Advanced Machine Learning **Fall 2020**

Lecture 4: Regression, bias-variance tradeoff

Note: *LaTeX template courtesy of UC Berkeley EECS dept.*

Disclaimer: *These notes are adapted from ETH's Advanced Machine Learning Course, "Linear Regression via Maximization of the Likelihood, COS 234, Princeton", "The Elements of Statistical Learning, Chapter 3, Springer" and "Pattern Recognition and Machine Learning, Chapter 3, Springer".*

4.1 Modelling assumptions for regression

Object space: O, measurement/feature space: $\mathcal{F} = \mathbb{R}^p \times \mathbb{R}$
Data: $\mathcal{Z} = \{(x_i, y_i) \in \mathbb{R}^p \times \mathbb{R} : 1 \leq i \leq n\}$
Model: $\boldsymbol{y_i}$ output, $\boldsymbol{x_i} = (x_{i,0}, x_{i,1}, ..., x_{i,p})$ features with $x_{i,0} = 1 \; \forall i = 1, ..., N$, ϵ noise with $\mathbb{E}[\epsilon] = 0$

$$\boldsymbol{y_i} = \hat{f}(\boldsymbol{x_i}, \boldsymbol{\theta}) + \epsilon = \theta_0 + \sum_{j=1}^{p} \boldsymbol{x_{i,j}} \boldsymbol{\theta_j} + \epsilon$$

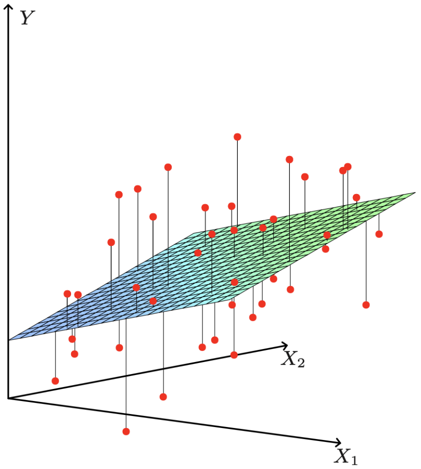
4.2 Least Squares

The most popular estimation method is least squares, in which we pick the coefficients $\boldsymbol{\theta} = (\theta_0, \theta_1, ..., \theta_p)$ to minimize the residual sum of squares:

$$\begin{aligned} \mathrm{RSS}(\boldsymbol{\theta}) &= \sum_{i=1}^{N} (\boldsymbol{y_i} - f(\boldsymbol{x_i}))^2 \\ &= \sum_{i=1}^{N} (\boldsymbol{y_i} - \theta_0 - \sum_{j=1}^{p} \boldsymbol{x_{i,j}} \theta_j)^2 \end{aligned} \tag{4.1}$$

From a statistical point of view, this criterion is reasonable if the training observations $(\boldsymbol{x_i}, \boldsymbol{y_i})$ represent independent random draws from their population. Even if the $\boldsymbol{x_i}$'s were not drawn randomly, the criterion is still valid if the $\boldsymbol{y_i}$'s are conditionally independent given the inputs $\boldsymbol{x_i}$.
Figure 4.1 illustrates the geometry of least-squares fitting in the $\mathbb{R}^{p+1}$-dimensional space occupied by the pairs $(\boldsymbol{x_i}, \boldsymbol{y_i})$.

Figure 4.1: Linear least squares fitting with $\boldsymbol{x_i} \in \mathbb{R}^2$. We seek the linear function of $\boldsymbol{X}$ that minimizes the sum of squared residuals from $\boldsymbol{Y}$.



How do we minimize Eq. 4.1? Denote by $\boldsymbol{X}$ the $N \times (p+1)$ matrix with each row an input vector (with a 1 in the first position), and similarly let $\boldsymbol{Y}$ be the N-vector of outputs in the training set. Then we can write the residual sum-of-squares as:

$$\mathrm{RSS}(\boldsymbol{\theta}) = (\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\theta})^\intercal(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\theta})$$

This is a quadratic function in the $p+1$ parameters. Differentiating with respect to $\boldsymbol{\theta}$ we obtain:

$$\frac{\partial \mathrm{RSS}}{\partial \theta} = -2\boldsymbol{X}^\intercal(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\theta})$$

$$\frac{\partial^2 \mathrm{RSS}}{\partial \theta \partial \theta^\intercal} = 2\boldsymbol{X}^\intercal\boldsymbol{X}$$

Assuming (for the moment) that $\boldsymbol{X}$ has full column rank, and hence $\boldsymbol{X}^\intercal\boldsymbol{X}$ is positive definite, we set the first derivative to zero:

$$\boldsymbol{X}^\intercal(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\theta}) = 0 \tag{4.2}$$

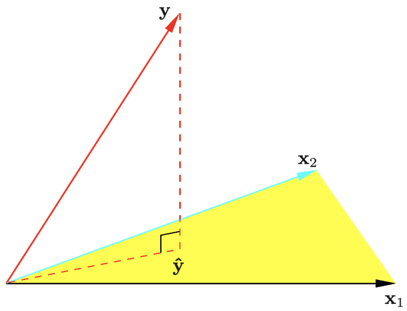
To obtain the unique solution:

$$\hat{\boldsymbol{\theta}} = (\boldsymbol{X}^\intercal\boldsymbol{X})^{-1}\boldsymbol{X}^\intercal\boldsymbol{Y}$$

The fitted values at the training inputs are:

$$\hat{\boldsymbol{Y}} = \boldsymbol{X}\hat{\boldsymbol{\theta}} = \boldsymbol{X}(\boldsymbol{X}^\intercal\boldsymbol{X})^{-1}\boldsymbol{X}^\intercal\boldsymbol{Y}$$

Figure 4.2: The N-dimensional geometry of least squares regression with two predictors. The outcome vector $\boldsymbol{Y}$ is orthogonally projected onto the hyperplane spanned by the input vectors $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$. The projection $\hat{\boldsymbol{Y}}$ represents the vector of the least squares predictions.



