Lecture Notes 6: Linear Models

1 Linear regression

1.1 The regression problem

In statistics, regression is the problem of characterizing the relation between a quantity of interest y, called the *response* or the *dependent variable*, and several observed variables x_1, x_2, …, x_p, known as *covariates*, *features* or *independent variables*. For example, the response could be the price of a house and the covariates could correspond to the extension, the number of rooms, the year it was built, etc. A regression model would describe how house prices are affected by all of these factors.

More formally, the main assumption in regression models is that the predictor is generated according to a function h applied to the features and then perturbed by some unknown noise z, which is often modeled as additive,

$$y = h\left(\vec{x}\right) + z. \tag{1}$$

The aim is to learn h from n examples of responses and their corresponding features

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

If the regression function h in a model of the form (1) is linear, then the response is modeled as a linear combination of the predictors:

$$y^{(i)} = \left\langle \vec{x}^{\,(i)}, \vec{\beta}^{*} \right\rangle + z^{(i)}, \quad 1 \leq i \leq n, \tag{3}$$

where $z^{(i)}$ is an entry of the unknown noise vector. The function is parametrized by a vector of coefficients $\vec{\beta}^{*} \in \mathbb{R}^p$. All we need to fit the linear model to the data is to estimate these coefficients.

Expressing the linear system (3) in matrix form, we have

$$\begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \cdots \\ y^{(n)} \end{bmatrix} = \begin{bmatrix} \vec{x}^{\,(1)}[1] & \vec{x}^{\,(1)}[2] & \cdots & \vec{x}^{\,(1)}[p] \\ \vec{x}^{\,(2)}[1] & \vec{x}^{\,(2)}[2] & \cdots & \vec{x}^{\,(2)}[p] \\ \cdots & \cdots & \cdots & \cdots \\ \vec{x}^{\,(n)}[1] & \vec{x}^{\,(n)}[2] & \cdots & \vec{x}^{\,(n)}[p] \end{bmatrix} \begin{bmatrix} \vec{\beta}^{*}[1] \\ \vec{\beta}^{*}[2] \\ \cdots \\ \vec{\beta}^{*}[p] \end{bmatrix} + \begin{bmatrix} z^{(1)} \\ z^{(2)} \\ \cdots \\ z^{(n)} \end{bmatrix}. \tag{4}$$

This yields a more succinct representation of the linear-regression model:

$$\vec{y} = X\vec{\beta}^{*} + \vec{z}, \tag{5}$$

where X is a $n \times p$ matrix containing the features, $\vec{y} \in \mathbb{R}^n$ contains the response and $\vec{z} \in \mathbb{R}^n$ represents the noise.

For simplicity we mostly discuss the linear model (3), but in practice we usually fit an *affine* model that includes a constant term β_0,

$$y^{(i)} = \beta_0 + \left\langle \vec{x}^{\,(i)}, \vec{\beta}^{*} \right\rangle + z^{(i)}, \quad 1 \leq i \leq n. \tag{6}$$

This term is called an *intercept*, because if there is no noise $y^{(i)}$ is equal to β_0 when the features are all equal to zero. For a least-squares fit (see Section 2 below), β_0 can be shown to equal zero as long as the response $\vec{y}$ and the features $\vec{x}_1, \ldots, \vec{x}_p$ are all centered. This is established rigorously in Lemma 2.2. In addition to centering, it is common to normalize the response and the features before fitting a regression model, in order to ensure that all the variables have the same order of magnitude and the model is invariant to changes in units.

Example 1.1 (Linear model for GDP)**.** We consider the problem of building a linear model to predict the gross domestic product (GDP) of a state in the US from its population and unemployment rate. We have available the following data:

$$\begin{array}{l|ccc}
 & \text{GDP} & \text{Population} & \text{Unemployment} \\
 & \text{(USD millions)} & & \text{rate (\%)} \\
\text{North Dakota} & 52\,089 & 757\,952 & 2.4 \\
\text{Alabama} & 204\,861 & 4\,863\,300 & 3.8 \\
\text{Mississippi} & 107\,680 & 2\,988\,726 & 5.2 \\
\text{Arkansas} & 120\,689 & 2\,988\,248 & 3.5 \\
\text{Kansas} & 153\,258 & 2\,907\,289 & 3.8 \\
\text{Georgia} & 525\,360 & 10\,310\,371 & 4.5 \\
\text{Iowa} & 178\,766 & 3\,134\,693 & 3.2 \\
\text{West Virginia} & 73\,374 & 1\,831\,102 & 5.1 \\
\text{Kentucky} & 197\,043 & 4\,436\,974 & 5.2 \\
\text{Tennessee} & ??? & 6\,651\,194 & 3.0
\end{array}$$

In this example, the GDP is the response, whereas the population and the unemployment rate are the features. Our goal is to fit a linear model to the data so that we can predict the GDP of Tennessee, using a linear model. We begin by centering and normalizing the data. The averages of the response and of the features are

$$\operatorname{av}(\vec{y}) = 179\,236, \qquad \operatorname{av}(X) = \begin{bmatrix} 3\,802\,073 & 4.1 \end{bmatrix}. \tag{7}$$

The empirical standard deviations are

$$\operatorname{std}(\vec{y}) = 396\,701, \qquad \operatorname{std}(X) = \begin{bmatrix} 7\,720\,656 & 2.80 \end{bmatrix}. \tag{8}$$

We subtract the average and divide by the standard deviations so that both the response and the features are centered and on the same scale,

$$\vec{y} = \begin{bmatrix} -0.321 \\ 0.065 \\ -0.180 \\ -0.148 \\ -0.065 \\ 0.872 \\ -0.001 \\ -0.267 \\ 0.045 \end{bmatrix}, \qquad X = \begin{bmatrix} -0.394 & -0.600 \\ 0.137 & -0.099 \\ -0.105 & 0.401 \\ -0.105 & -0.207 \\ -0.116 & -0.099 \\ 0.843 & 0.151 \\ -0.086 & -0.314 \\ -0.255 & 0.366 \\ 0.082 & 0.401 \end{bmatrix}. \tag{9}$$

To obtain the estimate for the GDP of Tennessee we fit the model

$$\vec{y} \approx X\vec{\beta}, \tag{10}$$

rescale according to the standard deviations (8) and recenter using the averages (7). The final estimate is

$$\vec{y}^{\,\text{Ten}} = \text{av}\left(\vec{y}\right) + \text{std}\left(\vec{y}\right) \left\langle \vec{x}_{\text{norm}}^{\,\text{Ten}}, \vec{\beta} \right\rangle \tag{11}$$

where $\vec{x}_{\text{norm}}^{\,\text{Ten}}$ is centered using $\text{av}\left(X\right)$ and normalized using $\text{std}\left(X\right)$. △

1.2 Overfitting

Imagine that a friend tells you:

I found a cool way to predict the daily 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. The problem is that in this example the number of data points is roughly the same as the number of parameters. If $n \leq p$ we can find a $\vec{\beta}$ such that $\vec{y} = X\vec{\beta}$ exactly, even if $\vec{y}$ and X have nothing to do with each other! This is called *overfitting*: the model is too flexible given the available data. Recall from linear algebra that for a matrix $A \in \mathbb{R}^{n\times p}$ that is full rank, the linear system of equations

$$A\vec{b} = \vec{c} \tag{12}$$

is (1) underdetermined if $n < p$, meaning that it has infinite solutions, (2) determined if $n = p$, meaning that there is a unique solution, and (3) overdetermined if $n > p$. Fitting a linear model without any additional assumptions only makes sense in the overdetermined regime. In that case, an exact solution exists if $\vec{b} \in \text{col}\left(A\right)$, which is never the case in practice due to the presence of noise. However, if we manage to find a vector $\vec{b}$ such that $A\vec{b}$ is a good approximation to $\vec{c}$ when $n > p$ then this is an indication that the linear model is capturing some underlying structure in the problem. We make this statement more precise in Section 2.4

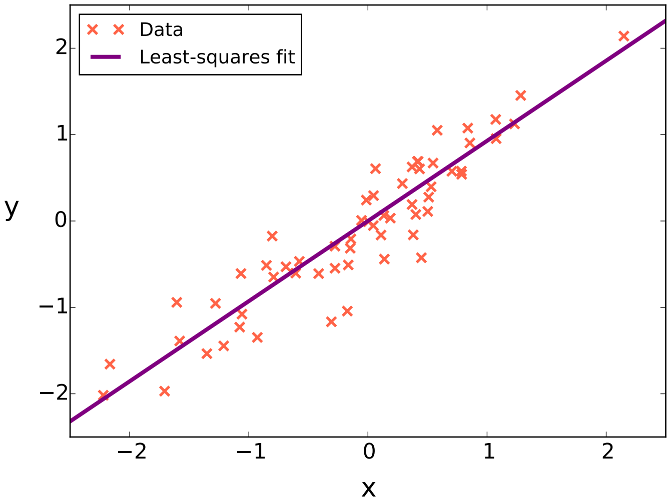


Figure 1: Linear model learned via least-squares fitting for a simple example where there is just one feature ($p = 1$) and 40 examples ($n = 40$).

2 Least-squares estimation

2.1 Minimizing the ℓ_2-norm approximation error

To calibrate the linear regression model $\vec{y} \approx X\vec{\beta}$ it is necessary to choose a metric to evaluate the fit achieved by the model. By far, the most popular metric is the sum of the squares of the fitting error,

$$\sum_{i=1}^{n} \left(y^{(i)} - \left\langle \vec{x}^{\,(i)}, \vec{\beta} \right\rangle \right)^2 = \left|\left| \vec{y} - X\vec{\beta} \right|\right|_2^2 . \tag{13}$$

The least-squares estimate $\vec{\beta}_{\text{LS}}$ is the vector of coefficients that minimizes this cost function,

$$\vec{\beta}_{\text{LS}} := \arg\min_{\vec{\beta}} \quad \left|\left| \vec{y} - X\vec{\beta} \right|\right|_2 . \tag{14}$$

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, it has intuitive geometric and probabilistic interpretations. Figure 1 shows the linear model learned using least squares in a simple example where there is just one feature ($p = 1$) and 40 examples ($n = 40$).

Theorem 2.1. *If X is full rank and $n \geq p$, for any $\vec{y} \in \mathbb{R}^n$ we have*

$$\begin{aligned} \vec{\beta}_{\text{LS}} &:= \arg\min_{\vec{\beta}} \quad \left|\left| \vec{y} - X\vec{\beta} \right|\right|_2 && (15) \\ &= VS^{-1}U^T\vec{y} && (16) \\ &= \left(X^TX\right)^{-1} X^T\vec{y}, && (17) \end{aligned}$$

where USV^T is the SVD of X.

Proof. We consider the decomposition of $\vec{y}$ into its orthogonal projection $UU^T\vec{y}$ onto the column space of X $\operatorname{col}(X)$ and its projection $\left(I - UU^T\right)\vec{y}$ onto the orthogonal complement of $\operatorname{col}(X)$. $X\vec{\beta}$ belongs to $\operatorname{col}(X)$ for any β and is consequently orthogonal to $\left(I - UU^T\right)\vec{y}$ (as is $UU^T\vec{y}$), so that

$$\arg\min_{\vec{\beta}} \left|\left|\vec{y} - X\vec{\beta}\right|\right|_2^2 = \arg\min_{\vec{\beta}} \left|\left|\left(I - UU^T\right)\vec{y}\right|\right|_2^2 + \left|\left|UU^T\vec{y} - X\vec{\beta}\right|\right|_2^2 \tag{18}$$

$$= \arg\min_{\vec{\beta}} \left|\left|UU^T\vec{y} - X\vec{\beta}\right|\right|_2^2 \tag{19}$$

$$= \arg\min_{\vec{\beta}} \left|\left|UU^T\vec{y} - USV^T\vec{\beta}\right|\right|_2^2. \tag{20}$$

Since U has orthonormal columns, for any vector $\vec{v} \in \mathbb{R}^p$ $||U\vec{v}||_2 = ||\vec{v}||_2$, which implies

$$\arg\min_{\vec{\beta}} \left|\left|\vec{y} - X\vec{\beta}\right|\right|_2^2 = \arg\min_{\vec{\beta}} \left|\left|U^T\vec{y} - SV^T\vec{\beta}\right|\right|_2^2 \tag{21}$$

If X is full rank and $n \geq p$, then SV^T is square and full rank. It therefore has a unique inverse, which is equal to VS^{-1}. As a result $VS^{-1}U^T\vec{y} = \left(X^TX\right)^{-1}X^T\vec{y}$ is the unique solution to the optimization problem (it is the only vector that yields a value of zero for the cost function). □

The following lemma shows that centering the data before computing the least-squares fit is exactly equivalent to fitting an affine model with the same cost function.

