \tag{75}$$

where $\alpha > 0$, is obtained by applying a block soft-thresholding operator to β

$$\text{prox}_{\alpha ||\cdot||_{1,2}}(\beta) = \mathcal{BS}_\alpha(\beta), \tag{76}$$

where

$$\mathcal{BS}_\alpha(\beta)_{\mathcal{G}_i} := \begin{cases} \beta_{\mathcal{G}_i} - \alpha \frac{\beta_{\mathcal{G}_i}}{||\beta_{\mathcal{G}_i}||_2} & \text{if } ||\beta_{\mathcal{G}_i}||_2 \geq \alpha \\ 0 & \text{otherwise.} \end{cases} \tag{77}$$

The proof of this result is in Section A.10 of the appendix.

The proximal-gradient method alternates between block-thresholding and taking a gradient step to minimize the least-squares fit.

Algorithm 4.4 (Iterative Block-Thresholding Algorithm)**.** *We set the initial point $x^{(0)}$ to an arbitrary value in $\mathbb{R}^n$. Then we compute*

$$x^{(k+1)} = \mathcal{BS}_{\alpha_k \lambda}\left(x^{(k)} - \alpha_k A^T\left(Ax^{(k)} - y\right)\right), \tag{78}$$

until a convergence criterion is satisfied.

Convergence may be accelerated using the ideas discussed in Lecture Notes 3 to motivate the FISTA method.

References

The book by Hastie, Tibshirani and Wainwright [3] is a great reference on sparse regression. We also recommend [1].

[1] J. Friedman, T. Hastie, and R. Tibshirani. *The elements of statistical learning.*

[2] J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2010.

[3] T. Hastie, R. Tibshirani, and M. Wainwright. *Statistical learning with sparsity: the lasso and generalizations*. CRC Press, 2015.

[4] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[5] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.

[6] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.

A Proofs

A.1 Proof of Proposition 1.1

Let $X = U\Sigma V_T$ be the singular-value decomposition (SVD) of X. Under the conditions of the proposition, $\left(X^T X\right)^{-1} X^T y = V\Sigma U^T$. We begin by separating y into two components

$$y = UU^T y + \left(I - UU^T\right) y \tag{79}$$

where $UU^T y$ is the projection of y onto the column space of X. Note that $\left(I - UU^T\right) y$ is orthogonal to the column space of X and consequently to both $UU^T y$ and $X\tilde{\beta}$ for any $\tilde{\beta}$. By Pythagoras's Theorem

$$\left|\left|y - X\tilde{\beta}\right|\right|_2^2 = \left|\left|\left(I - UU^T\right) y\right|\right|_2^2 + \left|\left|UU^T y - X\tilde{\beta}\right|\right|_2^2. \tag{80}$$

The minimum value of this cost function that can be achieved by optimizing over $\tilde{beta}$ is $||y_{X^\perp}||_2^2$. This can be achieved by solving the system of equations

$$UU^T y = X\tilde{\beta} = U\Sigma V_T \tilde{\beta}. \tag{81}$$

Since $U^T U = I$ because $p \geq n$, multiplying both sides of the equality yields the equivalent system

$$U^T y = \Sigma V_T \tilde{\beta}. \tag{82}$$

Since X is full rank, Σ and V are square and invertible (and by definition of the SVD $V^{-1} = V^T$), so

$$\beta_{\text{ls}} = V\Sigma U^T y \tag{83}$$

is the unique solution to the system and consequently also of the least-squares problem.