The matrix $\boldsymbol{H} = \boldsymbol{X}(\boldsymbol{X}^\intercal\boldsymbol{X})^{-1}\boldsymbol{X}^\intercal$ is sometimes called the "hat" matrix because it puts the hat on $\boldsymbol{Y}$. Figure 4.2 shows a different geometrical representation of the least squares estimate, this time in $\mathbb{R}^N$. We denote the column vectors of $\boldsymbol{X}$ by $x_0, x_1, ..., x_p$ with $x_0 = 1$. For much of what follows, this first column is treated like any other. These vectors span a subspace of $\mathbb{R}^N$, also referred to as the column space of $\boldsymbol{X}$. We minimize RSS $= \|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\theta}\|^2$ by choosing $\boldsymbol{\theta}$ so that the residual vector $\boldsymbol{Y} - \hat{\boldsymbol{Y}}$ is orthogonal to this subspace. This orthogonality is expressed in Eq. 4.2, and the resulting estimate $\hat{\boldsymbol{Y}}$ is hence the orthogonal projection of $\boldsymbol{Y}$ onto this subspace. The hat matrix $\boldsymbol{H}$ computes the orthogonal projection, and hence it is also known as a projection matrix.

4.2.1 The Gauss Markov Theorem

The least squares estimates of the parameters $\boldsymbol{\theta}$ have the smallest variance among all linear unbiased estimates. We will make this precise here, and also make clear that the restriction to unbiased estimates is not necessarily a wise one. This observation will lead us to consider biased estimates such as ridge regression later. We focus on estimation of any linear combination of the parameters $\beta = a^\intercal\theta$; for example, predictions $f(x_{n+1}) = x_{n+1}^\intercal\theta$ are of this form. The least squares estimate of $a^\intercal\theta$ is:

$$\hat{\beta} = a^\intercal\hat{\theta} = a^\intercal(\boldsymbol{X}^\intercal\boldsymbol{X})^{-1}\boldsymbol{X}^\intercal\boldsymbol{Y}$$

Considering $\boldsymbol{X}$ to be fixed, this is a linear function $c^\intercal\boldsymbol{Y}$ of the response vector $\boldsymbol{Y}$. If we assume that the linear model is correct, $a^\intercal\hat{\theta}$ is unbiased since:

$$\begin{aligned}
\mathbb{E}[a^\intercal\hat{\theta}] &= \mathbb{E}[a^\intercal(\boldsymbol{X}^\intercal\boldsymbol{X})^{-1}\boldsymbol{X}^\intercal\boldsymbol{Y}] \\
&= a^\intercal(\boldsymbol{X}^\intercal\boldsymbol{X})^{-1}\boldsymbol{X}^\intercal(\mathbb{E}[\boldsymbol{X}\theta + \epsilon]) \\
&= a^\intercal(\boldsymbol{X}^\intercal\boldsymbol{X})^{-1}\boldsymbol{X}^\intercal(\boldsymbol{X}\theta + \cancelto{0}{\mathbb{E}[\epsilon]}) \\
&= a^\intercal(\boldsymbol{X}^\intercal\boldsymbol{X})^{-1}\boldsymbol{X}^\intercal\boldsymbol{X}\theta \\
&= a^\intercal\theta
\end{aligned}$$

$$\begin{aligned}
\mathrm{Var}[a^\intercal\hat{\theta}] &= \mathrm{Var}[a^\intercal(\boldsymbol{X}^\intercal\boldsymbol{X})^{-1}\boldsymbol{X}^\intercal(\boldsymbol{X}\theta + \epsilon)] \\
&= \mathrm{Var}[a^\intercal(\boldsymbol{X}^\intercal\boldsymbol{X})^{-1}\boldsymbol{X}^\intercal\epsilon] \\
&= \mathbb{E}[a^\intercal(\boldsymbol{X}^\intercal\boldsymbol{X})^{-1}\boldsymbol{X}^\intercal\epsilon\epsilon^\intercal\boldsymbol{X}(\boldsymbol{X}^\intercal\boldsymbol{X})^{-1}a] \\
&= \sigma^2 a^\intercal(\boldsymbol{X}^\intercal\boldsymbol{X})^{-1}a
\end{aligned}$$

Alternative unbiased linear estimator $\widetilde{\beta} = c^\intercal \boldsymbol{Y} = a^\intercal \hat{\theta} + a^\intercal \boldsymbol{D}\boldsymbol{Y}$, where $c^\intercal = a^\intercal((\boldsymbol{X}^\intercal \boldsymbol{X})^{-1}\boldsymbol{X}^\intercal + \boldsymbol{D})$:

$$\begin{aligned}
\mathbb{E}[c^\intercal \boldsymbol{Y}] &= \mathbb{E}[a^\intercal \hat{\theta}] + \mathbb{E}[a^\intercal \boldsymbol{D}\boldsymbol{Y}] \\
&= a^\intercal \theta + \mathbb{E}[a^\intercal \boldsymbol{D}(\boldsymbol{X}\boldsymbol{\theta} + \epsilon)] \\
&= a^\intercal \theta + a^\intercal \boldsymbol{D}\boldsymbol{X}\theta + a^\intercal \boldsymbol{D}\cancelto{0}{\mathbb{E}[\epsilon]} \\
&= a^\intercal \theta
\end{aligned}$$

The unbiasedness condition $\mathbb{E}[c^\intercal \boldsymbol{Y}] = a^\intercal \theta$ implies $a^\intercal \boldsymbol{D}\boldsymbol{X} = \vec{0}$.