Lemma 2.2 (Proof in Section 5.1)**.** *For any matrix $X \in \mathbb{R}^{n\times m}$ and any vector $\vec{y}$, let*

$$\left\{\beta_{\text{LS},0}, \vec{\beta}_{\text{LS}}\right\} := \arg\min_{\beta_0,\vec{\beta}} \left|\left|\vec{y} - X\vec{\beta} - \beta_0\vec{1}\right|\right|_2^2 \tag{22}$$

be the coefficients corresponding to an affine fit, where $\vec{1}$ is a vector containing n ones, and let

$$\vec{\beta}_{\text{LS}}^{\text{cent}} := \arg\min_{\vec{\beta}} \left|\left|\vec{y}^{\text{cent}} - X^{\text{cent}}\vec{\beta}\right|\right|_2^2 \tag{23}$$

be the coefficients of a linear fit after centering both X and $\vec{y}$ using their respective averages (in the case of X, the column-wise average). Then,

$$X\vec{\beta}_{\text{LS}} + \beta_{\text{LS},0} = X^{\text{cent}}\vec{\beta}_{\text{LS}}^{\text{cent}} + \operatorname{av}(y). \tag{24}$$

Example 2.3 (Linear model for GDP (continued))**.** The least-squares estimate for the regression coefficients in the linear GDP model is equal to

$$\vec{\beta}_{\text{LS}} = \begin{bmatrix} 1.019 \\ -0.111 \end{bmatrix}. \tag{25}$$

The GDP seems to be proportional to the population and inversely proportional to the unemployment rate. We now compare the fit provided by the linear model to the original data, as well as its prediction of the GDP of Tennessee:

$$\begin{array}{lcc} & \text{GDP} & \text{Estimate} \\ \text{North Dakota} & 52\,089 & 46\,241 \\ \text{Alabama} & 204\,861 & 239\,165 \\ \text{Mississippi} & 107\,680 & 119\,005 \\ \text{Arkansas} & 120\,689 & 145\,712 \\ \text{Kansas} & 153\,258 & 136\,756 \\ \text{Georgia} & 525\,360 & 513\,343 \\ \text{Iowa} & 178\,766 & 158\,097 \\ \text{West Virginia} & 73\,374 & 59\,969 \\ \text{Kentucky} & 197\,043 & 194\,829 \\ \text{Tennessee} & 328\,770 & 345\,352 \end{array}$$

$\triangle$

Example 2.4 (Global warming)**.** In this example 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

$$y \approx \beta_0 + \beta_1 \cos\left(\frac{2\pi t}{12}\right) + \beta_2 \sin\left(\frac{2\pi t}{12}\right) + \beta_3\, t \tag{26}$$

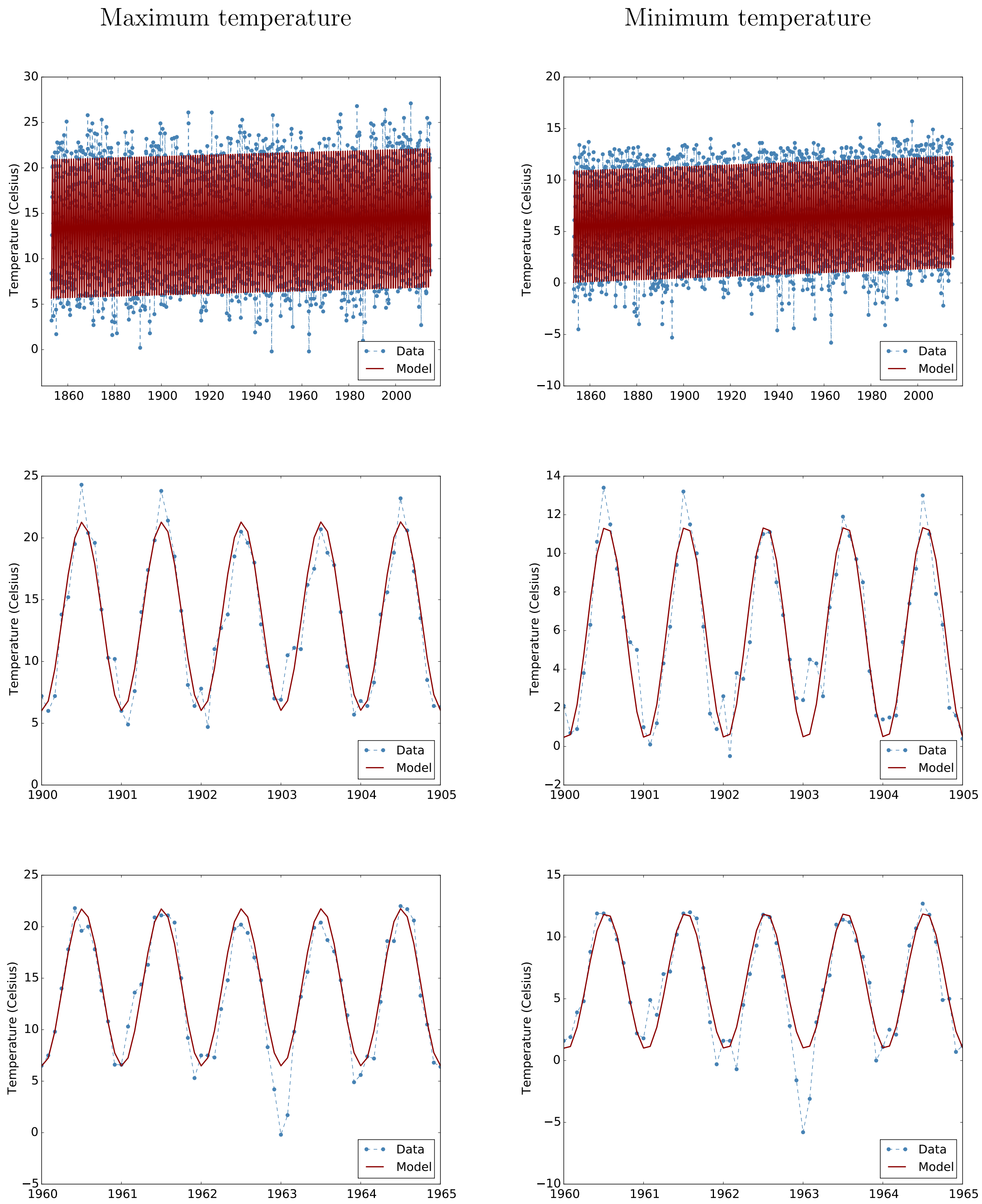
where t denotes the time in months. The corresponding matrix of predictors is

$$X := \begin{bmatrix} 1 & \cos\left(\frac{2\pi t_1}{12}\right) & \sin\left(\frac{2\pi t_1}{12}\right) & t_1 \\ 1 & \cos\left(\frac{2\pi t_2}{12}\right) & \sin\left(\frac{2\pi t_2}{12}\right) & t_2 \\ \cdots & \cdots & \cdots & \cdots \\ 1 & \cos\left(\frac{2\pi t_n}{12}\right) & \sin\left(\frac{2\pi t_n}{12}\right) & t_n \end{bmatrix}. \tag{27}$$

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 using least squares are shown in Figures 2 and 3. 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). $\triangle$

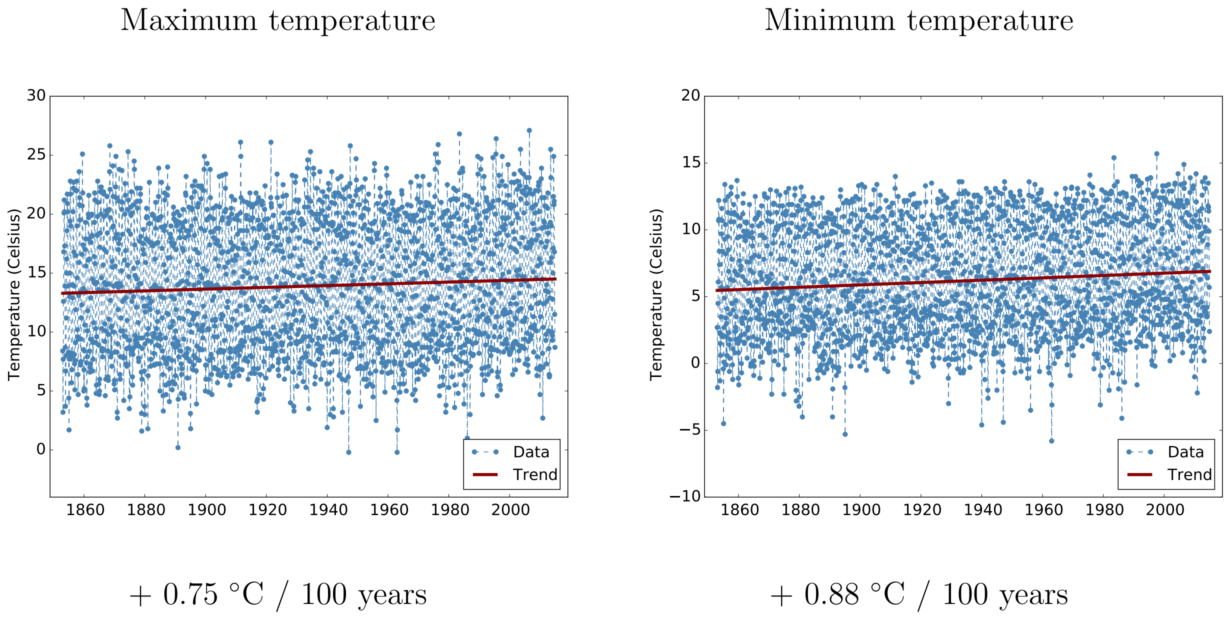
[1] The data are available at http://www.metoffice.gov.uk/pub/data/weather/uk/climate/stationdata/oxforddata.txt.



Maximum temperature

Minimum temperature

Figure 2: Temperature data together with the linear model described by (26) for both maximum and minimum temperatures.


Maximum temperature

Minimum temperature

+ 0.75 °C / 100 years

+ 0.88 °C / 100 years

Figure 3: Temperature trend obtained by fitting the model described by (26) for both maximum and minimum temperatures.

2.2 Geometric interpretation of least-squares regression

The following corollary of Theorem 2.1 provides an intuitive geometric interpretation of the linear approximation obtained from a least-squares fit. The least-squares fit yields the vector $X\vec{\beta}$ in the column space $\operatorname{col}(X)$ of the features that is closest to $\vec{y}$ in ℓ_2 norm. $X\vec{\beta}_{\text{LS}}$ is therefore the orthogonal projection of $\vec{y}$ onto $\operatorname{col}(X)$, as depicted in Figure 4.

Corollary 2.5. *The least-squares approximation of $\vec{y}$ obtained by solving problem* (14)

$$\vec{y}_{\text{LS}} = X\vec{\beta}_{\text{LS}} \tag{28}$$

is equal to the orthogonal projection of $\vec{y}$ onto the column space of X.