A.2 Proof of Proposition 1.3

We model the noise as a random vector $\check{z}$ which has entries that are independent Gaussian random variables with mean zero and a certain variance σ. Note that $\check{z}$ is a random vector, whereas z is the *realization* of the random vector. Similarly, the data y that we observe is interpreted as a realization of a random vector $\check{y}$. The ith entry of

$$\check{y} = X\beta + \check{z} \tag{84}$$

is a Gaussian random variable with mean $(X\beta)_i$ and variance σ^2. The pdf of $\check{y}_i$ is consequently of the form

$$f_{\check{y}_i}(t) := \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\left(t - \left(X\tilde{\beta}\right)_i\right)^2}{2\sigma^2} \right). \tag{85}$$

By assumption, the entries of $\check{z}$ are independent, so the joint pdf of $\check{y}$ is equal to

$$f_{\check{y}}(\tilde{y}) := \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\left(\tilde{y}_i - \left(X\tilde{\beta}\right)_i\right)^2}{2\sigma^2} \right) \tag{86}$$

$$:= \frac{1}{\sqrt{(2\pi)^n}\sigma^n} \exp\left(-\frac{1}{2\sigma^2} \left|\left| \tilde{y} - X\tilde{\beta} \right|\right|_2^2 \right). \tag{87}$$

The likelihood is the probability density function of $\check{y}$ evaluated at the observed data y and interpreted as a function of $\tilde{\beta}$.

$$\mathcal{L}\left(\tilde{\beta}\right) = \frac{1}{\sqrt{(2\pi)^n}} \exp\left(-\frac{1}{2} \left|\left| y - X\tilde{\beta} \right|\right|_2^2 \right). \tag{88}$$

Since the function is nonnegative and the logarithm is a monotone function, we can take optimize over the logarithm of the likelihood to find the maximum-likelihood estimate. We conclude that it is given by the solution to the least-squares problem, since

$$\beta_{\mathrm{ML}} = \arg\max_{\tilde{\beta}} \mathcal{L}\left(\tilde{\beta}\right) \tag{89}$$

$$= \arg\max_{\tilde{\beta}} \log \mathcal{L}\left(\tilde{\beta}\right) \tag{90}$$

$$= \arg\min_{\tilde{\beta}} \left|\left| y - X\tilde{\beta} \right|\right|_2 \tag{91}$$

A.3 Proof of Lemma 1.4

Let 1 denote an n-dimensional vector of ones. The model with an intercept is equivalent to

$$y \approx \begin{bmatrix} X & 1 \end{bmatrix} \begin{bmatrix} \tilde{\beta} \\ \tilde{\beta}_0 \end{bmatrix}. \tag{92}$$

Applying Proposition 1.1 the least-squares fit is

$$\begin{aligned}
\begin{bmatrix} \beta \\ \beta_0 \end{bmatrix}_{\text{ls}} &= \left(\begin{bmatrix} X & 1 \end{bmatrix}^T \begin{bmatrix} X & 1 \end{bmatrix} \right)^{-1} \begin{bmatrix} X & 1 \end{bmatrix}^T y && (93) \\
&= \begin{bmatrix} X^T X & X^T 1 \\ 1^T X & n \end{bmatrix}^{-1} \begin{bmatrix} X^T y \\ 1^T y \end{bmatrix} && (94) \\
&= \begin{bmatrix} X^T X & 0 \\ 0 & n \end{bmatrix}^{-1} \begin{bmatrix} X^T y \\ 0 \end{bmatrix} && (95) \\
&= \begin{bmatrix} \left(X^T X\right)^{-1} & 0 \\ 0 & \frac{1}{n} \end{bmatrix} \begin{bmatrix} X^T y \\ 0 \end{bmatrix} && (96) \\
&= \begin{bmatrix} \left(X^T X\right)^{-1} X^T y \\ 0 \end{bmatrix}, && (97)
\end{aligned}$$

where have used the fact that $1^T y = 0$ and $X^T 1 = 0$ because the mean of y and of each of the columns of X is equal to zero.

A.4 Proof of Theorem 1.5

By Proposition 1.1

$$\begin{aligned}
||\beta - \beta_{\text{ls}}||_2^2 &= \left|\left|\beta - V\Sigma^{-1}U^T y\right|\right|_2^2 && (98) \\
&= \left|\left|\beta - V\Sigma^{-1}U^T \left(X\beta + z\right)\right|\right|_2^2 && (99) \\
&= \left|\left|V\Sigma^{-1}U^T z\right|\right|_2^2 && (100) \\
&= \left|\left|\Sigma^{-1}U^T z\right|\right|_2^2 && (101) \\
&= \sum_{j=1}^{p} \left(\frac{U_j^T z}{\sigma_j} \right)^2, && (102)
\end{aligned}$$

which implies

$$\frac{\left|\left|U^T z\right|\right|_2^2}{\sigma_{\max}^2} \leq ||\beta - \beta_{\text{ls}}||_2^2 \leq \frac{\left|\left|U^T z\right|\right|_2^2}{\sigma_{\min}^2}. \tag{103}$$