Theorem 4.1 *The Gauss Markov Theorem*
For any linear estimator $\widetilde{\beta} = c^\intercal \boldsymbol{Y}$ that is unbiased for $a^\intercal \theta$, that is, $\mathbb{E}[c^\intercal \boldsymbol{Y}] = a^\intercal \theta$, the following holds:

$$\mathit{Var}[a^\intercal \hat{\theta}] \leq \mathit{Var}[c^\intercal \boldsymbol{Y}]$$

Proof:

$$\begin{aligned}
\mathrm{Var}[c^\intercal \boldsymbol{Y}] &= c^\intercal \mathrm{Cov}[\boldsymbol{Y}]c = c^\intercal \sigma^2 \mathbb{1} c = \sigma^2 c^\intercal c \\
&= \sigma^2 a^\intercal((\boldsymbol{X}^\intercal \boldsymbol{X})^{-1}\boldsymbol{X}^\intercal + \boldsymbol{D})(\boldsymbol{X}(\boldsymbol{X}^\intercal \boldsymbol{X})^{-1} + \boldsymbol{D}^\intercal))a \\
&= \sigma^2 a^\intercal((\boldsymbol{X}^\intercal \boldsymbol{X})^{-1}\cancelto{\mathbb{1}}{\boldsymbol{X}^\intercal \boldsymbol{X}(\boldsymbol{X}^\intercal \boldsymbol{X})^{-1}} + \boldsymbol{D}\boldsymbol{D}^\intercal + \underbrace{(\boldsymbol{X}^\intercal \boldsymbol{X})^{-1}\boldsymbol{X}^\intercal \boldsymbol{D}^\intercal}_{\boldsymbol{X}^\intercal \boldsymbol{D}^\intercal a = \vec{0}} + \underbrace{\boldsymbol{D}\boldsymbol{X}(\boldsymbol{X}^\intercal \boldsymbol{X})^{-1}}_{a^\intercal \boldsymbol{D}\boldsymbol{X} = \vec{0}})a \\
&= \sigma^2 a^\intercal (\boldsymbol{X}^\intercal \boldsymbol{X})^{-1} a + \sigma^2 a^\intercal \boldsymbol{D}\boldsymbol{D}^\intercal a \\
&= \mathrm{Var}[a^\intercal \hat{\theta}] + \underbrace{\sigma^2 \left\|\boldsymbol{D}^\intercal a\right\|^2}_{\geq 0} \geq \mathrm{Var}[a^\intercal \hat{\theta}]
\end{aligned}$$

■

The key consequence of Gauss Markov Theorem: to beat the least squares estimate, you need bias or non-normality. Notice, however, that **over-fitting** might be a problem: we are assuming $\boldsymbol{X}$ fixed and thus, the Gauss-Markov theorem holds only for a specific instance of the training set $\boldsymbol{X}$. This can lead to high generalization error.

4.2.2 The Bias-Variance decomposition

The use of maximum likelihood, or equivalently least squares, can lead to severe over-fitting if complex models are trained using data sets of limited size. However, limiting the number of basis functions in order to avoid over-fitting has the side effect of limiting the flexibility of the model to capture interesting and important trends in the data. Although the introduction of regularization terms can control over-fitting for models with many parameters, this raises the question of how to determine a suitable value for the regularization coefficient λ. Seeking the solution that minimizes the regularized error function with respect to both the weight vector θ and the regularization coefficient λ is clearly not the right approach since this leads to the unregularized solution with $\lambda = 0$.
The phenomenon of over-fitting is really an unfortunate property of maximum likelihood and does not arise when we marginalize over parameters in a Bayesian setting.

Each loss function leads to a corresponding optimal prediction once we are given the conditional distribution $p(y|\boldsymbol{x})$. A popular choice is the squared loss function, for which the optimal prediction is given by the conditional expectation, which we denote by $h(\boldsymbol{x})$ and which is given by:

$$h(\boldsymbol{x}) = \mathbb{E}[y|\boldsymbol{x}] = \int yp(y|\boldsymbol{x})dy$$

We might use more sophisticated techniques than least squares, for example regularization or a fully Bayesian approach, to determine the conditional distribution $p(y|\boldsymbol{x})$. These can all be combined with the squared loss function for the purpose of making predictions.
Armed with the knowledge that the optimal solution is the conditional expectation, we can expand the square term as follows:

$$\begin{aligned}\{\hat{f}(\boldsymbol{x}) - y\}^2 &= \{\hat{f}(\boldsymbol{x}) - \mathbb{E}[y|\boldsymbol{x}] + \mathbb{E}[y|\boldsymbol{x}] - y\}^2 \\ &= \{\hat{f}(\boldsymbol{x}) - \mathbb{E}[y|\boldsymbol{x}]\}^2 + 2\{\hat{f}(\boldsymbol{x}) - \mathbb{E}[y|\boldsymbol{x}]\}\{\mathbb{E}[y|\boldsymbol{x}] - y\} + \{\mathbb{E}[y|\boldsymbol{x}] - y\}^2\end{aligned}$$

Substituting into the loss function and performing the integral over y, we see that the cross-term vanishes and we obtain an expression for the loss function in the form:

$$\mathbb{E}[L(\boldsymbol{X}, \boldsymbol{Y}; \hat{f})] = \int \{\hat{f}(\boldsymbol{x}) - h(\boldsymbol{x})\}^2 p(\boldsymbol{x})d\boldsymbol{x} + \underbrace{\int \{h(\boldsymbol{x}) - \boldsymbol{y}\}^2 p(\boldsymbol{x}, \boldsymbol{y})d\boldsymbol{x}d\boldsymbol{y}}_{\text{noise}} \tag{4.3}$$