Proof.

$$\begin{aligned} X\vec{\beta}_{\text{LS}} &= USV^TVS^{-1}U^T\vec{y} && (29) \\ &= UU^T\vec{y} && (30) \end{aligned}$$

□

Example 2.6 (Denoising of face images)**.** In Example 7.4 of Lecture Notes 1, we denoised a noisy image by projecting it onto the span of a set of clean images. This is equivalent to solving a least-squares linear-regression problem in which the response is the noisy images and the columns of the matrix of features correspond to the clean faces. The regression coefficients are used to combine the different clean faces linearly to produce the estimate. △

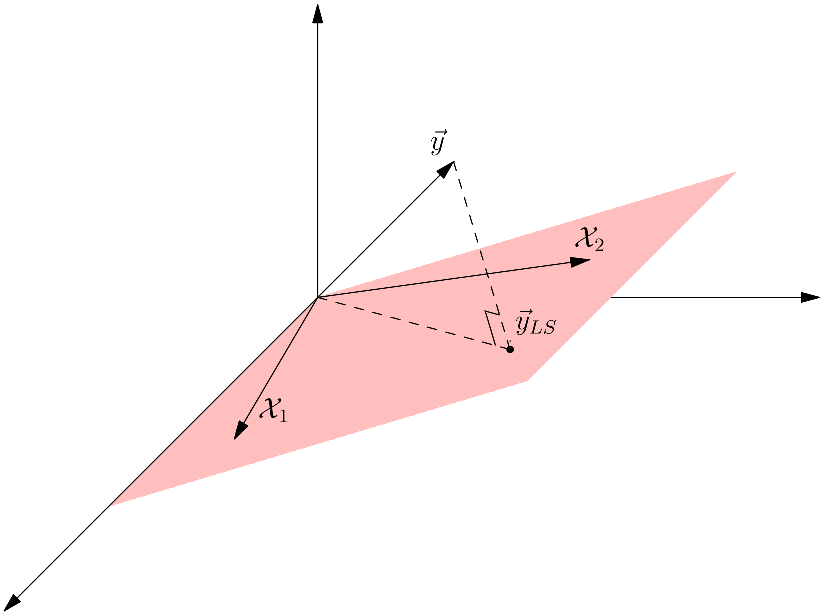


Figure 4: Illustration of Corollary 2.5. The least-squares solution is the orthogonal projection of the data onto the subspace spanned by the columns of X, denoted by X_1 and X_2.

2.3 Probabilistic interpretation of least-squares regression

In this section we derive the least-squares regression estimate as a maximum-likelihood (ML) estimator. ML estimation is a popular method for learning parametric models. In parametric estimation we assume that the data are sampled from a known distribution that depends on some unknown parameters, which we aim to estimate. The *likelihood* function is the joint pmf or pdf of the data, interpreted as a function of the unknown parameters.

Definition 2.7 (Likelihood function)**.** *Given a realization $\vec{y} \in \mathbb{R}^n$ of random vector $\vec{\mathbf{y}}$ with joint pdf $f_{\vec{\beta}}$ parameterized by a vector of parameters $\vec{\beta} \in \mathbb{R}^m$, the likelihood function is*

$$\mathcal{L}_{\vec{y}}\left(\vec{\beta}\right) := f_{\vec{\beta}}\left(\vec{y}\right). \tag{31}$$

The log-likelihood function *is equal to the logarithm of the likelihood function* $\log \mathcal{L}_{\vec{y}}\left(\vec{\beta}\right)$.

The likelihood function represents the probability density of the parametric distribution at the observed data, i.e. it quantifies how *likely* the data are according to the model. Therefore, higher likelihood values indicate that the model is better adapted to the samples. The maximum-likelihood (ML) estimator is a very popular parameter estimator based on maximizing the likelihood (or equivalently the log-likelihood).

Definition 2.8 (Maximum-likelihood estimator)**.** *The maximum likelihood (ML) estimator of the* *vector of parameters $\vec{\beta} \in \mathbb{R}^m$ is*

$$\vec{\beta}_{\text{ML}}\left(\vec{y}\right) := \arg\max_{\vec{\beta}} \mathcal{L}_{\vec{y}}\left(\vec{\beta}\right) \tag{32}$$

$$= \arg\max_{\vec{\beta}} \log \mathcal{L}_{\vec{y}}\left(\vec{\beta}\right). \tag{33}$$

The maximum of the likelihood function and that of the log-likelihood function are at the same location because the logarithm is a monotone function.

The following lemma shows that the least-squares estimate can be interpreted as an ML estimator.

Lemma 2.9. *Let $\vec{y} \in \mathbb{R}^n$ be a realization of a random vector*

$$\vec{\mathbf{y}} := X\vec{\beta} + \vec{\mathbf{z}}, \tag{34}$$

where $\vec{\mathbf{z}}$ is iid Gaussian with mean zero and variance σ^2. If $X \in \mathbb{R}^{n\times m}$ is known, then the ML estimate of $\vec{\beta}$ is equal to the least-squares estimate

$$\vec{\beta}_{\text{ML}} = \arg\min_{\vec{\beta}} \left|\left|\vec{y} - X\vec{\beta}\right|\right|_2^2. \tag{35}$$

Proof. For a fixed $\vec{\beta}$, the joint pdf of $\vec{\mathbf{y}}$ is equal to

$$f_{\vec{\beta}}\left(\vec{y}\right) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}\left(\vec{y}[i] - \left(X\vec{\beta}\right)[i]\right)^2\right) \tag{36}$$

$$= \frac{1}{\sqrt{(2\pi)^n}\sigma^n} \exp\left(-\frac{1}{2\sigma^2}\left|\left|\vec{y} - X\vec{\beta}\right|\right|_2^2\right). \tag{37}$$

The likelihood is the probability density function of $\vec{\mathbf{y}}$ evaluated at the observed data $\vec{y}$ and interpreted as a function of the coefficient vector $\vec{\beta}$,

$$\mathcal{L}_{\vec{y}}\left(\vec{\beta}\right) = \frac{1}{\sqrt{(2\pi)^n}} \exp\left(-\frac{1}{2}\left|\left|\vec{y} - X\vec{\beta}\right|\right|_2^2\right). \tag{38}$$

To find the ML estimate, we maximize the log likelihood

$$\vec{\beta}_{\text{ML}} = \arg\max_{\vec{\beta}} \mathcal{L}_{\vec{y}}\left(\vec{\beta}\right) \tag{39}$$

$$= \arg\max_{\vec{\beta}} \log \mathcal{L}_{\vec{y}}\left(\vec{\beta}\right) \tag{40}$$

$$= \arg\min_{\vec{\beta}} \left|\left|\vec{y} - X\vec{\beta}\right|\right|_2^2. \tag{41}$$

□

2.4 Analysis of the least-squares estimate

In this section we analyze the solution of the least-squares regression fit under the assumption that the data are indeed generated according to a linear model with additive noise,

$$\vec{y} := X\vec{\beta}^* + \vec{z}, \tag{42}$$

where $X \in \mathbb{R}^{n\times m}$ and $\vec{z} \in \mathbb{R}^n$. In that case, we can express the least-squares solution in terms of the true coefficients $\vec{\beta}^*$, the feature matrix X and the noise $\vec{z}$ applying Theorem 2.1. The estimation error equals

$$\begin{aligned} \vec{\beta}_{\text{LS}} - \vec{\beta}^* &= \left(X^T X\right)^{-1} X^T \left(X\vec{\beta}^* + \vec{z}\right) && (43) \\ &= \left(X^T X\right)^{-1} X^T \vec{z}, && (44) \end{aligned}$$

as long as X is full rank.

Equation (44) implies that if the noise is random and has zero mean, then the expected error is equal to zero. In statistics lingo, the least-squares estimate is *unbiased*, which means that the estimator is centered at the true coefficient vector $\vec{\beta}^*$.

Lemma 2.10 (Least-squares estimator is unbiased)**.** *If the noise* $\vec{\mathbf{z}}$ *is a random vector with zero mean, then*

$$\mathrm{E}\left(\vec{\boldsymbol{\beta}}_{\text{LS}} - \vec{\beta}^*\right) = 0. \tag{45}$$

Proof. By (44) and linearity of expectation

$$\mathrm{E}\left(\vec{\boldsymbol{\beta}}_{\text{LS}} - \vec{\beta}^*\right) = \left(X^T X\right)^{-1} X^T \mathrm{E}\left(\vec{\mathbf{z}}\right) = 0. \tag{46}$$

□

We can bound the error incurred by the least-squares estimate in terms of the noise and the singular values of the feature matrix X.

Theorem 2.11 (Least-squares error)**.** *For data of the form* (42)*, we have*

$$\frac{||\vec{z}||_2}{\sigma_1} \leq \left|\left|\vec{\beta}_{\text{LS}} - \vec{\beta}^*\right|\right|_2 \leq \frac{||\vec{z}||_2}{\sigma_p}, \tag{47}$$

as long as X *is full rank, where* σ_1 *and* σ_p *denote the largest and smallest singular value of* X *respectively.*

Proof. By (44)

$$\vec{\beta}_{\text{LS}} - \vec{\beta}^* = VS^{-1}U^T\vec{z}. \tag{48}$$

The smallest and largest singular values of $VS^{-1}U$ are $1/\sigma_1$ and $1/\sigma_p$ respectively so by Theorem 2.7 in Lecture Notes 2

$$\frac{||\vec{z}||_2}{\sigma_1} \leq \left|\left|VS^{-1}U^T\vec{z}\right|\right|_2 \leq \frac{||\vec{z}||_2}{\sigma_p}. \tag{49}$$

□

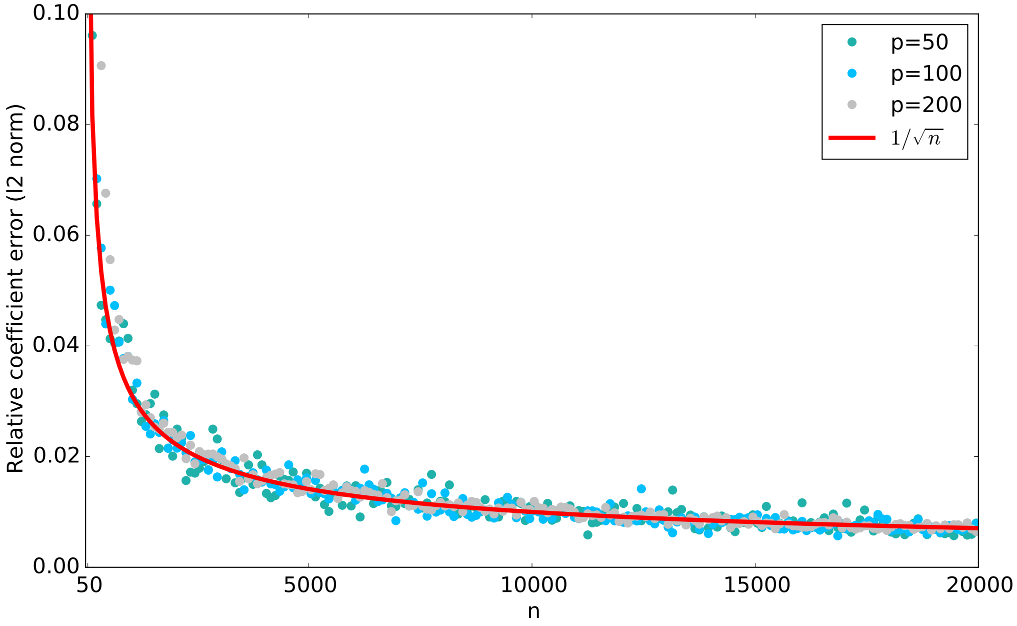


Figure 5: Relative ℓ_2-norm error of the least-squares coefficient estimate as n grows. The entries of $\mathbf{X}$, $\vec{\beta}^*$ and $\vec{\mathbf{z}}$ are sampled iid from a standard Gaussian distribution. The error scales as $1/\sqrt{n}$ as predicted by Theorem 2.12.

Let us assume that the norm of the noise $||\vec{z}||_2$ is fixed. In that case, by (48) the largest error occurs when $\vec{z}$ is aligned with $\vec{u}_p$, the singular vector corresponding to σ_p, whereas the smallest error occurs when $\vec{z}$ is aligned with $\vec{u}_1$, the singular vector corresponding to σ_1. To analyze what happens in a *typical* linear-regression problem, we can assume that X and $\vec{z}$ are sampled from a Gaussian distribution. The following theorem shows that in this case, the ratio between the norms of the error and the noise (or equivalently the error when the norm of the noise is fixed to one) concentrates around $\sqrt{p/n}$. In particular, for a fixed number of features it decreases as $1/\sqrt{n}$ with the number of available data, becoming arbitrarily small as $n \to \infty$. This is illustrated by Figure 5, which shows the results of a numerical experiment that match the theoretical analysis very closely.

Theorem 2.12 (Non-asymptotic bound on least-squares error)**.** *Let*

$$\vec{\mathbf{y}} := \mathbf{X}\vec{\beta}^* + \vec{\mathbf{z}}, \tag{50}$$

where the entries of the $n \times p$ matrix $\mathbf{X}$ and the n-dimensional vector $\vec{\mathbf{z}}$ are iid standard Gaussians. The least-squares estimate satisfies

$$\sqrt{\frac{(1-\epsilon)}{(1+\epsilon)}}\sqrt{\frac{p}{n}} \leq \left|\left|\vec{\boldsymbol{\beta}}_{\text{LS}} - \vec{\beta}^*\right|\right|_2 \leq \sqrt{\frac{(1+\epsilon)}{(1-\epsilon)}}\sqrt{\frac{p}{n}} \tag{51}$$

with probability at least $1 - 1/p - 2\exp\left(-p\epsilon^2/8\right)$ as long as $n \geq 64p\log(12/\epsilon)/\epsilon^2$.