The distribution of the p-dimensional vector $U^T z$ is Gaussian with mean zero and covariance matrix

$$U^T \, \Sigma_z \, U = \sigma_z^2 I, \tag{104}$$

where $\Sigma_z = \sigma_z^2 I$ is the covariance of matrix of z. As a result, $\frac{1}{\sigma_z^2}\left|\left|U^T z\right|\right|_2^2$ is a chi-square random variable with p degrees of freedom. By Proposition A.2 in Lecture Notes 5 and the union bound

$$p\sigma_z^2 \, (1-\epsilon) \leq \left|\left|U^T z\right|\right|_2^2 \leq p\sigma_z^2 \, (1+\epsilon) \tag{105}$$

with probability at least $1 - \exp\left(-\frac{p\epsilon^2}{8}\right) - \exp\left(-\frac{p\epsilon^2}{2}\right) \geq 1 - 2\exp\left(-\frac{p\epsilon^2}{8}\right)$.

A.5 Proof of Theorem 2.3

We define the error

$$h := \beta - \beta_{\text{lasso}}. \tag{106}$$

The following lemma shows that the error satisfies the robust-sparsity condition in the definition of the restricted-eigenvalue property.

Lemma A.1. *In the setting of Theorem 2.3 $h := \beta - \beta_{\text{lasso}}$ satisfies*

$$||h_{T^c}||_1 \leq ||h_T||_1 \tag{107}$$

where T is the support of the nonzero entries of β.

Proof. Since β_{lasso} is feasible and β is supported on T

$$\begin{aligned}
\tau = ||\beta||_1 &\geq ||\beta_{\text{lasso}}||_1 && (108)\\
&= ||\beta + h||_1 && (109)\\
&= ||\beta + h_T||_1 + ||h_{T^c}||_1 && (110)\\
&\geq ||\beta||_1 - ||h_T||_1 + ||h_{T^c}||_1 . && (111)
\end{aligned}$$

□

This implies that by the restricted-eigenvalue property we have

$$||h||_2^2 \leq \frac{1}{\gamma n} ||X\ h||_2^2 . \tag{112}$$

The following lemma allows to bound the right-hand side.

Lemma A.2. *In the setting of Theorem 2.3*

$$||Xh||_2^2 \leq 2\, z^T X h. \tag{113}$$

Proof. Because β_{lasso} is the solution to the constrained optimization problem and β is also feasible we have

$$||y - X\beta_{\text{lasso}}||_2^2 \geq ||y - X\beta_{\text{lasso}}||_2^2 . \tag{114}$$

Substituting $y = X\beta + z$,

$$||z - Xh||_2^2 \geq ||z||_2^2 \tag{115}$$

which implies the result. □

Since $||h_{T^c}||_1 \leq ||h_T||_1$ and h_T only has s nonzero entries

$$||h||_1 \leq 2\sqrt{s}\, ||h||_2 \tag{116}$$

so by Lemma A.2, Hölder's inequality and (112)

$$||h||_2^2 \leq \frac{2\, z^T X h}{\gamma n} \tag{117}$$

$$\leq \frac{2\ \left|\left|X^T z\right|\right|_\infty ||h||_1}{\gamma n} \tag{118}$$

$$\leq \frac{4\sqrt{s}\, ||h||_2 \left|\left|X^T z\right|\right|_\infty}{\gamma n}. \tag{119}$$

The proof of the theorem is completed by the following lemma, that uses the assumption on the noise z to bound $\left|\left|X^T z\right|\right|_\infty$.