The second term, which is independent of $\hat{f}(\boldsymbol{x})$, arises from the intrinsic noise on the data and represents the minimum achievable value of the expected loss. The first term depends on our choice for the function $\hat{f}(\boldsymbol{x})$, and we will seek a solution for $\hat{f}(\boldsymbol{x})$ which makes this term a minimum. Because it is non-negative, the smallest that we can hope to make this term is zero. If we had an unlimited supply of data (and unlimited computational resources), we could in principle find the regression function $h(\boldsymbol{x})$ to any desired degree of accuracy, and this would represent the optimal choice for $\hat{f}(\boldsymbol{x})$. However, in practice we have a data set containing only a finite number N of data points, and consequently we do not know the regression function $h(\boldsymbol{x})$ exactly.
If we model the $h(\boldsymbol{x})$ using a parametric function $\hat{f}(\boldsymbol{x}, \theta)$ governed by a parameter vector θ, then from a Bayesian perspective the uncertainty in our model is expressed through a posterior distribution over θ. A frequentist treatment, however, involves making a point estimate of θ based on the data set D, and tries instead to interpret the uncertainty of this estimate through the following thought experiment:
Suppose we had a large number of data sets each of size N and each drawn independently from the distribution $p(\boldsymbol{x}, \boldsymbol{y})$. For any given data set D, we can run our learning algorithm and obtain a prediction function $\hat{f}(\boldsymbol{x}; D)$. Different data sets from the ensemble will give different functions and consequently different values of the squared loss. The performance of a particular learning algorithm is then assessed by taking the average over this ensemble of data sets.

Consider the integrand of the first term in Eq. 4.3, which for a particular data set D takes the form:

$$\{\hat{f}(\boldsymbol{x}; D) - h(\boldsymbol{x})\}^2$$

Because this quantity will be dependent on the particular data set D, we take its average over the ensemble of data sets. If we add and subtract the quantity $\mathbb{E}_D[\hat{f}(\boldsymbol{x}; D)]$ inside the braces, and then expand, we obtain:

$$\begin{aligned}&\{\hat{f}(\boldsymbol{x}; D) - \mathbb{E}_D[\hat{f}(\boldsymbol{x}; D)] + \mathbb{E}_D[\hat{f}(\boldsymbol{x}; D)] - h(\boldsymbol{x})\}^2 \\ &= \{\hat{f}(\boldsymbol{x}; D) - \mathbb{E}_D[\hat{f}(\boldsymbol{x}; D)]\}^2 + \{\mathbb{E}_D[\hat{f}(\boldsymbol{x}; D)] - h(\boldsymbol{x})\}^2 \\ &+ 2\{\hat{f}(\boldsymbol{x}; D) - \mathbb{E}_D[\hat{f}(\boldsymbol{x}; D)]\}\{\mathbb{E}_D[\hat{f}(\boldsymbol{x}; D)] - h(\boldsymbol{x})\}\end{aligned}$$

We now take the expectation of this expression with respect to D and note that the final term will vanish, giving:

$$\begin{aligned}&\mathbb{E}_D[\{\hat{f}(\boldsymbol{x};D)-h(\boldsymbol{x})\}^2]\\&=\underbrace{\{\mathbb{E}_D[\hat{f}(\boldsymbol{x};D)]-h(\boldsymbol{x})\}^2}_{(\text{bias})^2}+\underbrace{\mathbb{E}_D[\{\hat{f}(\boldsymbol{x};D)-\mathbb{E}_D[\hat{f}(\boldsymbol{x};D)]\}^2]}_{\text{variance}}\end{aligned}$$

We see that the expected squared difference between $\hat{f}(\boldsymbol{x};D)$ and the regression function $h(\boldsymbol{x})$ can be expressed as the sum of two terms. The first term, called the squared bias, represents the extent to which the average prediction over all data sets differs from the desired regression function. The second term, called the variance, measures the extent to which the solutions for individual data sets vary around their average, and hence this measures the extent to which the function $\hat{f}(\boldsymbol{x};D)$ is sensitive to the particular choice of data set.
So far, we have considered a single input value $\boldsymbol{x}$. If we substitute this expansion back into Eq. 4.3, we obtain the following decomposition of the expected squared loss:

$$\textbf{expected loss} = (\text{bias})^2 + \text{variance} + \text{noise}$$

where:

$$\begin{aligned}(\text{bias})^2 &= \int \{\mathbb{E}_D[\hat{f}(\boldsymbol{x};D)]-h(\boldsymbol{x})\}^2 p(\boldsymbol{x})d\boldsymbol{x}\\ \text{variance} &= \int \mathbb{E}_D[\{\hat{f}(\boldsymbol{x};D)-\mathbb{E}_D[\hat{f}(\boldsymbol{x};D)]\}^2]p(\boldsymbol{x})d\boldsymbol{x}\\ \text{noise} &= \int \{h(\boldsymbol{x})-\boldsymbol{y}\}^2 p(\boldsymbol{x},\boldsymbol{y})d\boldsymbol{x}d\boldsymbol{y}\end{aligned}$$

and the bias and variance terms now refer to integrated quantities.

Our goal is to minimize the expected loss, which we have decomposed into the sum of a (squared) bias, a variance, and a constant noise term. As we shall see, there is a trade-off between bias and variance, with very flexible models having low bias and high variance, and relatively rigid models having high bias and low variance. The model with the optimal predictive capability is the one that leads to the best balance between bias and variance.
Although the bias-variance decomposition may provide some interesting insights into the model complexity issue from a frequentist perspective, it is of limited practical value, because the bias-variance decomposition is based on averages with respect to ensembles of data sets, whereas in practice we have only the single observed data set. If we had a large number of independent training sets of a given size, we would be better off combining them into a single large training set, which of course would reduce the level of over-fitting for a given model complexity.