Proof. By the same argument used to derive (49), we have

$$\frac{\left|\left|\mathbf{U}^T\vec{\mathbf{z}}\right|\right|_2}{\boldsymbol{\sigma_1}} \leq \left|\left|\mathbf{V}\mathbf{S}^{-1}\mathbf{U}^T\vec{\mathbf{z}}\right|\right|_2 \leq \frac{\left|\left|\mathbf{U}^T\vec{\mathbf{z}}\right|\right|_2}{\boldsymbol{\sigma_p}}. \tag{52}$$

By Theorem 2.10 in Lecture Notes 3 with probability $1 - 2\exp\left(-p\epsilon^2/8\right)$

$$(1-\epsilon)\,p \leq \left|\left|\mathbf{U}^T\vec{\mathbf{z}}\right|\right|_2^2 \leq (1+\epsilon)\,p, \tag{53}$$

where $\mathbf{U}$ contains the left singular vectors of $\mathbf{X}$. By Theorem 3.7 in Lecture Notes 3 with probability $1 - 1/p$

$$\sqrt{n\,(1-\epsilon)} \leq \boldsymbol{\sigma_p} \leq \boldsymbol{\sigma_1} \leq \sqrt{n\,(1+\epsilon)} \tag{54}$$

as long as $n \geq 64p\log(12/\epsilon)/\epsilon^2$. The result follows from combining (52) with (53) and (54) which hold simultaneously with probability at least $1 - 1/p - 2\exp\left(-p\epsilon^2/8\right)$ by the union bound. □

3 Regularization

3.1 Noise amplification

Theorem 2.12 characterizes the performance of least-squares regression when the feature matrix is *well-conditioned*, which means that its smallest singular value is not too small with respect to the largest singular value.

Definition 3.1 (Condition number). *The condition number of a matrix $A \in \mathbb{R}^{n\times p}$, $n \geq p$, is equal to the ratio σ_1/σ_p of its largest and smallest singular values σ_1 and σ_p.*

In numerical linear algebra, a system of equations is said to be *ill conditioned* if the condition number is large. The reason is that perturbations aligned with the singular vector corresponding to the smallest singular value may be amplified dramatically when inverting the system. This is exactly what happens in linear regression problems when the feature matrix X is not well conditioned. The component of the noise that falls in the direction of the singular vector corresponding to the smallest singular value *blows up*, as proven in the following theorem.

Lemma 3.2 (Noise amplification). *Let $X \in \mathbb{R}^{n\times p}$ be a matrix such that m singular values are smaller than η and let*

$$\vec{\mathbf{y}} := X\vec{\beta}^* + \mathbf{z}, \tag{55}$$

where the entries of $\vec{\mathbf{z}}$ are iid standard Gaussians. Then, with probability at least $1-2\exp\left(-m\epsilon^2/8\right)$

$$\left|\left|\vec{\boldsymbol{\beta}}_{\text{LS}} - \vec{\beta}^*\right|\right|_2 \geq \frac{m\sqrt{1-\epsilon}}{\eta}. \tag{56}$$

Proof. Let $X = USV^T$ be the SVD of X, $\vec{u}_1, \ldots, \vec{u}_p$ the columns of U and $\sigma_1, \ldots, \sigma_p$ the singular values. By (44)

$$\left|\left|\vec{\beta}_{\text{LS}} - \vec{\beta}^*\right|\right|_2^2 = \left|\left|VS^{-1}U^T\vec{\mathbf{z}}\right|\right|_2^2 \tag{57}$$

$$= \left|\left|S^{-1}U^T\vec{\mathbf{z}}\right|\right|_2^2 \qquad V \text{ is an orthogonal matrix} \tag{58}$$

$$= \sum_i^p \frac{\left(\vec{u}_i^T\vec{\mathbf{z}}\right)^2}{\sigma_i^2} \tag{59}$$

$$\geq \frac{1}{\eta^2}\sum_i^m \left(\vec{u}_i^T\vec{\mathbf{z}}\right)^2 . \tag{60}$$

The result follows because $\sum_i^m \left(\vec{u}_i^T\vec{\mathbf{z}}\right)^2 \geq 1 - \epsilon$ with probability at least $1 - 2\exp\left(-m\epsilon^2/8\right)$ by Theorem 2.10 in Lecture Notes 3 . □

We illustrate noise amplification in least-squares regression through a simple example.

Example 3.3 (Noise amplification)**.** Consider a linear-regression problem with data of the form

$$\vec{y} := X\vec{\beta}^* + \vec{z}, \tag{61}$$

where

$$X := \begin{bmatrix} 0.212 & -0.099 \\ 0.605 & -0.298 \\ -0.213 & 0.113 \\ 0.589 & -0.285 \\ 0.016 & 0.006 \\ 0.059 & 0.032 \end{bmatrix}, \qquad \vec{\beta}^* := \begin{bmatrix} 0.471 \\ -1.191 \end{bmatrix}, \qquad \vec{z} := \begin{bmatrix} 0.066 \\ -0.077 \\ -0.010 \\ -0.033 \\ 0.010 \\ 0.028 \end{bmatrix}. \tag{62}$$

The ℓ_2 norm of the noise is 0.11. The feature matrix is ill conditioned, its condition number is 100,

$$X = USV^T = \begin{bmatrix} -0.234 & 0.427 \\ -0.674 & -0.202 \\ 0.241 & 0.744 \\ -0.654 & 0.350 \\ 0.017 & -0.189 \\ 0.067 & 0.257 \end{bmatrix} \begin{bmatrix} 1.00 & 0 \\ 0 & 0.01 \end{bmatrix} \begin{bmatrix} -0.898 & 0.440 \\ 0.440 & 0.898 \end{bmatrix}. \tag{63}$$

As a result, the component of $\vec{z}$ in the direction of the second singular vector is amplified by a factor of 100! By (44), the error in the coefficient estimate is

$$\begin{aligned}
\vec{\beta}_{\text{LS}} - \vec{\beta}^* &= V S^{-1} U^T \vec{z} && (64)\\
&= V \begin{bmatrix} 1.00 & 0 \\ 0 & 100.00 \end{bmatrix} U^T \vec{z} && (65)\\
&= V \begin{bmatrix} 0.058 \\ 3.004 \end{bmatrix} && (66)\\
&= \begin{bmatrix} 1.270 \\ 2.723 \end{bmatrix}, && (67)
\end{aligned}$$

so that the norm of the error satisfies

$$\frac{\left|\left|\vec{\beta}_{\text{LS}} - \vec{\beta}^*\right|\right|_2}{\left|\left|\vec{z}\right|\right|_2} = 27.00. \tag{68}$$

△

The feature matrix is ill conditioned if any subset of columns is close to being linearly dependent, since in that case there must be a vector that is *almost* in the null space of the matrix. This occurs when some of the feature vectors are highly correlated, a phenomenon known as *multicollinearity* in the statistics ling. The following lemma shows how two feature vectors being very correlated results in poor conditioning.

Lemma 3.4 (Proof in Section 5.2)**.** *For any matrix $X \in \mathbb{R}^{n \times p}$, with columns normalized to have unit ℓ_2 norm, if any two distinct columns X_i and X_j satisfy*

$$\langle X_i, X_j \rangle^2 \geq 1 - \epsilon^2 \tag{69}$$

then $\sigma_p \leq \epsilon$, where σ_p is the smallest singular value of X.

3.2 Ridge regression

As described in the previous section, if the feature matrix is ill conditioned, then small shifts in the data produce large changes in the least-squares solution. In particular, some of the coefficients may blow up due to noise amplification. In order to avoid this, we can add a term penalizing the norm of the coefficient vector to the least-squares cost function. The aim is to promote solutions that yield a good fit with small coefficients. Incorporating prior assumptions on the desired solution– in this case that the coefficients should not be too large– is called *regularization*. Least-squares regression combined with ℓ_2-norm regularization is called ridge regression in statistics and Tikhonov regularization in the inverse-problems literature.

Definition 3.5 (Ridge regression / Tikhonov regularization)**.** *For any $X \in \mathbb{R}^{n \times p}$ and $\vec{y} \in \mathbb{R}^p$ the ridge-regression estimate is the minimizer of the optimization problem*

$$\vec{\beta}_{\text{ridge}} := \arg\min_{\vec{\beta}} \left|\left|\vec{y} - X\vec{\beta}\right|\right|_2^2 + \lambda \left|\left|\vec{\beta}\right|\right|_2^2, \tag{70}$$

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

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

Theorem 3.6 (Ridge-regression estimate)**.** *For any $X \in R^{n\times p}$ and $\vec{y} \in \mathbb{R}^n$ we have*

$$\vec{\beta}_{\text{ridge}} := \left(X^T X + \lambda I\right)^{-1} X^T \vec{y}. \tag{71}$$

Proof. The ridge-regression estimate is the solution to a modified least-squares problem

$$\vec{\beta}_{\text{ridge}} = \arg\min_{\vec{\beta}} \left\| \begin{bmatrix} \vec{y} \\ 0 \end{bmatrix} - \begin{bmatrix} X \\ \sqrt{\lambda} I \end{bmatrix} \vec{\beta} \right\|_2^2 . \tag{72}$$

By Theorem 2.1 the solution equals

$$\vec{\beta}_{\text{ridge}} := \left(\begin{bmatrix} X \\ \sqrt{\lambda} I \end{bmatrix}^T \begin{bmatrix} X \\ \sqrt{\lambda} I \end{bmatrix} \right)^{-1} \begin{bmatrix} X \\ \sqrt{\lambda} I \end{bmatrix}^T \begin{bmatrix} \vec{y} \\ 0 \end{bmatrix} \tag{73}$$

$$= \left(X^T X + \lambda I\right)^{-1} X^T \vec{y}. \tag{74}$$

□

When $\lambda \to 0$ then $\vec{\beta}_{\text{ridge}}$ converges to the least-squares estimator. When $\lambda \to \infty$, it converges to zero.

The approximation $X\vec{\beta}_{\text{ridge}}$ corresponding to the ridge-regression estimate is no longer the orthogonal projection of the data onto the column space of the feature matrix. It is a modified projection where the component of the data in the direction of each left singular vector of the feature matrix is shrunk by a factor of $\sigma_i^2 / \left(\sigma_i^2 + \lambda\right)$ where σ_i is the corresponding singular value. Intuitively, this reduces the influence of the directions corresponding to the smaller singular values which are the ones responsible for more noise amplification.

Corollary 3.7 (Modified projection)**.** *For any $X \in R^{n\times p}$ and $\vec{y} \in \mathbb{R}^n$ we have*

$$\vec{y}_{\text{ridge}} := X\vec{\beta}_{\text{ridge}} \tag{75}$$

$$= \sum_{i=1}^{p} \frac{\sigma_i^2}{\sigma_i^2 + \lambda} \langle \vec{y}, \vec{u}_i \rangle \vec{u}_i, \tag{76}$$

where $\vec{u}_1, \ldots, \vec{u}_p$ are the left singular vectors of X and $\sigma_1 \geq \ldots \geq \sigma_p$ the corresponding singular values.