Lemma A.3. *In the setting of Theorem 2.3*

$$\mathrm{P}\left(\left|\left|X^T z\right|\right|_\infty > \sigma_z \sqrt{2\alpha\ n \log p}\right) \leq 2 \exp\left(-\left(\alpha - 1\right) \log p\right) \tag{120}$$

for any $\alpha > 2$.

Proof. $X_i^T z$ is Gaussian with variance $\sigma_z^2 \left|\left|X_i\right|\right|_2^2$, so for $t > 0$ by Lemma 3.5 in Lecture Notes 5

$$\mathrm{P}\left(\left|X_i^T z\right| > t\sigma_z \left|\left|X_i\right|\right|_2\right) \leq 2\exp\left(-\frac{t^2}{2}\right) \tag{121}$$

By the union bound,

$$\mathrm{P}\left(\left|\left|X^T z\right|\right|_\infty > t\sigma_z \max_i \left|\left|X_i\right|\right|_2\right) \leq 2\, p \exp\left(-\frac{t^2}{2}\right) \tag{122}$$

$$= 2\exp\left(-\frac{t^2}{2} + \log p\right) \tag{123}$$

Choosing $t = \sqrt{2\alpha \log p}$ for $\alpha > 2$ we obtain the desired result. □

A.6 Proof of Proposition 3.1

The ridge-regression cost function is equivalent to the least-squares cost function

$$\text{minimize} \quad \left|\left|\begin{bmatrix} y \\ 0 \end{bmatrix} - \begin{bmatrix} X \\ \lambda I \end{bmatrix} \tilde{\beta}\right|\right|_2^2. \tag{124}$$

By Proposition 1.1 the solution to this problem is

$$\beta_{\text{ridge}} := \left(\begin{bmatrix} X \\ \lambda I \end{bmatrix}^T \begin{bmatrix} X \\ \lambda I \end{bmatrix}\right)^{-1} \begin{bmatrix} X \\ \lambda I \end{bmatrix}^T \begin{bmatrix} y \\ 0 \end{bmatrix} \tag{125}$$

$$= \left(X^T X + \lambda^2 I\right)^{-1} X^T \left(X\beta + z\right) \tag{126}$$

$$= \left(V\Sigma^2 V^T + \lambda^2 V V^T\right)^{-1} \left(V\Sigma^2 V^T \beta + V\Sigma U^T z\right) \tag{127}$$

$$= V\left(\Sigma^2 + \lambda^2 I\right)^{-1} V^T \left(V\Sigma^2 V^T \beta + V\Sigma U^T z\right). \tag{128}$$

A.7 Proof of Lemma 3.2

Since $X_i = X_j$,

$$X\beta\left(\alpha\right) = \left(\alpha\, \beta_{\text{lasso},i} + (1-\alpha)\ \beta_{\text{lasso},j}\right) X_i + \left((1-\alpha)\ \beta_{\text{lasso},i} + \alpha\, \beta_{\text{lasso},j}\right) X_j + \sum_{k \notin \{i,j\}} \beta_{\text{lasso},k} X_k$$

$$= \beta_{\text{lasso},i} X_i + \beta_{\text{lasso},j} X_j + \sum_{k \notin \{i,j\}} \beta_{\text{lasso},k} X_k \tag{129}$$

$$= X\beta_{\text{lasso}}, \tag{130}$$

which implies $||y - X\beta(\alpha)||_2 = ||y - X\beta_{\text{lasso}}||_2$. Similarly,

$$||\beta(\alpha)||_1 = |\alpha\,\beta_{\text{lasso},i} + (1-\alpha)\;\beta_{\text{lasso},j}| + |(1-\alpha)\;\beta_{\text{lasso},i} + \alpha\,\beta_{\text{lasso},j}| + \sum_{k\notin\{i,j\}} |\beta_{\text{lasso},k}| \tag{131}$$

$$\leq \alpha\,|\beta_{\text{lasso},i}| + (1-\alpha)\,|\beta_{\text{lasso},j}| + (1-\alpha)\,|\beta_{\text{lasso},i}| + \alpha\,|\beta_{\text{lasso},j}| + \sum_{k\notin\{i,j\}} |\beta_{\text{lasso},k}|$$

$$= ||\beta_{\text{lasso}}||_1\,. \tag{132}$$

This implies that $\beta(\alpha)$ must also be a solution.