4.2.3 Shrinkage Methods

4.2.3.1 Ride regression

Ridge regression shrinks the regression coefficients by imposing a penalty on their size. The ridge coefficients minimize a penalized residual sum of squares

$$\hat{\boldsymbol{\theta}}^{\text{ridge}} = \arg\min_{\theta}\{\sum_{i=1}^{N}(\boldsymbol{y_i}-\theta_0-\sum_{j=1}^{p}\boldsymbol{x_{i,j}}\theta_j)^2+\lambda\sum_{j=1}^{p}\theta_j^2\}$$

Here $\lambda \geq 0$ is a complexity parameter that controls the amount of shrinkage: the larger the value of λ, the greater the amount of shrinkage. The coefficients are shrunk toward zero (and each other).

The ridge solutions are not equivariant under scaling of the inputs, and so one normally standardizes the inputs before solving for $\boldsymbol{\theta}$.
In addition, notice that the intercept θ_0 has been left out of the penalty term. Penalization of the intercept would make the procedure depend on the origin chosen for $\boldsymbol{Y}$; that is, adding a constant c to each of the targets $\boldsymbol{y_i}$ would not simply result in a shift of the predictions by the same amount c.
Writing the RSS in matrix form we obtain

$$\mathrm{RSS}(\lambda) = (\boldsymbol{Y} - \boldsymbol{X\theta})^\intercal(\boldsymbol{Y} - \boldsymbol{X\theta}) + \lambda\boldsymbol{\theta}^\intercal\boldsymbol{\theta}$$

The ridge regression solutions are easily seen to be

$$\theta^{\text{ridge}} = (\boldsymbol{X}^\intercal\boldsymbol{X} + \lambda\mathbb{1})\boldsymbol{X}^\intercal\boldsymbol{Y} \tag{4.4}$$

Notice that with the choice of quadratic penalty $\boldsymbol{\theta}^\intercal\boldsymbol{\theta}$, the ridge regression solution is again a linear function of $\boldsymbol{Y}$. The solution adds a positive constant to the diagonal of $\boldsymbol{X}^\intercal\boldsymbol{X}$ before inversion. This makes the problem nonsingular, even if $\boldsymbol{X}^\intercal\boldsymbol{X}$ is not of full rank.

Ridge regression can also be derived as the mean or mode of a posterior distribution, with a suitably chosen prior distribution. In detail, suppose $y_i \sim \mathcal{N}(\boldsymbol{x_i\theta}, \sigma^2)$, and the parameters θ_j are each distributed as $\mathcal{N}(0, \tau^2)$, independently of one another.
Then the (negative) log-posterior density of $\boldsymbol{\theta}$, with τ^2 and σ^2 assumed known, is equal to the Eq.4.4, with $\lambda = \sigma^2/\tau^2$. Thus the ridge estimate is the mode of the posterior distribution; since the distribution is Gaussian, it is also the posterior mean.

The singular value decomposition (SVD) of the centered input matrix $\boldsymbol{X}$ gives us some additional insight into the nature of ridge regression.
The SVD of the $N \times p$ matrix $\boldsymbol{X}$ has the form

$$\boldsymbol{X} = \boldsymbol{UDV}^\intercal$$

Here $\boldsymbol{U}$ and $\boldsymbol{V}$ are $N \times p$ and $p \times p$ orthogonal matrices, with the columns of $\boldsymbol{U}$ spanning the column space of $\boldsymbol{X}$, and the columns of $\boldsymbol{V}$ spanning the row space. $\boldsymbol{D}$ is a $p \times p$ diagonal matrix, with diagonal entries $d_1 \geq d_2 \geq ... \geq d_p \geq 0$ called the singular values of $\boldsymbol{X}$. If one or more values $d_j = 0$, $\boldsymbol{X}$ is singular.
Using the singular value decomposition we can write the least squares fitted vector as

$$\boldsymbol{X}\hat{\boldsymbol{\theta}}^{\text{ls}} = \boldsymbol{X}(\boldsymbol{X}^\intercal\boldsymbol{X})^{-1}\boldsymbol{X}^\intercal\boldsymbol{Y} = \boldsymbol{UDV}^\intercal(\boldsymbol{VD}^2\boldsymbol{V}^\intercal)^{-1}\boldsymbol{VDU}^\intercal\boldsymbol{Y} = \boldsymbol{UU}^\intercal\boldsymbol{Y}$$

Note that $\boldsymbol{U}^\intercal Y$ are the coordinates of $\boldsymbol{Y}$ with respect to the orthonormal basis $\boldsymbol{U}$.
Now the ridge solutions are

$$\begin{aligned}\boldsymbol{X}\hat{\boldsymbol{\theta}}^{\text{ridge}} &= \boldsymbol{X}(\boldsymbol{X}^\intercal\boldsymbol{X} + \lambda\mathbb{1})^{-1}\boldsymbol{X}^\intercal\boldsymbol{Y} \\ &= \boldsymbol{UD}(\boldsymbol{D}^2 + \lambda\mathbb{1})^{-1}\boldsymbol{DU}^\intercal\boldsymbol{Y} \\ &= \sum_{j=1}^{p}\boldsymbol{u}_j\frac{d_j^2}{d_j^2+\lambda}\boldsymbol{u}_j^\intercal\boldsymbol{Y}\end{aligned}$$

where the $\boldsymbol{u}_j$ columns are the columns of $\boldsymbol{U}$. Note that since $\lambda \geq 0$ we have $d_j^2/(d_j^2 + \lambda) \leq 1$. Like linear regression, ridge regression computes the coordinates of $\boldsymbol{Y}$ with respect to the orthonormal basis $\boldsymbol{U}$. It then shrinks these coordinates by the factors $d_j^2/(d_j^2 + \lambda)$. This means that a greater amount of shrinkage is applied to the coordinates of basis vectors with smaller d_j^2.
The small singular values d_j correspond to directions in the column space of $\boldsymbol{X}$ having small variance, and ridge regression shrinks these directions the most.
Ridge regression protects against the potentially high variance of gradients estimated in the short directions. The implicit assumption is that the response will tend to vary most in the directions of high variance of the inputs. This is often a reasonable assumption, since predictors are often chosen for study because they vary with the response variable, but need not hold in general.

