Statistical Learning: Regression

Dr Michael Whitehouse

Imperial | MLGH | AIMS

Regression

Introduction

Suppose we have some data of the form

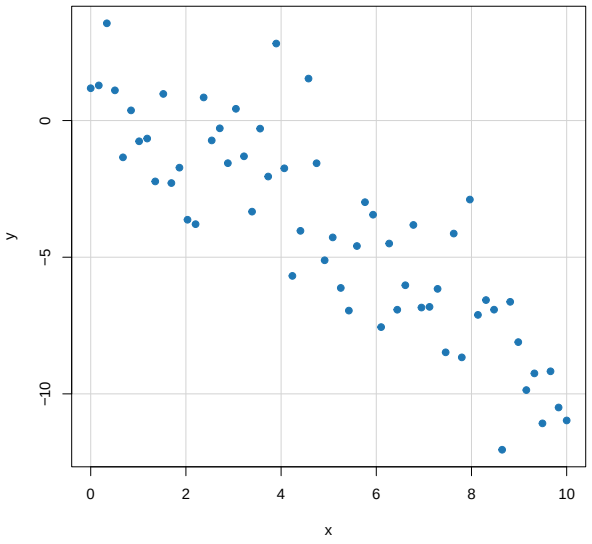
$$(x_1, y_1), \ldots, (x_N, y_N)$$

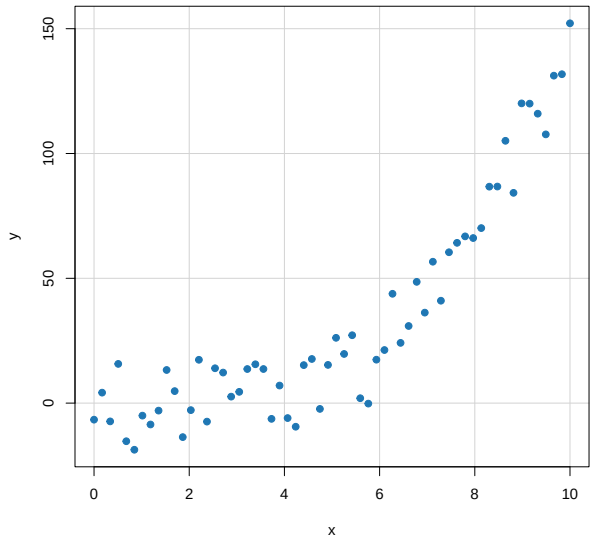
where, for $i = 1, \ldots, N$:

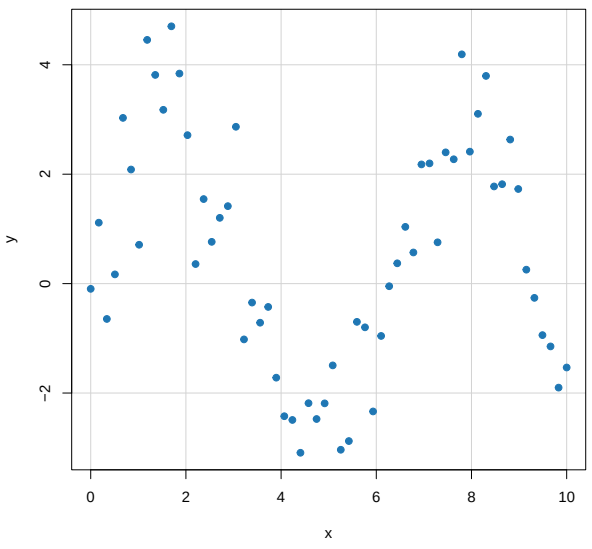
- $x_i = (x_{i1}, \ldots, x_{ip})^\top \in \mathbb{R}^p$. These are the **covariates**, also known as **predictors**, **regressors**, or **explanatory variables**, which we think of as the vectors of inputs.
- $y_i \in \mathbb{R}$. These are the **response variables**, the real-valued outputs we would like to predict.