A.8 Proof of Lemma 3.3

Consider

$$\beta(\alpha)_i := \alpha\,\beta_{\mathcal{R},i} + (1-\alpha)\;\beta_{\mathcal{R},j}, \tag{133}$$

$$\beta(\alpha)_j := (1-\alpha)\;\beta_{\mathcal{R},i} + \alpha\,\beta_{\mathcal{R},j}, \tag{134}$$

$$\beta(\alpha)_k := \beta_{\mathcal{R},k}, \quad k \notin \{i,j\}\,. \tag{135}$$

for $0 < \alpha < 1$. By the same argument in (130) $||y - X\beta(\alpha)||_2 = ||y - X\beta_{\mathcal{R}}||_2$. We define

$$\beta'_{\mathcal{R},i} := \beta_{\mathcal{R},j}, \tag{136}$$

$$\beta'_{\mathcal{R},j} := \beta_{\mathcal{R},i}, \tag{137}$$

$$\beta'_{\mathcal{R},k} := \beta_{\mathcal{R},k}, \quad k \notin \{i,j\}\,. \tag{138}$$

Note that because $\mathcal{R}$ is invariant to the ordering of its argument $\mathcal{R}(\beta_{\mathcal{R}}) = \mathcal{R}(\beta'_{\mathcal{R}})$. Since $\beta(\alpha) = \alpha\beta_{\mathcal{R}} + (1-\alpha)\,\beta'_{\mathcal{R}}$, by strict convexity of $\mathcal{R}$

$$\mathcal{R}(\beta(\alpha)) < \alpha\;\mathcal{R}(\beta_{\mathcal{R}}) + (1-\alpha)\,\mathcal{R}(\beta'_{\mathcal{R}}) \tag{139}$$

$$= \mathcal{R}(\beta_{\mathcal{R}}) \tag{140}$$

if $\beta_{\mathcal{R},i} \neq \beta_{\mathcal{R},j}$. Since this would mean that $\beta_{\mathcal{R}}$ is not a solution to the regularization problem, this implies that $\beta_{\mathcal{R},i} = \beta_{\mathcal{R},j}$.

A.9 Proof of Lemma 4.2

We have that

$$||\beta + h||_{1,2} \geq ||\beta||_{1,2} + g^T h \tag{141}$$

for all possible $h \in \mathbb{R}^p$ if and only if

$$||\beta_{\mathcal{G}_i} + h_{\mathcal{G}_i}||_2 \geq ||\beta_{\mathcal{G}_i}||_2 + g_{\mathcal{G}_i}^T h_{\mathcal{G}_i} \tag{142}$$

for all possible $h_{\mathcal{G}_i} \in \mathbb{R}^{|\mathcal{G}_i|}$, for $1 \leq i \leq k$.

If $\beta_{\mathcal{G}_i} \neq 0$ the only vector $g_{\mathcal{G}_i}$ that satisfies (142) is the gradient of the ℓ_2 norm at $\beta_{\mathcal{G}_i}$

$$\nabla ||\cdot||_2 (\beta_{\mathcal{G}_i}) = \frac{\beta_{\mathcal{G}_i}}{||\beta_{\mathcal{G}_i}||_2}. \tag{143}$$

The fact that the gradient is of this form follows from the chain rule.

If $\beta_{\mathcal{G}_i} = 0$ any vector $g_{\mathcal{G}_i}$ with ℓ_2 norm bounded by one satisfies (142) by the Cauchy-Schwarz inequality.