4.2.3.2 The Lasso

The lasso is a shrinkage method like ridge, with subtle but important differences. The lasso estimate is defined by

$$\hat{\boldsymbol{\theta}}^{\text{lasso}} = \arg\min_{\theta} \sum_{i=1}^{N} (\boldsymbol{y_i} - \theta_0 - \sum_{j=1}^{p} \boldsymbol{x_{i,j}}\theta_j)^2$$

$$\text{subject to } \sum_{j=1}^{p} |\theta_j| \leq t$$

We can also write the lasso problem in the equivalent Lagrangian form

$$\hat{\boldsymbol{\theta}}^{\text{lasso}} = \arg\min_{\theta} \{\sum_{i=1}^{N} (\boldsymbol{y_i} - \theta_0 - \sum_{j=1}^{p} \boldsymbol{x_{i,j}}\theta_j)^2 + \lambda \sum_{j=1}^{p} |\theta_j|\}$$

Notice the similarity to the ridge regression problem: the L_2 ridge penalty $\sum_{j=1}^{p} \theta_j^2$ is replaced by the L_1 lasso penalty $\sum_{j=1}^{p} |\theta_j|$. This latter constraint makes the solutions nonlinear in the $\boldsymbol{y_i}$, and there is no closed form expression as in ridge regression.
Because of the nature of the constraint, making t sufficiently small will cause some of the coefficients to be exactly zero. Thus the lasso does a kind of continuous subset selection. If t is chosen larger than $t_0 = \sum_{j=1}^{p} |\theta_j|$ (where $\hat{\theta}_j = \hat{\theta}_j^{\text{ls}}$, the least squares estimates), then the lasso estimates are the $\hat{\theta}_j$'s. On the other hand, for $t = t_0/2$ say, then the least squares coefficients are shrunk by about 50% on average. Like the penalty parameter in ridge regression, t should be adaptively chosen to minimize an estimate of expected prediction error.

4.2.3.3 Ridge vs. Lasso Estimation

Figure 4.3: Estimation picture for the lasso (left) and ridge regression (right). Shown are contours of the error and constraint functions. The solid blue areas are the constraint regions $|\beta_1| + |\beta_2| \leq t$ and $\beta_1^2 + \beta_2^2 \leq t^2$, respectively, while the red ellipses are the contours of the least squares error function.

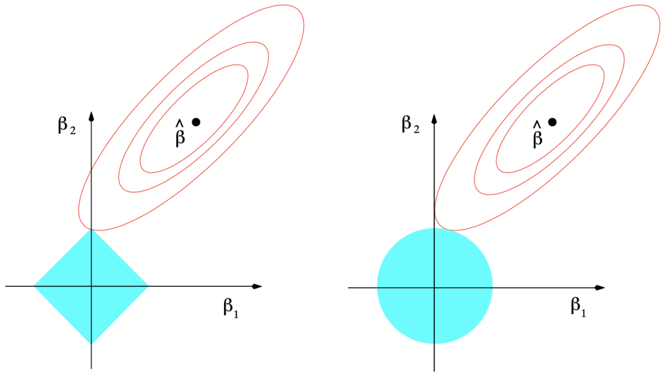
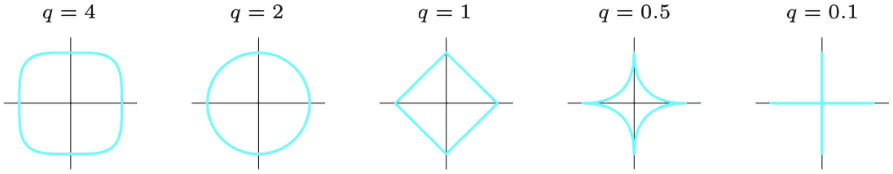


Figure 4.3 depicts the lasso (left) and ridge regression (right) when there are only two parameters. The residual sum of squares has elliptical contours, centered at the full least squares estimate. The constraint

region for ridge regression is the disk $\beta_1^2 + \beta_2^2 \leq t^2$, while that for lasso is the diamond $|\beta_1| + |\beta_2| \leq t$. Both methods find the first point where the elliptical contours hit the constraint region. Unlike the disk, the diamond has corners; if the solution occurs at a corner, then it has one parameter β_j equal to zero. When $p > 2$, the diamond becomes a rhomboid, and has many corners, flat edges and faces; there are many more opportunities for the estimated parameters to be zero. We can generalize ridge regression and the lasso, and view them as Bayes estimates. Consider the criterion for $q \geq 0$.

$$\tilde{\boldsymbol{\beta}} = \arg\min_{\beta}\{\sum_{i=1}^{N}(\boldsymbol{y_i} - \beta_0 - \sum_{j=1}^{p} \boldsymbol{x_{i,j}}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j|^q\}$$

Figure 4.4: Contours of constant value of $\sum\limits_{j=1}^{p} |\beta_j|^q$ for given values of q.