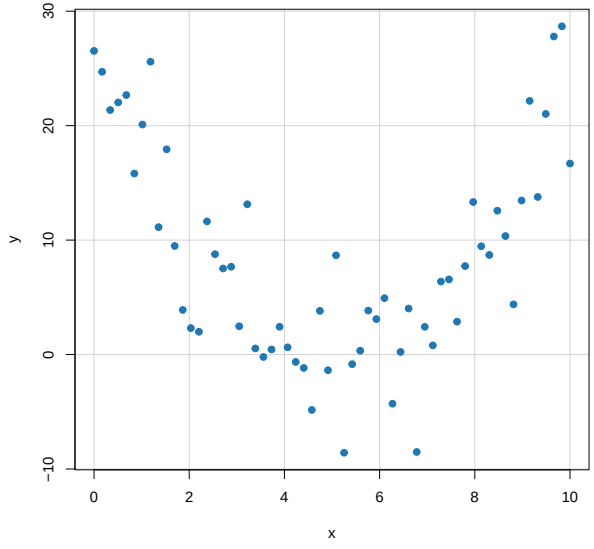
We will view the data points (x_i, y_i) as independent, identically distributed (i.i.d.) realisations of some random variables (X, Y), which take values in $\mathbb{R}^p \times \mathbb{R}$.

Introduction









Introduction

The goal of **regression** is to find a function $f : \mathbb{R}^p \to \mathbb{R}$ to model the relationship

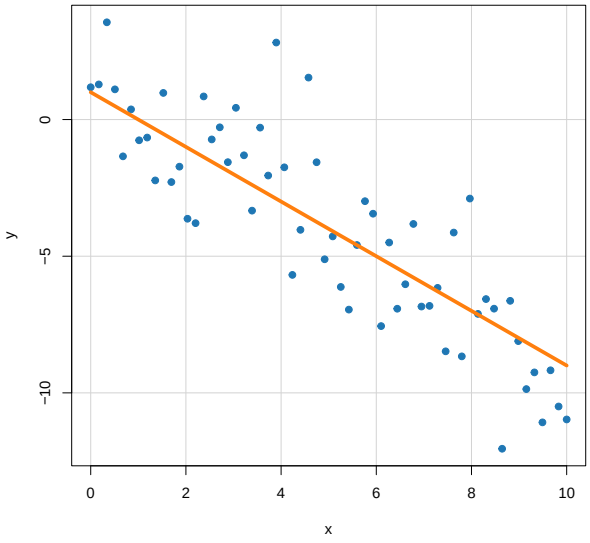
$$Y = f(X) + \varepsilon,$$

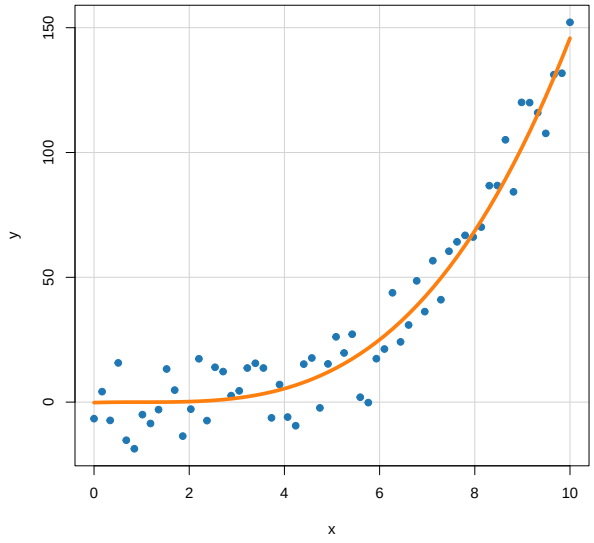
for some noise variable $\varepsilon = (\varepsilon_1, \ldots, \varepsilon_N)^\top$. We will assume that the noise variable ε:

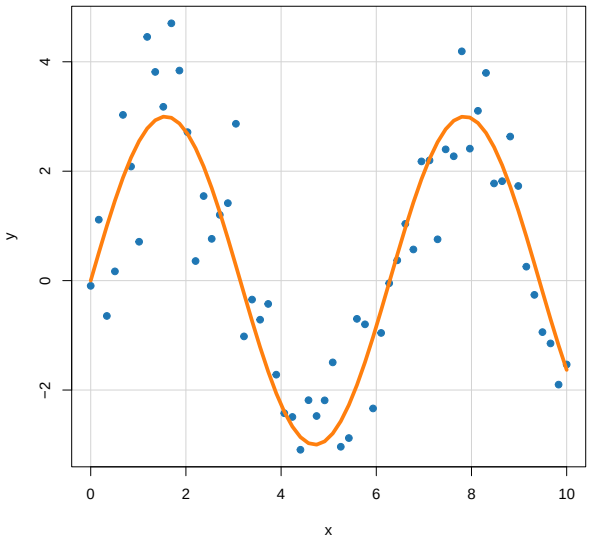
- is independent of X;
- satisfied $\mathbb{E}[\varepsilon] = 0$.

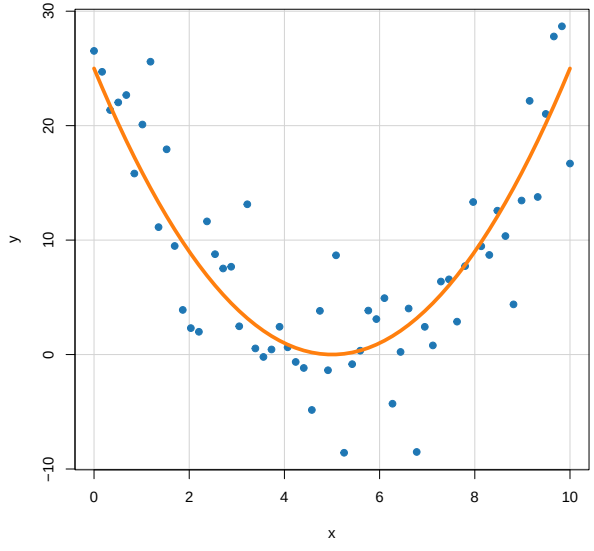
We then have $\mathbb{E}[Y|X = x] = f(x)$. In other words, the function f models the **expectation of Y given the covariates X**.

Introduction









Linear Regression

The Model

We'll begin with the simplest case where we assume that f is a linear function of the covariates. This is known as **linear regression**.

To be precise, writing $X = (X_1, \ldots, X_p)$, we will assume

$$f(X) = \sum_{j=1}^{p} X_j \beta_j \tag{1}$$

where $\beta_1, \ldots, \beta_p \in \mathbb{R}$ are a set of (unknown) real parameters.

The Model

This means that each realisation y_i of the response obeys a linear relationship with the covariates $x_{i1}, \ldots, x_{ip}$ of the form

$$y_i = x_i^\top \beta + \varepsilon_i = \sum_{j=1}^{p} x_{ij}\beta_j + \varepsilon_i,$$

To include an intercept in our model, we can include an additional 'dummy' covariate, $x_{i,0} = 1$, for all $i = 1, \ldots, N$. The previous equation then becomes

$$y_i = x_i^\top \beta + \varepsilon_i = \beta_0 + \sum_{j=1}^{p} x_{ij}\beta_j + \varepsilon_i,$$

The Model

Using vector-matrix notation, we can rewrite this equation compactly as

$$\mathbf{y} = \mathbf{X}\beta + \varepsilon,$$

where

- $\mathbf{y} = (y_1, \ldots, y_N)^\top \in \mathbb{R}^N$ is the vector of responses;
- $\beta = (\beta_1, \ldots, \beta_p)^\top \in \mathbb{R}^p$ is the vector of parameters;
- $\varepsilon = (\varepsilon_1, \ldots, \varepsilon_N)^\top \in \mathbb{R}^N$ is the vector of zero mean i.i.d. noise variables;
- $\mathbf{X}$ is the $N \times p$ matrix of covariates (the **design matrix**)

$$\mathbf{X} = \begin{pmatrix} x_1^\top \\ x_2^\top \\ \vdots \\ x_N^\top \end{pmatrix} = \begin{pmatrix} x_{11} & \cdots & x_{1p} \\ x_{21} & \cdots & x_{2p} \\ \vdots & \vdots & \vdots \\ x_{N1} & \cdots & x_{Np} \end{pmatrix},$$

whose i^{th} row $x_i^\top$ contains the i^{th} observations of all the covariates.