Proof. Let $X = USV^T$ be the SVD of X. By the theorem,

$$X\vec{\beta}_{\text{ridge}} := X \left(X^T X + \lambda I\right)^{-1} X^T \vec{y} \tag{77}$$

$$= USV^T \left(VS^2V^T + \lambda VV^T\right)^{-1} VSU^T \vec{y} \tag{78}$$

$$= USV^T V \left(S^2 + \lambda I\right)^{-1} V^T VSU^T \vec{y} \tag{79}$$

$$= US \left(S^2 + \lambda I\right)^{-1} SU^T \vec{y}, \tag{80}$$

since V is an orthogonal matrix. □

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

Theorem 3.8 (Ridge-regression estimate). *If $\vec{y} := X\vec{\beta}^* + \vec{z}$, where $X \in \mathbb{R}^{n \times p}$, $\vec{z} \in \mathbb{R}^n$ and $\vec{\beta}^* \in \mathbb{R}^p$, then the solution of Problem (70) is equal to*

$$\vec{\beta}_{\text{ridge}} = V \begin{bmatrix} \frac{\sigma_1^2}{\sigma_1^2+\lambda} & 0 & \cdots & 0 \\ 0 & \frac{\sigma_2^2}{\sigma_2^2+\lambda} & \cdots & 0 \\ & & \cdots & \\ 0 & 0 & \cdots & \frac{\sigma_p^2}{\sigma_p^2+\lambda} \end{bmatrix} V^T \vec{\beta}^* + V \begin{bmatrix} \frac{\sigma_1}{\sigma_1^2+\lambda} & 0 & \cdots & 0 \\ 0 & \frac{\sigma_2}{\sigma_2^2+\lambda} & \cdots & 0 \\ & & \cdots & \\ 0 & 0 & \cdots & \frac{\sigma_p}{\sigma_p^2+\lambda} \end{bmatrix} U^T \vec{z}, \tag{81}$$

where $X = USV^T$ is the SVD of X and $\sigma_1, \ldots, \sigma_p$ are the singular values.

Proof. By Theorem 2.1 the solution equals

$$\begin{aligned} \vec{\beta}_{\text{ridge}} &= \left(X^T X + \lambda I\right)^{-1} X^T \left(X\vec{\beta}^* + \vec{z}\right) && (82) \\ &= \left(VS^2V^T + \lambda VV^T\right)^{-1} \left(VS^2V^T\vec{\beta}^* + VSU^T\vec{z}\right) && (83) \\ &= V\left(S^2 + \lambda I\right)^{-1} V^T \left(VS^2V^T\vec{\beta}^* + VSU^T\vec{z}\right) && (84) \\ &= V\left(S^2 + \lambda I\right)^{-1} S^2V^T\vec{\beta}^* + V\left(S^2 + \lambda I\right)^{-1} SU^T\vec{z}, && (85) \end{aligned}$$

because V is an orthogonal matrix. □

If we consider the difference between the true coefficients $\vec{\beta}^*$ and the ridge-regression estimator, the term that depends on $\vec{\beta}^*$ is usually known as the *bias* of the estimate, whereas the term that depends on the noise is the *variance*. The reason is that if we model the noise as being random and zero mean, then the mean or bias of the ridge-regression estimator equals the first term and the variance is equal to the variance of the second term.

Corollary 3.9 (Bias of ridge-regression estimator). *If the noise vector $\vec{\mathbf{z}}$ is random and zero mean,*

$$\mathrm{E}\left(\vec{\boldsymbol{\beta}}_{\text{ridge}} - \vec{\beta}^*\right) = V \begin{bmatrix} \frac{\lambda}{\sigma_1^2+\lambda} & 0 & \cdots & 0 \\ 0 & \frac{\lambda}{\sigma_2^2+\lambda} & \cdots & 0 \\ & & \cdots & \\ 0 & 0 & \cdots & \frac{\lambda}{\sigma_p^2+\lambda} \end{bmatrix} V^T \vec{\beta}^*. \tag{86}$$

Proof. The result follows from the lemma and linearity of expectation. □

Increasing λ increases the bias, moving the mean of the estimator farther from the true value of the coefficients, but in exchange dampens the noise component. In statistics jargon, we introduce bias in order to reduce the variance of the estimator. Calibrating the regularization parameter allows us to adapt to the conditioning of the predictor matrix and the noise level in order to achieve a good tradeoff between both terms.

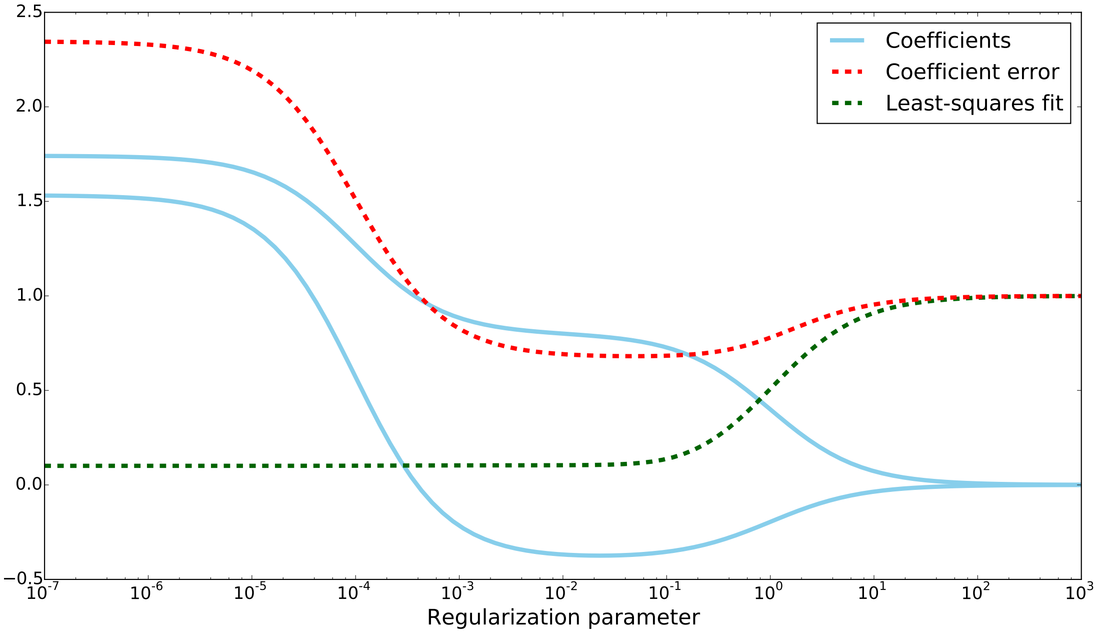


Figure 6: Coefficients in the ridge-regression model (blue) for different values of the regularization parameter λ (horizontal axis). The fit to the data improves as we reduce λ (green). The relative error of the coefficient estimate $\left|\left|\vec{\beta}^* - \vec{\beta}_{\text{ridge}}\right|\right|_2 / \left|\left|\vec{\beta}^*\right|\right|_2$ is equal to one when λ is large (because $\vec{\beta}_{\text{ridge}} = 0$), then it decreases as λ is reduced and finally it blows up due to noise amplification (red).

Example 3.10 (Noise amplification (continued))**.** By Theorem 3.8, the ridge-regression estimator for the regression problem in Example 3.3 equals

$$\vec{\beta}_{\text{ridge}} - \vec{\beta}^* = V \begin{bmatrix} \frac{\lambda}{1+\lambda} & 0 \\ 0 & \frac{\lambda}{0.01^2+\lambda} \end{bmatrix} V^T \vec{\beta}^* - V \begin{bmatrix} \frac{1}{1+\lambda} & 0 \\ 0 & \frac{0.01}{0.01^2+\lambda} \end{bmatrix} U^T \vec{z}, \tag{87}$$

The regularization λ should be set so to achieve a good balance between the two terms in the error. Setting $\lambda = 0.01$

$$\vec{\beta}_{\text{ridge}} - \vec{\beta}^* = -V \begin{bmatrix} 0.001 & 0 \\ 0 & 0.99 \end{bmatrix} V^T \vec{\beta}^* + V \begin{bmatrix} 0.99 & 0 \\ 0 & 0.99 \end{bmatrix} U^T \vec{z} \tag{88}$$

$$= \begin{bmatrix} 0.329 \\ 0.823 \end{bmatrix}. \tag{89}$$

The error is reduced significantly with respect to the least-squares estimate, we have

$$\frac{\left|\left|\vec{\beta}_{\text{ridge}} - \vec{\beta}^*\right|\right|_2}{\left|\left|\vec{z}\right|\right|_2} = 7.96. \tag{90}$$

Figure 6 shows the values of the coefficients for different values of the regularization parameter. They vary wildly due to the ill conditioning of the problem. The figure shows how least squares (to the left where $\lambda \to 0$) achieves the best fit to the data, but this does not result in a smaller error in the coefficient vector. $\lambda = 0.01$ achieves a good compromise. At that point the coefficients are smaller, while yielding a similar fit to the data as least squares. △

3.3 Ridge regression as maximum-a-posteriori estimation

From a probabilistic point of view, we can view the ridge-regression estimate as a maximum-a-posteriori (MAP) estimate. In Bayesian statistics, the MAP estimate is the mode of the posterior distribution of the parameter that we aim to estimate given the observed data.

Definition 3.11 (Maximum-a-posteriori estimator)**.** *The maximum-a-posteriori (MAP) estimator of a random vector of parameters $\vec{\boldsymbol{\beta}} \in \mathbb{R}^m$ given a realization of the data vector $\vec{y}$ is*

$$\vec{\beta}_{\text{MAP}}\left(\vec{y}\right) := \arg\max_{\vec{\beta}} f_{\vec{\boldsymbol{\beta}}\,|\,\vec{\mathbf{y}}}\left(\vec{\beta}\,|\,\vec{y}\right), \tag{91}$$

where $f_{\vec{\boldsymbol{\beta}}\,|\,\vec{\mathbf{y}}}$ is the conditional pdf of the parameter $\vec{\boldsymbol{\beta}}$ given the data $\vec{\mathbf{y}}$.

In contrast to ML estimation, the parameters of interest (in our case the regression coefficients) are modeled as random variables, not as deterministic quantities. This allows us to incorporate prior assumptions about them through their marginal distribution. Ridge regression is equivalent to modeling the distribution of the coefficients as an iid Gaussian random vector.

Lemma 3.12 (Proof in Section 5.3)**.** *Let $\vec{y} \in \mathbb{R}^n$ be a realization of a random vector*

$$\vec{\mathbf{y}} := X\vec{\boldsymbol{\beta}} + \vec{\mathbf{z}}, \tag{92}$$

where $\vec{\boldsymbol{\beta}}$ and $\vec{\mathbf{z}}$ are iid Gaussian random vectors with mean zero and variance σ_1^2 and σ_2^2, respectively. If $X \in \mathbb{R}^{n\times m}$ is known, then the MAP estimate of $\vec{\beta}$ is equal to the ridge-regression estimate

$$\vec{\beta}_{\text{MAP}} = \arg\min_{\vec{\beta}} \left|\left|\vec{y} - X\vec{\beta}\right|\right|_2^2 + \lambda \left|\left|\vec{\beta}\right|\right|_2^2, \tag{93}$$

where $\lambda := \sigma_2^2/\sigma_1^2$.

3.4 Cross validation

An important issue when applying ridge regression, and also other forms of regularization, is how to calibrate the regularization parameter λ. With real data, we do not know the true value of the coefficients as in Example 3.3 (otherwise we wouldn't need to do regression in the first place!). In addition, we cannot rely on how well the model fits the data, since this will always occur for $\lambda = 0$, which can lead to overfitting and noise amplification. However, we can evaluate the fit achieved by the model on *new* data, different from the ones used to estimate the regression coefficients. If the fit is accurate, this is a strong indication that the model is not overfitting the noise. Calibrating the regularization parameter using a different set of data is known as cross validation.

Algorithm 3.13 (Cross validation)**.** *Given a set of examples*

$$\left(y^{(1)}, \vec{x}^{\,(1)}\right), \left(y^{(2)}, \vec{x}^{\,(2)}\right), \ldots, \left(y^{(n)}, \vec{x}^{\,(n)}\right), \tag{94}$$

which are centered and normalized, to determine the best value for λ we:

1. *Partition the data into a* training *set $X_{\text{train}} \in \mathbb{R}^{n_{\text{train}} \times p}$, $\vec{y}_{\text{train}} \in \mathbb{R}^{n_{\text{train}}}$ and a* validation *set $X_{\text{val}} \in \mathbb{R}^{n_{\text{val}} \times p}$, $\vec{y}_{\text{val}} \in \mathbb{R}^{n_{\text{val}}}$, such that $n_{\text{train}} + n_{\text{val}} = n$.*
2. *Fit the model using the training set for every λ in a set Λ (usually a logarithmic grid of values)*

   $$\vec{\beta}_{\text{ridge}}(\lambda) := \arg\min_{\vec{\beta}} \left|\left|\vec{y}_{\text{train}} - X_{\text{train}}\vec{\beta}\right|\right|_2^2 + \lambda \left|\left|\vec{\beta}\right|\right|_2^2 \tag{95}$$

   and evaluate the fitting error on the validation set

   $$\text{err}(\lambda) := \left|\left|\vec{y}_{\text{train}} - X_{\text{train}}\vec{\beta}_{\text{ridge}}(\lambda)\right|\right|_2^2. \tag{96}$$

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

   $$\lambda_{\text{cv}} := \arg\min_{\lambda \in \Lambda} \text{err}(\lambda). \tag{97}$$

In practice, more sophisticated cross-validation procedures are applied to make an efficient use of the data. For example, in *k-fold* cross validation we randomly partition the data into k sets of equal size. Then we evaluate the fitting error k times, each time using one of the k sets as the validation set and the rest as the training set.