Thinking of $|\beta_j|^q$ as the log-prior density for β_j, these are also the equicontours of the prior distribution of the parameters. The value $q = 1$ corresponds to the lasso, while $q = 2$ to ridge regression. Notice that for $q \leq 1$, the prior is not uniform in direction, but concentrates more mass in the coordinate directions. The prior corresponding to the $q = 1$ case is an independent double exponential (or Laplace) distribution for each input, with density $(1/2\tau)\exp(-|\boldsymbol{\beta}/\tau)$ and $\tau = 1/\lambda$. The case $q = 1$ (lasso) is the smallest q such that the constraint region is convex; non-convex constraint regions make the optimization problem more difficult.
In this view, the lasso and ridge regression are Bayes estimates with different priors. Note, however, that they are derived as posterior modes, that is, maximizers of the posterior. It is more common to use the mean of the posterior as the Bayes estimate. Ridge regression is also the posterior mean, but the lasso is not.

4.3 MLE Regression with Gaussian Noise

Assumption: noise comes from a zero-mean Gaussian distribution with variance σ^2, i.e $\epsilon \sim \mathcal{N}(0, \sigma^2)$.
Adding a constant to a Gaussian just has the effect of shifting its mean, so the resulting conditional probability distribution for our generative probabilistic process is:

$$\mathbb{P}(\boldsymbol{y_i}|\, \boldsymbol{x_i}, \boldsymbol{\theta}, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{-\frac{1}{2\sigma^2}(\boldsymbol{y_i} - \boldsymbol{x_i}^\intercal\boldsymbol{\theta})\right\}$$

We denote the noise associated with the ith observation as ϵ_i and we will take these to be independent and identically distributed. This allows us to write the overall likelihood function as a product over these N terms:

$$\mathbb{P}(\{\boldsymbol{y_i}\}_{i=1}^{N}|\, \{\boldsymbol{x_i}\}_{i=1}^{N}, \boldsymbol{\theta}, \sigma^2) = \prod_{i=1}^{N} \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{-\frac{1}{2\sigma^2}(\boldsymbol{y_i} - \boldsymbol{x_i}^\intercal\boldsymbol{\theta})\right\}$$

Now we are going to turn this univariate Gaussian distribution into a multivariate Gaussian distribution with a diagonal covariance matrix:

$$\mathbb{P}(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{X\theta}, \sigma^2\mathbb{1}) = (2\sigma^2\pi)^{-N/2} \exp\left\{-\frac{1}{2\sigma^2}(\boldsymbol{Y} - \boldsymbol{X\theta})^\intercal(\boldsymbol{Y} - \boldsymbol{X\theta})\right\}$$

We can now think about how we should maximize this with respect to $\boldsymbol{\theta}$ in order to find the maximum likelihood estimate. Thus, it is helpful to take the natural log first:

$$\log \mathbb{P}(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{\theta}) = -\frac{N}{2}\log(2\sigma^2\pi) - \frac{1}{2\sigma^2}(\boldsymbol{Y} - \boldsymbol{X\theta})^\intercal(\boldsymbol{Y} - \boldsymbol{X\theta})$$

The additive term does not have a $\boldsymbol{\theta}$. We are then left with the following optimization problem:

$$\boldsymbol{\theta}^{\text{MLE}} = \arg\max_{\theta}\left\{-\frac{1}{2\sigma^2}(\boldsymbol{Y} - \boldsymbol{X\theta})^\intercal(\boldsymbol{Y} - \boldsymbol{X\theta})\right\} \tag{4.5}$$

The $\frac{1}{2\sigma^2}$ does not change the solution to this problem and of course we could change the sign and make this maximization into a minimization:

$$\boldsymbol{\theta}^{\text{MLE}} = \arg\min_{\theta}(\boldsymbol{Y} - \boldsymbol{X\theta})^\intercal(\boldsymbol{Y} - \boldsymbol{X\theta})$$

This is exactly the same optimization problem for the least-squares linear regression! While it seems like the loss function view and the maximum likelihood view are different, this reveals that they are often the same under the hood: least squares can be interpreted as assuming Gaussian noise, and particular choices of likelihood can be interpreted directly as (usually exponentiated) loss functions.

4.3.1 Fitting σ^2

One thing that is different about maximum likelihood, however, is that it gives us an additional parameter to play with that helps us reason about the *predictive distribution.* The predictive distribution is the distribution over the label, given parameters we have just fit. Rather than simply producing a single estimate, when we have a probabilistic model we can account for noise when we look at test data.
That is, after finding $\boldsymbol{\theta}^{\text{MLE}}$ if we have a query input $\boldsymbol{x}_{\text{pred}}$ for which we do not know the $\boldsymbol{y}$, we could compute a guess via $y_{\text{pred}} = \boldsymbol{x}_{\text{pred}}\boldsymbol{\theta}^{\text{MLE}}$, or we could actually construct a whole distribution:

$$\mathbb{P}(y_{\text{pred}}|\, \boldsymbol{x}_{\text{pred}}, \boldsymbol{\theta}^{\text{MLE}}, \sigma^2) = \mathcal{N}(\boldsymbol{x}_{\text{pred}}{}^\intercal\boldsymbol{\theta}^{\text{MLE}}, \sigma^2)$$

We can start by assuming that we have already computed $\boldsymbol{\theta}^{\text{MLE}}$, in order to calculate σ^2 we set up the problem the same way except we keep the additive term in Eq. 4.5:

$$\sigma^{\text{MLE}} = \arg\max_{\sigma}\left\{-\frac{N}{2}\log(2\sigma^2\pi) - \frac{1}{2\sigma^2}(\boldsymbol{X\theta} - \boldsymbol{Y})^\intercal(\boldsymbol{X\theta} - \boldsymbol{Y})\right\}$$