The Model (with Intercept)

Using vector-matrix notation, we can rewrite this equation compactly as

$$\mathbf{y} = \mathbf{X}\beta + \varepsilon,$$

where

- $\mathbf{y} = (y_1, \ldots, y_N)^\top \in \mathbb{R}^N$ is the vector of responses;
- $\beta = (\beta_0, \beta_1, \ldots, \beta_p)^\top \in \mathbb{R}^{p+1}$ is the vector of parameters;
- $\varepsilon = (\varepsilon_1, \ldots, \varepsilon_N)^\top \in \mathbb{R}^N$ is the vector of zero mean i.i.d. noise variables;
- $\mathbf{X}$ is the $N \times (p+1)$ matrix of covariates (the **design matrix**)

$$\mathbf{X} = \begin{pmatrix} x_1^\top \\ x_2^\top \\ \vdots \\ x_N^\top \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & \ldots & x_{1p} \\ 1 & x_{21} & \ldots & x_{2p} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_{N1} & \ldots & x_{Np} \end{pmatrix},$$

whose i^{th} row $x_i^\top$ contains the i^{th} observations of all the covariates.

Estimation

Our statistical problem can be stated as follows:

> *Given the training data* $(x_1, y_1), \ldots, (x_N, y_N)$*, can we infer (or 'learn') the model parameters* β*?*

Least Squares Estimation

We can estimate these parameters by minimising an appropriate **objective function** or **loss function**.

A natural choice for the objective function is the **least squares** objective. In this case, we choose the parameters $\beta = (\beta_1, \ldots, \beta_p)^\top$ which minimise the residual sum of squares (RSS):

$$\begin{aligned} \mathrm{RSS}(\beta) &:= \sum_{i=1}^{N} (y_i - f(x_i))^2 \\ &= \sum_{i=1}^{N} \left(y_i - x_i^\top \beta\right)^2 \\ &= \|\mathbf{y} - \mathbf{X}\beta\|_2^2. \end{aligned} \tag{2}$$

We will assume for now that the matrix $\mathbf{X}$ has full column rank (the columns of $\mathbf{X}$ are linearly independent). This implies that $p \leq N$.