A.10 Proof of Lemma 4.3

We can separate the minimization problem into the different groups

$$\min_{\tilde{\beta}} \frac{1}{2} \left|\left|\beta - \tilde{\beta}\right|\right|_2^2 + \alpha \left|\left|\tilde{\beta}\right|\right|_{1,2} = \sum_{i=1}^{k} \min_{\tilde{\beta}_{\mathcal{G}_i}} \frac{1}{2} \left|\left|\beta_{\mathcal{G}_i} - \tilde{\beta}_{\mathcal{G}_i}\right|\right|_2^2 + \alpha \left|\left|\tilde{\beta}_{\mathcal{G}_i}\right|\right|_2. \tag{144}$$

We can therefore minimize the different terms of the sum separately. Each term

$$\frac{1}{2} \left|\left|\beta_{\mathcal{G}_i} - \tilde{\beta}_{\mathcal{G}_i}\right|\right|_2^2 + \alpha \left|\left|\tilde{\beta}_{\mathcal{G}_i}\right|\right|_2 \tag{145}$$

is convex and has subgradients of the form

$$g\left(\tilde{\beta}_{\mathcal{G}_i}\right) := \tilde{\beta}_{\mathcal{G}_i} - \beta_{\mathcal{G}_i} + \alpha\, q\left(\tilde{\beta}_{\mathcal{G}_i}\right), \tag{146}$$

$$q\left(\tilde{\beta}_{\mathcal{G}_i}\right) := \begin{cases} \frac{\tilde{\beta}_{\mathcal{G}_i}}{||\tilde{\beta}_{\mathcal{G}_i}||_2} & \text{if } \tilde{\beta}_{\mathcal{G}_i} \neq 0, \\ 0 \quad \text{if } \tilde{\beta}_{\mathcal{G}_i} \neq 0. \end{cases} \tag{147}$$

This follows from Lemma 4.2 and the fact that the sum of subgradients of several functions is a subgradient of their sum.

Any minimizer $\hat{\beta}_{\mathcal{G}_i}$ of (145) must satisfy

$$g\left(\hat{\beta}_{\mathcal{G}_i}\right) = 0. \tag{148}$$

This implies that if $\hat{\beta}_{\mathcal{G}_i} \neq 0$ then

$$\beta_{\mathcal{G}_i} = \hat{\beta}_{\mathcal{G}_i} + \frac{\alpha\, \hat{\beta}_{\mathcal{G}_i}}{\left|\left|\hat{\beta}_{\mathcal{G}_i}\right|\right|_2} \tag{149}$$

$$= \left(\left|\left|\hat{\beta}_{\mathcal{G}_i}\right|\right|_2 + \alpha\right) \frac{\hat{\beta}_{\mathcal{G}_i}}{\left|\left|\hat{\beta}_{\mathcal{G}_i}\right|\right|_2}. \tag{150}$$

As a result, $\beta_{\mathcal{G}_i}$ and $\hat{\beta}_{\mathcal{G}_i}$ are collinear and

$$\left|\left|\hat{\beta}_{\mathcal{G}_i}\right|\right|_2 = ||\beta_{\mathcal{G}_i}||_2 - \alpha, \tag{151}$$

which can only hold if $||\beta_{\mathcal{G}_i}||_2 \geq \alpha$. In that case,

$$\hat{\beta}_{\mathcal{G}_i} = \beta_{\mathcal{G}_i} - \frac{\alpha\, \hat{\beta}_{\mathcal{G}_i}}{\left|\left|\hat{\beta}_{\mathcal{G}_i}\right|\right|_2} \tag{152}$$

$$= \beta_{\mathcal{G}_i} - \frac{\alpha\, \beta_{\mathcal{G}_i}}{||\beta_{\mathcal{G}_i}||_2}. \tag{153}$$

This establishes that as long as $||\beta_{\mathcal{G}_i}||_2 \geq \alpha$ (153) is a solution to the proximal problem.

If $\hat{\beta}_{\mathcal{G}_i} = 0$ then by (148)

$$\alpha \geq \left|\left|\hat{\beta}_{\mathcal{G}_i} - \beta_{\mathcal{G}_i}\right|\right|_2 \tag{154}$$

$$= ||\beta_{\mathcal{G}_i}||_2 \,. \tag{155}$$

This establishes that as long as $||\beta_{\mathcal{G}_i}||_2 \leq \alpha$ $\hat{\beta}_{\mathcal{G}_i} = 0$ is a solution to the proximal problem.