Solving this maximization problem is again just a question of differentiating and setting to zero:

$$\begin{aligned}
\frac{\partial\left[-\frac{N}{2}\log(\sigma^2) - \frac{1}{2\sigma^2}(\boldsymbol{X\theta} - \boldsymbol{Y})^\intercal(\boldsymbol{X\theta} - \boldsymbol{Y})\right]}{\partial\sigma^2} &= 0 \\
-\frac{N}{2\sigma^2} + \frac{1}{2\sigma^4}(\boldsymbol{X\theta} - \boldsymbol{Y})^\intercal(\boldsymbol{X\theta} - \boldsymbol{Y}) &= 0 \\
-N + \frac{1}{\sigma^2}(\boldsymbol{X\theta} - \boldsymbol{Y})^\intercal(\boldsymbol{X\theta} - \boldsymbol{Y}) &= 0 \\
\sigma^2 = \frac{1}{N}(\boldsymbol{X\theta} - \boldsymbol{Y})^\intercal(\boldsymbol{X\theta} - \boldsymbol{Y})&
\end{aligned}$$

This is a satisfying result because it is just finding the sample average of the squared deviations between what $\boldsymbol{\theta}^{\text{MLE}}$ predicts and what the training data actually are.

4.4 Bayesian Linear Regression

Bayesian treatment of linear regression avoids over-fitting and also leads to automatic methods of determining model complexity using the training data alone. It makes predictions using all possible parameters, weighted by their posterior probability.
Since the likelihood function $\mathbb{P}(\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{X\theta},\sigma^2\mathbb{1})$ is the exponential of a quadratic function, the corresponding conjugate prior is defined by $\mathbb{P}(\boldsymbol{\theta}) = \mathcal{N}(0,\Lambda^{-1})$.
(N.B Assuming Λ and σ^2 as fixed is a big assumption).

Now we compute the posterior distribution, which is proportional to the product of the likelihood function and the prior.

$$\begin{aligned}\mathbb{P}(\boldsymbol{\theta}|\boldsymbol{X},\boldsymbol{Y}) &\propto \mathbb{P}(\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{\theta})\mathbb{P}(\boldsymbol{\theta})\\ \log\mathbb{P}(\boldsymbol{\theta}|\boldsymbol{X},\boldsymbol{Y}) &\propto -\frac{1}{2}\sigma^2(\boldsymbol{Y}-\boldsymbol{X\theta})^\intercal(\boldsymbol{Y}-\boldsymbol{X\theta}) - \frac{1}{2}\boldsymbol{\theta}^\intercal\Lambda\boldsymbol{\theta} + const\\ &\propto -\frac{1}{2}\boldsymbol{\theta}^\intercal(\sigma^2\boldsymbol{X}^\intercal\boldsymbol{X}+\Lambda)\boldsymbol{\theta} + \boldsymbol{\theta}^\intercal(\sigma^2\boldsymbol{X}^\intercal\boldsymbol{Y}) + const\end{aligned}$$

By completing the square we note that:

$$\Sigma_\theta = \sigma^2(\boldsymbol{X}^\intercal\boldsymbol{X}+\sigma^2\Lambda)^{-1}$$

$$\mu_\theta = \sigma^{-2}\Sigma_\theta\boldsymbol{X}^\intercal\boldsymbol{Y} = (\boldsymbol{X}^\intercal\boldsymbol{X}+\sigma^2\Lambda)^{-1}\boldsymbol{X}^\intercal\boldsymbol{Y}$$

If we consider an infinitely broad prior $\Lambda = \alpha\mathbb{1}$ with $\alpha \to 0$, the mean μ_θ of the posterior distribution reduces to the maximum likelihood value.
Similarly if $N = 0$, then the posterior distribution refers to the prior.
Furthermore, if data points arrive sequentially, then the posterior distribution at any stage acts as the prior distribution for the subsequent data point.
If we consider a zero-mean isotropic Gaussian governed by a precision parameter α so that:

$$\mathbb{P}(\boldsymbol{\theta}|\,\alpha) = \mathcal{N}(\boldsymbol{\theta}|\,0,\alpha^{-1}\mathbb{1})$$

The corresponding posterior distribution over $\boldsymbol{\theta}$ has:

$$\mu_\theta = \sigma^{-2}(\alpha\mathbb{1}+\sigma^{-2}\boldsymbol{X}^\intercal\boldsymbol{X})^{-1}\boldsymbol{X}^\intercal\boldsymbol{Y} = (\alpha\sigma^2\mathbb{1}+\boldsymbol{X}^\intercal\boldsymbol{X})\boldsymbol{X}^\intercal\boldsymbol{Y}$$

$$\Sigma_\theta^{-1} = \alpha\mathbb{1}+\sigma^{-2}\boldsymbol{X}^\intercal\boldsymbol{X}$$

The log of the posterior distribution is given by the sum of the log-likelihood and the log of the prior, and as a function of $\boldsymbol{\theta}$, takes the form:

$$\log\mathbb{P}(\boldsymbol{\theta}|\,\alpha) = -\frac{\sigma^{-2}}{2}\sum_{i=1}^{N}(\boldsymbol{y_i}-\boldsymbol{\theta}^\intercal\boldsymbol{x_i})^2 - \frac{\alpha}{2}\boldsymbol{\theta}^\intercal\boldsymbol{\theta} + const$$

Maximization of this posterior distribution with respect to $\boldsymbol{\theta}$ is therefore equivalent to the minimization of the sum-of-squares error function with the addition of a quadratic regularization term ($\lambda = \alpha\sigma^2$).