Finally, it is important to note that if we have used the validation set to fit the regularization parameter, we *cannot* use it to evaluate our results. This wouldn't be fair, since we have calibrated one the parameter to do well precisely on those data! It is crucial to evaluate the model on a test set that is completely different from both the training and validation tests.

Example 3.14 (Prediction of house prices)**.** In this example we consider the problem of predicting the price of a house[2]. The features that we consider are:

1. Area of the living room.
2. Condition (an integer between 1 and 5 evaluating the state of the house).
3. Grade (an integer between 7 and 12 evaluating the house).
4. Area of the house without the basement.
5. Area of the basement.
6. The year it was built.
7. Latitude.
8. Longitude.

[2] The data are available at http://www.kaggle.com/harlfoxem/housesalesprediction

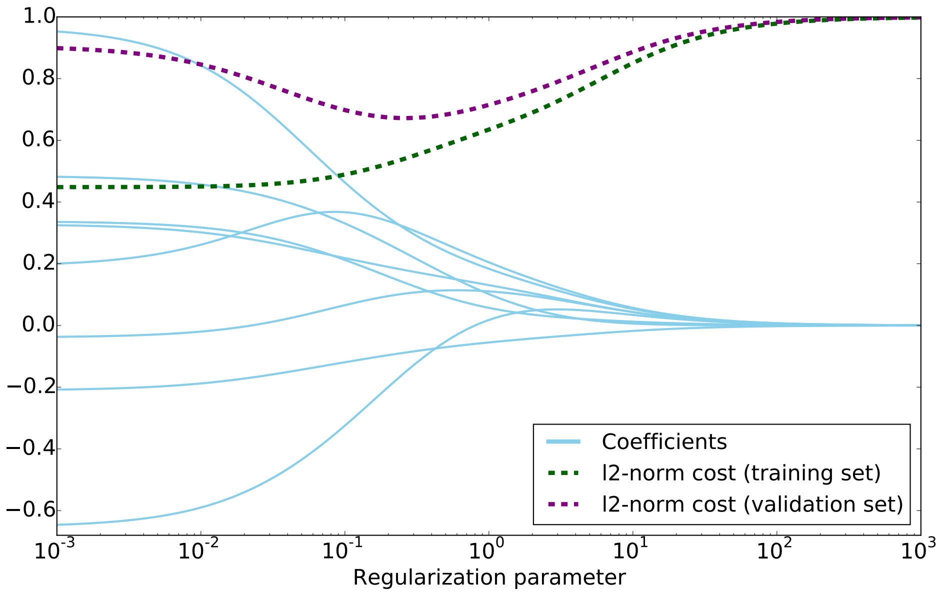


Figure 7: Coefficients in the ridge-regression model (blue) for different values of the regularization parameter λ (horizontal axis). The relative ℓ_2-norm error evaluated on the training data is shown in green. The relative ℓ_2-norm error evaluated on the validation data is shown in purple.

9. Average area of the living room of the houses within 15 blocks.

We use 15 houses to train the data, a validation set of 15 houses to calibrate the regularization parameter of the ridge regression model and a test set of 15 houses to evaluate the results. The feature matrix has significant correlations (the condition number is equal to 9.94), so we decide to apply ridge regression. Figure 7 shows the value of the coefficients obtained by fitting the model to the training set for different values of λ. It also shows the corresponding relative ℓ_2-norm fit

$$\frac{\left|\left|\vec{y} - X\vec{\beta}_{\text{ridge}}\right|\right|_2}{\left|\left|\vec{y}\right|\right|_2} \tag{98}$$

to the training and validation sets. For small λ the model fits the training set much better than the validation set, a clear indication that it is overfitting. The validation-set error is minimized for $\lambda = 0.27$. For that value the error is 0.672 on the validation set and 0.799 on the test set. In contrast, the error of the least-squares estimator is 0.906 on the validation set and 1.186 on the test set. Figure 8 shows the prices estimated by the least-squares and the ridge-regression models plotted against the true prices. The least-squares estimate is much more accurate on the training set than on the validation and test sets due to overfitting. Adding regularization and computing a ridge-regression estimate substantially improves the prediction results on the test set. △

4 Classification

In this section, we consider the problem of classification. The goal is to learn a model that assigns one of several predefined categories to a set of examples, represented by the values of certain

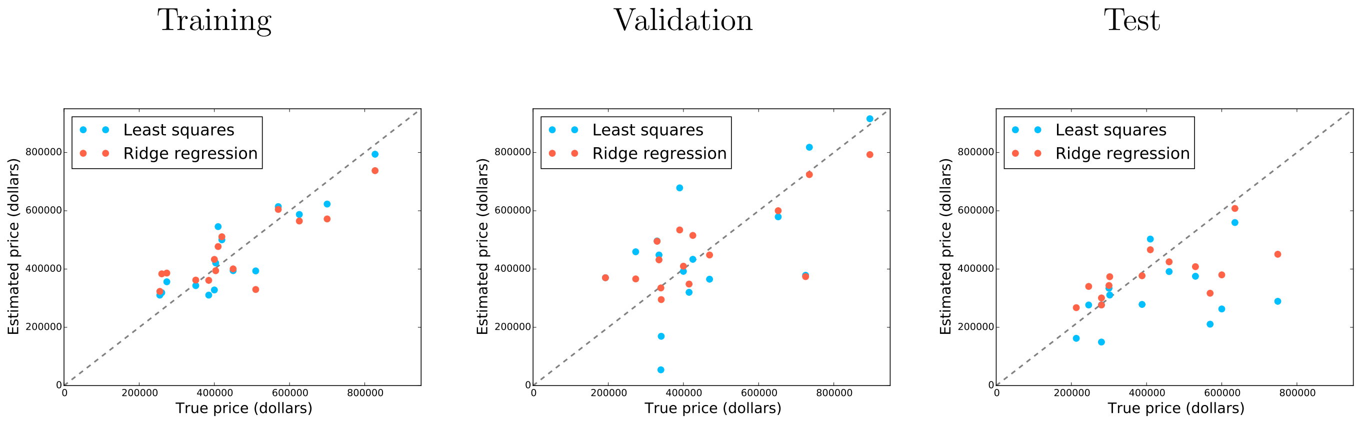


Figure 8: Prices estimated by the least-squares (blue) and the ridge-regression (orange) models plotted against the true prices for the training, validation and test sets.

features, as in the case of regression. To be more precise, we have available n examples of category labels and their corresponding features

$$\left(y^{(1)}, \vec{x}^{\,(1)}\right), \left(y^{(2)}, \vec{x}^{\,(2)}\right), \ldots, \left(y^{(n)}, \vec{x}^{\,(n)}\right). \tag{99}$$

The label $y^{(i)}$ indicates what category example i belongs to. Here, we consider the simple case where there are only two categories and set the labels to equal either 0 or 1. Our aim is to predict the label $y^{(i)} \in \{0, 1\}$ from p real-valued features $\vec{x}^{\,(i)} \in \mathbb{R}^p$. This is a regression problem, where the response is binary.

4.1 Perceptron

Inspired by linear regression, let us consider how to use a linear model to perform classification. A reasonable idea is to fit a vector of coefficients $\vec{\beta}$ such that the label is predicted to equal 1 if $\langle \vec{x}^{\,(i)}, \vec{\beta} \rangle$ is larger than a certain quantity, and 0 if it is smaller. This requires finding $\vec{\beta} \in \mathbb{R}^p$ and β_0 such that

$$y^{(i)} = \begin{cases} 1 & \text{if } \beta_0 + \langle \vec{x}^{\,(i)}, \vec{\beta} \rangle > 0 \\ 0 & \text{otherwise} \end{cases} \tag{100}$$

for as many $1 \leq i \leq n$ as possible. This method is called the *perceptron* algorithm. The model is fit by considering each feature vector sequentially and updating $\vec{\beta}$ if the current classification is wrong. This method is guaranteed to converge if the data are *linearly separable*, i.e. if there is a hyperplane in the p-dimensional feature space $\mathbb{R}^p$ separating the two classes. However, if this is not the case, then the method becomes unstable.

4.2 Logistic regression

Logistic regression is an example of a *generalized linear model*. Generalized linear models extend the linear regression paradigm by incorporating a *link function* that performs an entry-wise non-

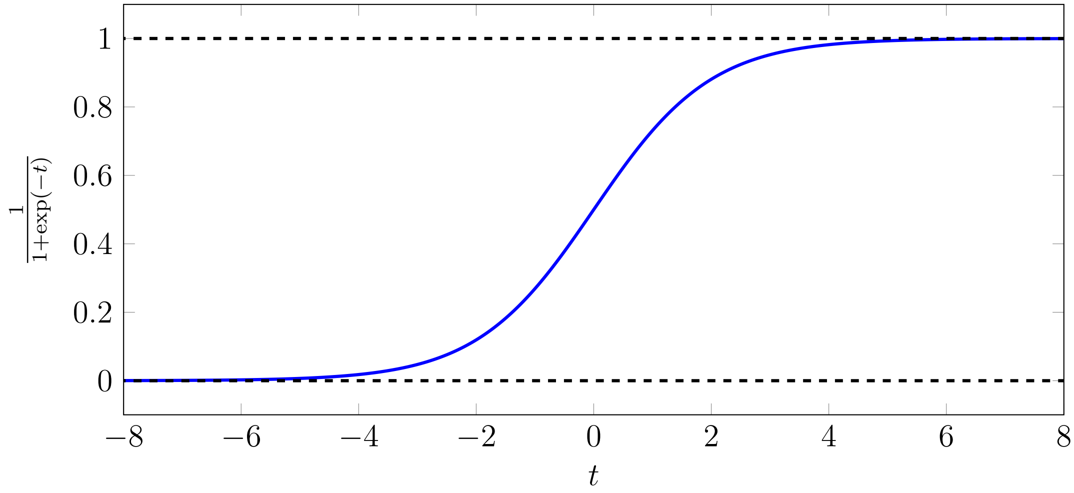


Figure 9: The logistic function used as a link function in logistic regression.

linear transformation of the output of a linear model. In the case of logistic regression, this link function is the *logistic function*

$$g\left(t\right) := \frac{1}{1+\exp(-t)}, \tag{101}$$

depicted in Figure 9. The output of g is always between 0 and 1. We can interpret the function as a smoothed version of the step function used by the perceptron algorithm, as it maps large values to 1 and small values to 0.

The logistic-regression model is of the form

$$y^{(i)} \approx g\left(\beta_0 + \langle \vec{x}^{\,(i)}, \vec{\beta} \rangle\right). \tag{102}$$

To simplify notation, from now on we assume that one of the feature vectors is equal to a constant, so that β_0 is included in $\vec{\beta}$. The logistic-regression estimator is obtained by calibrating $\vec{\beta}$ in order to optimize the fit to the training data. This can be achieved by maximizing the log-likelihood function derived in the following theorem.

Theorem 4.1 (Logistic-regression cost function)**.** *Assume that $y^{(1)}$, ..., $y^{(n)}$ are independent samples from Bernoulli random variables with parameter*

$$p_{\mathbf{y^{(i)}}}\left(1\right) := g\left(\langle \vec{x}^{\,(i)}, \vec{\beta} \rangle\right), \tag{103}$$

where the vectors $\vec{x}^{\,(1)}$, ..., $\vec{x}^{\,(n)} \in \mathbb{R}^p$ are known. The maximum-likelihood estimate of $\vec{\beta}$ given $y^{(1)}$, ..., $y^{(n)}$ is equal to

$$\vec{\beta}_{\text{ML}} := \sum_{i=1}^{n} y^{(i)} \log g\left(\langle \vec{x}^{\,(i)}, \vec{\beta} \rangle\right) + \left(1 - y^{(i)}\right) \log\left(1 - g\left(\langle \vec{x}^{\,(i)}, \vec{\beta} \rangle\right)\right). \tag{104}$$

Proof. The likelihood $\mathcal{L}\left(\vec{\beta}\right)$ is defined as the joint pmf of the random variables $\mathbf{y^{(1)}}$, ..., $\mathbf{y^{(n)}}$ interpreted as a function of the coefficient vector. Due to the independence assumption,

$$\mathcal{L}\left(\vec{\beta}\right) := p_{\mathbf{y^{(1)}},\ldots,\mathbf{y^{(n)}}}\left(y^{(1)},\ldots,y^{(n)}\right) \tag{105}$$

$$= \prod_{i=1}^{n} g\left(\langle \vec{x}^{\,(i)}, \vec{\beta}\rangle\right)^{y^{(i)}} \left(1 - g\left(\langle \vec{x}^{\,(i)}, \vec{\beta}\rangle\right)\right)^{1-y^{(i)}}. \tag{106}$$

Maximizing this nonnegative function is the same as maximizing its logarithm, so the proof is complete. □

Even though it is quite implausible that the probabilistic assumptions assumed in this theorem actually hold in practice, the corresponding log-likelihood function is very useful. It penalizes classification errors in a smooth way and is easy to optimize (as we will see later on).

Definition 4.2 (Logistic-regression estimator)**.** *Given a set of examples* $\left(y^{(1)}, \vec{x}^{\,(1)}\right)$, $\left(y^{(2)}, \vec{x}^{\,(2)}\right)$, ..., $\left(y^{(n)}, \vec{x}^{\,(n)}\right)$, *we define the logistic-regression coefficient vector as*

$$\vec{\beta}_{\text{LR}} := \sum_{i=1}^{n} y^{(i)} \log g\left(\langle \vec{x}^{\,(i)}, \vec{\beta}\rangle\right) + \left(1 - y^{(i)}\right) \log\left(1 - g\left(\langle \vec{x}^{\,(i)}, \vec{\beta}\rangle\right)\right), \tag{107}$$

where we assume that one of the features is always equal to one, so we don't have to fit an intercept. For a new feature vector $\vec{x}$ *the logistic-regression prediction is*

$$y_{\text{LR}} := \begin{cases} 1 & \text{if } g\left(\langle \vec{x}, \vec{\beta}_{LR}\rangle\right) \geq 0, \\ 0 & \text{otherwise.} \end{cases} \tag{108}$$

The value $g\left(\langle \vec{x}, \vec{\beta}_{LR}\rangle\right)$ *can be interpreted as the probability under the model that the label of the example equals 1.*

Example 4.3 (Flower classification)**.** The *Iris* data set was compiled by the statistician Ronald Fisher in 1936. It contains examples of three species of flowers, together with measurements of the length and width of their sepal and petal. In this example, we consider the problem of distinguishing between two of the species using only the sepal lengths and widths.

We assume that we just have access to 5 examples of *Iris setosa* (label 0) with sepal lengths 5.4, 4.3, 4.8, 5.1 and 5.7, and sepal widths 3.7, 3, 3.1, 3.8 and 3.8, and to 5 examples of *Iris versicolor* (label 1) with sepal lengths 6.5, 5.7, 7, 6.3 and 6.1, and sepal widths 2.8, 2.8, 3.2, 2.3 and 2.8. We want to classify two new examples: one has a sepal length of 5.1 and width 3.5, the other has length 5 and width 2. $\beta_0 = 2.06$. After centering and normalizing the data set (note that we ignore the labels to center and normalize), we fit a logistic regression model, where the coefficient vector equals

$$\vec{\beta}_{\text{LR}} = \begin{bmatrix} 32.1 \\ -29.6 \end{bmatrix} \tag{109}$$

and the intercept β_0 equals 2.06. The coefficients suggest that *versicolor* has larger sepal length than *setosa*, but smaller sepal width. The following table shows the values of the features, their inner product with $\vec{\beta}_{\text{LR}}$ and the output of the logistic function.

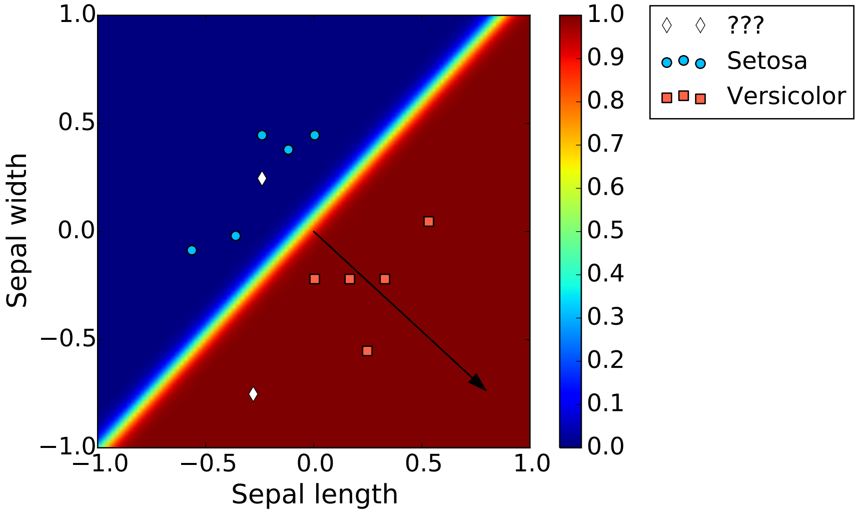


Figure 10: The data used in Example 4.3 is plotted in different colors depending on the corresponding flower species. The direction of $\vec{\beta}_{\text{LR}}$ is shown as a black arrow. The heat map corresponds to the value of $g\left(\langle\vec{x},\vec{\beta}_{\text{LR}}\rangle+\beta_0\right)$ at every point. The two new examples are depicted as white diamonds.

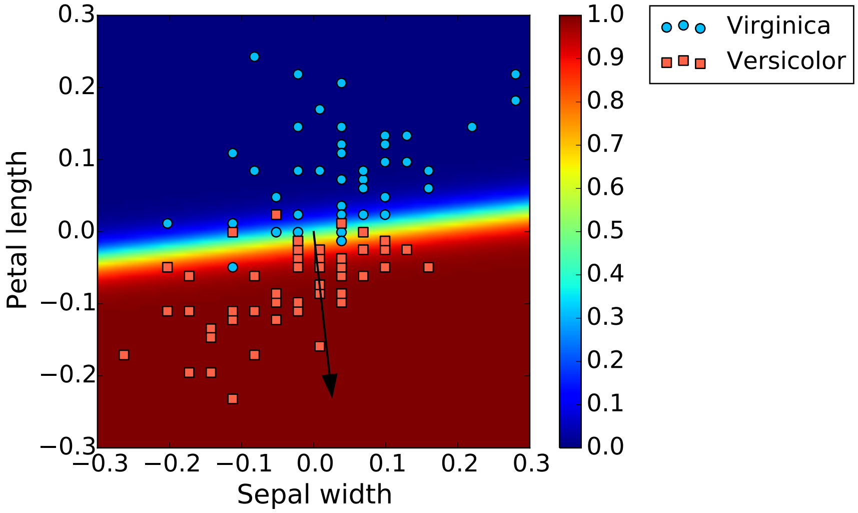


Figure 11: The data from the Iris data set plotted in different colors depending on the corresponding flower species. The direction of $\vec{\beta}_{\text{LR}}$ is shown as a black arrow. The heat map corresponds to the value of $g\left(\langle\vec{x},\vec{\beta}_{\text{LR}}\rangle+\beta_0\right)$ at every point.

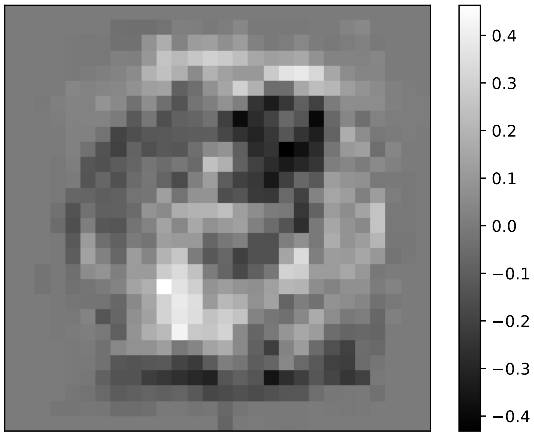


Figure 12: The coefficient vector $\vec{\beta}_{\text{LR}}$ obtained by fitting a logistic-regression model to distinguish between 6 and 9. The vector is reshaped so that each coefficient is shown at the position of the corresponding pixel.

i	1	2	3	4	5	6	7	8	9	10
$\vec{x}^{(i)}[1]$	-0.12	-0.56	-0.36	-0.24	0.00	0.33	0.00	0.53	0.25	0.17
$\vec{x}^{(i)}[2]$	0.38	-0.09	-0.02	0.45	0.45	-0.22	-0.22	0.05	-0.05	-0.22
$\langle \vec{x}^{(i)}, \vec{\beta}_{\text{LR}} \rangle + \beta_0$	-12.9	-13.5	-8.9	-18.8	-11.0	19.1	8.7	17.7	26.3	13.9
$g\left(\langle \vec{x}^{(i)}, \vec{\beta}_{\text{LR}} \rangle + \beta_0\right)$	0.00	0.00	0.00	0.00	0.00	1.00	1.00	1.00	1.00	1.00

Figure 10 shows the data, which are linearly separable, the direction of $\vec{\beta}_{\text{LR}}$ (black arrow) and a heat map of values for $g\left(\langle \vec{x}, \vec{\beta}_{\text{LR}} \rangle\right)$ which shows are assigned to what category and with how much certainty. The two new examples are depicted as white diamonds, the first is assigned to *setosa* and the second to *versicolor* with almost total certainty. Both decisions are correct.

Figure 11 shows the result of trying to classify between *Iris virginica* and *Iris versicolor* based on petal length and sepal width. In this case the data is not linearly separable, but the logistic-regression model still partitions the space in a way that approximately separates the two classes. The value of the likelihood $g\left(\langle \vec{x}, \vec{\beta}_{LR} \rangle\right)$ allows us to quantify the certainty with which the model classifies each example. Note that the examples that are misclassified are assigned low values. △

Example 4.4 (Digit classification)**.** In this example we use the MNIST data set[3] to illustrate image classification. We consider the task of distinguishing a digit from another. The feature vector $\vec{x}_i$ contains the pixel values of an image of a 6 ($\vec{y}_i = 1$) or a 9 ($\vec{y}_i = 0$). We use 2000 training examples to fit a logistic regression model. The coefficient vector is shown in Figure 12, the intercept is equal to 0.053. The model manages to fit the training set perfectly. When tested on 2000 new examples, it achieves a test error rate of 0.006. Figure 13 shows some test examples and the corresponding probabilities assigned by the model. △

[3] Available at http://yann.lecun.com/exdb/mnist/

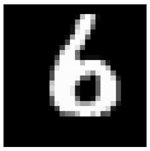
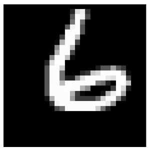
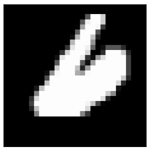
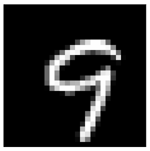
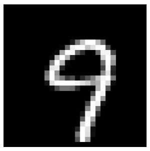
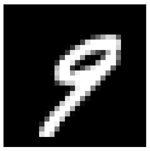
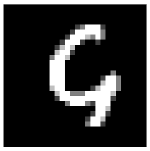
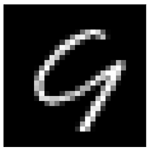
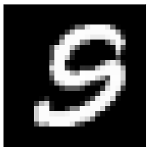
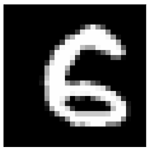
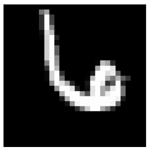
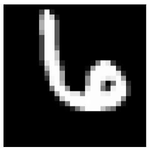
$\vec{x}$	$\vec{\beta}^T\vec{x}$	$g\left(\vec{\beta}^T\vec{x}+\beta_0\right)$	Pred.	True label	$\vec{x}$	$\vec{\beta}^T\vec{x}$	$g\left(\vec{\beta}^T\vec{x}+\beta_0\right)$	Pred.	True label
	20.88	1.00	6	6		18.22	1.00	6	6
	16.41	1.00	6	6		-14.71	0.00	9	9
	-15.83	0.00	9	9		-17.02	0.00	9	9
	7.612	0.9995	6	9		0.434	0.606	6	9
	7.822	0.9996	6	9		-5.984	0.0025	9	6
	-2.384	0.084	9	6		-1.164	0.238	9	6

Figure 13: Examples of digits in the MNIST data set along with the value of $\vec{\beta}^T\vec{x}+\beta_0$ and the probability assigned by the model.

5 Proofs

5.1 Proof of Lemma 2.2

To ease notation let $\widetilde{X} := X^{\,\text{cent}}$ and $\widetilde{x} := X^T\vec{1}$. Note that

$$\vec{y}^{\,\text{cent}} = \vec{y} - \frac{1}{n}\vec{1}\,\vec{1}^{\,T}\vec{y}, \tag{110}$$

$$\widetilde{X} = X - \frac{1}{n}\vec{1}\,\widetilde{x}^T. \tag{111}$$

By Theorem 2.1

$$\begin{bmatrix} \vec{\beta}_{\text{LS}} \\ \beta_{\text{LS},0} \end{bmatrix} = \left(\begin{bmatrix} X & \vec{1} \end{bmatrix}^T \begin{bmatrix} X & \vec{1} \end{bmatrix} \right)^{-1} \begin{bmatrix} X & \vec{1} \end{bmatrix}^T \vec{y} \tag{112}$$

$$= \begin{bmatrix} X^T X & \widetilde{x} \\ \widetilde{x}^T & n \end{bmatrix}^{-1} \begin{bmatrix} X^T\vec{y} \\ \vec{1}^{\,T}\vec{y} \end{bmatrix}. \tag{113}$$

We now apply the following lemma.

Lemma 5.1. *For any matrices $A \in \mathbb{R}^{m\times}$, let*

$$B = A - \frac{1}{n}\widetilde{x}\widetilde{x}^T \tag{114}$$

be invertible, then

$$\begin{bmatrix} A & \widetilde{x} \\ \widetilde{x}^T & n \end{bmatrix}^{-1} = \begin{bmatrix} B^{-1} & -\frac{1}{n}B^{-1}\widetilde{x} \\ -\frac{1}{n}\widetilde{x}^T B^{-1} & \frac{1}{n} + \frac{1}{n^2}\widetilde{x}^T B^{-1}\widetilde{x} \end{bmatrix} \tag{115}$$

Proof. One can check the result by multiplying the two matrices and verifying that the product is the identity. □

Setting $A := X^T X$, we have

$$B = X^T X - \frac{1}{n}\widetilde{x}\widetilde{x}^T \tag{116}$$

$$= \left(X - \frac{1}{n}\vec{1}\,\widetilde{x}^T \right)^T \left(X - \frac{1}{n}\vec{1}\,\widetilde{x}^T \right) \tag{117}$$

$$= \widetilde{X}^T\widetilde{X}. \tag{118}$$

As a result, by the lemma

$$\begin{bmatrix} \vec{\beta}_{\text{LS}} \\ \beta_{\text{LS},0} \end{bmatrix} = \begin{bmatrix} \left(\widetilde{X}^T\widetilde{X}\right)^{-1} & -\frac{1}{n}\left(\widetilde{X}^T\widetilde{X}\right)^{-1}\widetilde{x} \\ -\frac{1}{n}\widetilde{x}^T\left(\widetilde{X}^T\widetilde{X}\right)^{-1} & \frac{1}{n} + \frac{1}{n^2}\widetilde{x}^T\left(\widetilde{X}^T\widetilde{X}\right)^{-1}\widetilde{x} \end{bmatrix} \begin{bmatrix} X^T\vec{y} \\ \vec{1}^{\,T}\vec{y} \end{bmatrix} \tag{119}$$

$$= \begin{bmatrix} \left(\widetilde{X}^T\widetilde{X}\right)^{-1} X^T \left(\vec{y} - \frac{1}{n}\vec{1}\,\vec{1}^{\,T}\vec{y}\right) \\ -\frac{1}{n}\widetilde{x}^T\left(\widetilde{X}^T\widetilde{X}\right)^{-1} X^T \left(\vec{y} - \frac{1}{n}\vec{1}\vec{1}^{\,T}\vec{y}\right) + \frac{\vec{1}^{\,T}\vec{y}}{n} \end{bmatrix}, \tag{120}$$

which implies

$$X\vec{\beta}_{\text{LS}} + \beta_{\text{LS},0}\vec{1} = X\left(\widetilde{X}^T\widetilde{X}\right)^{-1} X^T\vec{y}^{\text{cent}} - \frac{1}{n}\vec{1}\widetilde{x}^T\left(\widetilde{X}^T\widetilde{X}\right)^{-1} X^T\vec{y}^{\text{cent}} + \text{av}\left(\vec{y}\right)\vec{1} \tag{121}$$

$$= \widetilde{X}\left(\widetilde{X}^T\widetilde{X}\right)^{-1} X^T\vec{y}^{\text{cent}} + \text{av}\left(\vec{y}\right)\vec{1} \tag{122}$$

$$= \widetilde{X}\left(\widetilde{X}^T\widetilde{X}\right)^{-1} \widetilde{X}^T\vec{y}^{\text{cent}} + \text{av}\left(\vec{y}\right)\vec{1}, \tag{123}$$

where the last inequality follows from

$$\widetilde{X}^T\vec{y}^{\text{cent}} = \left(X - \frac{1}{n}\vec{1}\,\vec{1}^T X\right)^T\left(\vec{y} - \frac{1}{n}\vec{1}\vec{1}^T\vec{y}\right) \tag{124}$$

$$= X^T\vec{y} - \frac{1}{n}X^T\vec{1}\vec{1}^T\vec{y} - \frac{1}{n}X^T\vec{1}\vec{1}^T\vec{y} + \frac{1}{n^2}X^T\vec{1}\vec{1}^T\vec{1}\vec{1}^T\vec{y} \tag{125}$$

$$= X^T\vec{y} - \frac{1}{n}X^T\vec{1}\vec{1}^T\vec{y} \tag{126}$$

$$= X^T\vec{y}^{\text{cent}}. \tag{127}$$

Since $\vec{\beta}_{\text{LS}}^{\text{cent}} = \left(\widetilde{X}^T\widetilde{X}\right)^{-1}\widetilde{X}^T\vec{y}^{\text{cent}}$ the proof is complete.

5.2 Proof of Lemma 3.4

The orthogonal projection of X_i onto the span of X_j equals

$$\mathcal{P}_{\text{span}(X_j)}\,X_i = \langle X_i, X_j\rangle\, X_j \tag{128}$$

so

$$\left|\left|\mathcal{P}_{\text{span}(X_j)}\,X_i\right|\right|_2^2 = \langle X_i, X_j\rangle^2\,||X_j||_2^2 = 1 - \epsilon^2 \tag{129}$$

and

$$\left|\left|\mathcal{P}_{\text{span}(X_j)^\perp}\,X_i\right|\right|_2^2 = ||X_i||_2^2 - \left|\left|\mathcal{P}_{\text{span}(X_j)}\,X_i\right|\right|_2^2 = \epsilon^2. \tag{130}$$

Consider the unit norm vector $\vec{w} \in \mathbb{R}^p$

$$\vec{w}[l] := \begin{cases} \frac{1}{\sqrt{2}} & \text{if } l = i \\ -\frac{1}{\sqrt{2}} & \text{if } l = j \\ 0 & \text{otherwise.} \end{cases} \tag{131}$$

We have

$$\begin{aligned}
||X\vec{w}||_2^2 &= \frac{1}{2}\,||X_i - X_j||_2^2 && (132)\\
&= \frac{1}{2}\left|\left|\mathcal{P}_{\text{span}(X_j)}\, X_i + \mathcal{P}_{\text{span}(X_j)^\perp}\, X_i - X_j\right|\right|_2^2 && (133)\\
&= \frac{1}{2}\left|\left|\mathcal{P}_{\text{span}(X_j)}\, X_i - X_j\right|\right|_2^2 + \frac{1}{2}\left|\left|\mathcal{P}_{\text{span}(X_j)^\perp}\, X_i\right|\right|_2^2 && (134)\\
&= \frac{1}{2}\,||\langle X_i, X_j\rangle\, X_j - X_j||_2^2 + \frac{\epsilon^2}{2} && (135)\\
&= \frac{\langle X_i, X_j\rangle^2}{2}\,||X_j||_2^2 + \frac{\epsilon^2}{2} && (136)\\
&= \epsilon^2. && (137)
\end{aligned}$$

Finally by Theorem 2.7 in Lecture Notes 2

$$\sigma_p = \min_{||v||_2=1} ||X\vec{v}||_2 \geq ||X\vec{w}||_2 = \epsilon. \tag{138}$$

5.3 Proof of Lemma 3.12

By Bayes' rule, the posterior pdf of $\vec{\boldsymbol{x}}$ given $\vec{\mathbf{y}}$ is equal to

$$f_{\vec{\boldsymbol{\beta}}\,|\,\vec{\mathbf{y}}}\left(\vec{\beta}\,|\,\vec{y}\right) = \frac{f_{\vec{\boldsymbol{\beta}},\vec{\mathbf{y}}}\left(\vec{\beta},\vec{y}\right)}{f_{\vec{\mathbf{y}}}\left(\vec{y}\right)} \tag{139}$$

so for fixed $\vec{y}$

$$\begin{aligned}
\arg\max_{\vec{\beta}} f_{\vec{\boldsymbol{\beta}}\,|\,\vec{\mathbf{y}}}\left(\vec{\beta}\,|\,\vec{y}\right) &= \arg\max_{\vec{\beta}} f_{\vec{\boldsymbol{\beta}},\vec{\mathbf{y}}}\left(\vec{\beta},\vec{y}\right) && (140)\\
&= \arg\max_{\vec{\beta}} f_{\vec{\boldsymbol{\beta}}}\left(\vec{\beta}\right) f_{\vec{\mathbf{y}}\,|\,\vec{\boldsymbol{\beta}}}\left(\vec{y}\,|\,\vec{\beta}\right). && (141)
\end{aligned}$$

Since all the quantities are nonnegative, we can take logarithms

$$\arg\max_{\vec{\beta}} f_{\vec{\boldsymbol{\beta}}\,|\,\vec{\mathbf{y}}}\left(\vec{\beta}\,|\,\vec{y}\right) = \arg\max_{\vec{\beta}} \log f_{\vec{\boldsymbol{\beta}}}\left(\vec{\beta}\right) + \log f_{\vec{\mathbf{y}}\,|\,\vec{\boldsymbol{\beta}}}\left(\vec{y}\,|\,\vec{\beta}\right). \tag{142}$$

Since, conditioned on $\vec{\boldsymbol{\beta}} = \vec{\beta}$, $\vec{\mathbf{y}}$ is iid Gaussian with mean $X\vec{\beta}$ and variance σ_2^2

$$\begin{aligned}
\log f_{\vec{\mathbf{y}}\,|\,\vec{\boldsymbol{\beta}}}\left(\vec{y}\,|\,\vec{\beta}\right) &= \log \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma_2} \exp\left(-\frac{1}{2\sigma_2^2}\left(\vec{y}[i] - \left(X\vec{\beta}\right)[i]\right)^2\right) && (143)\\
&= \log \frac{1}{\sqrt{(2\pi)^n}\sigma_2^n} \exp\left(-\frac{1}{2\sigma_2^2}\left|\left|\vec{y} - X\vec{\beta}\right|\right|_2^2\right) && (144)\\
&= -\frac{1}{2\sigma_2^2}\left|\left|\vec{y} - X\vec{\beta}\right|\right|_2^2 + \log\frac{1}{\sqrt{(2\pi)^n}\sigma_2^n}. && (145)
\end{aligned}$$

Similarly,

$$\log f_{\boldsymbol{\vec{\beta}}}\left(\vec{\beta}\right) = -\frac{1}{2\sigma_1^2}\left|\left|\vec{\beta}\right|\right|_2^2 + \log\frac{1}{\sqrt{(2\pi)^n}\sigma_1^n}. \tag{146}$$

Setting

$$\lambda := \frac{\sigma_2^2}{\sigma_1^2}, \tag{147}$$

combining (142), (145) and (146) and ignoring the terms that do not depend on $\vec{\beta}$ completes the proof.