Least Squares Estimation

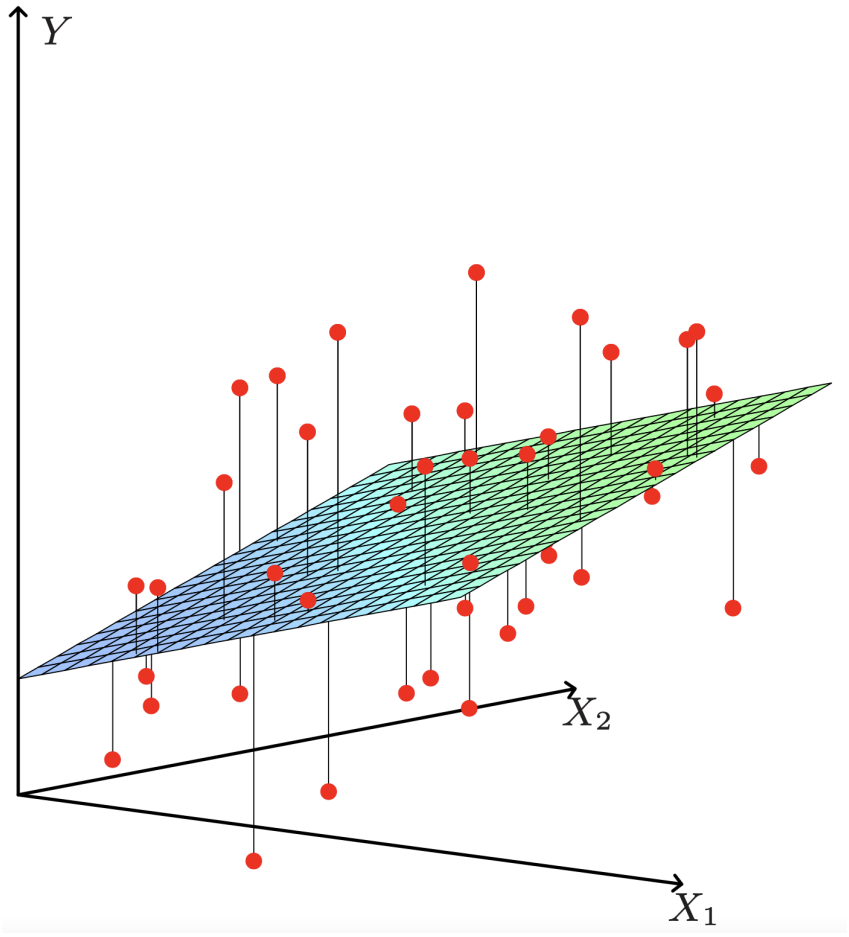


Figure: **Fitting the linear least squares model for $X = (X_1, X_2) \in \mathbb{R}^2$.** We look for the linear function of X (in this case, a plane) which minimises the sum of squared residuals from the observations Y (red).

The OLS estimator

Proposition 1

Assume $\mathbf{X}$ *has full column rank. The unique minimizer of the RSS objective is given by the the* **ordinary least squares** *(OLS) estimator,*

$$\hat{\beta}^{\mathrm{OLS}} = \underset{\beta \in \mathbb{R}^p}{\arg\min}\, \mathrm{RSS}(\beta) = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}.$$

Proof.

We can directly differentiate (2) and set equal to 0 to find the expression for $\hat{\beta}^{\mathrm{OLS}}$. The fact that this is a minimizer is seen by checking the second derivatives. □

Properties of the OLS Estimator

Under the additional assumption that that the errors are normally distributed, that is, $\varepsilon_i \sim N(0, \sigma^2)$, we then have

$$\hat{\beta}^{\text{OLS}} \sim N\left(\beta, (\mathbf{X}^\top \mathbf{X})^{-1}\sigma^2\right).$$

In this case, it also turns out that the OLS estimate $\hat{\beta}^{\text{OLS}}$ coincides with the **maximum likelihood estimate** $\hat{\beta}^{\text{MLE}}$, defined as the maximiser of the likelihood function

$$\begin{aligned}
\hat{\beta}^{\text{MLE}} &= \arg\max_{\beta} \mathcal{L}(\beta, \sigma^2) \\
&= \arg\max_{\beta} \prod_{i=1}^{N} \mathcal{N}(y_i | x_i, \beta, \sigma^2) \\
&= \arg\max_{\beta} \frac{1}{(2\pi\sigma^2)^{\frac{N}{2}}} \exp\left[-\frac{1}{2\sigma^2}\sum_{i=1}^{N}(y_i - x_i^\top \beta)^2\right].
\end{aligned}$$

Bayesian Linear Regression

Bayesian Linear Regression

Suppose we want to perform a Bayesian analysis of the Linear regression model. Following the Bayesian model set up we choose priors and set a likelihood:

$$\beta \sim p(\beta)$$
$$y_i \mid \beta \sim p(y_i \mid \beta, x_i),$$

our interest lies in deriving the posterior distribution:

$$p(\beta \mid y_i, x_i) \propto p(y_i \mid \beta, x_i) p(\beta)$$

Bayesian Linear Regression: Setup

Let $y_1, \ldots, y_n \sim i.i.d.\ \mathcal{N}(\mathbf{x}_i^\top \beta, \sigma^2)$. We can re-write this in matrix notation as follows:

$$\mathbf{y} \sim \text{MVN}(\mathbf{X}\beta, \sigma^2 \boldsymbol{I}_n)$$

where $\mathbf{y} = (Y_1, \ldots, Y_n)^\top$ is an n-dimensional vector of outcomes, $\mathbf{X}$ is a $n \times p$ design matrix, and $\boldsymbol{I}_n$ is an $n \times n$ identity matrix. The expression for the likelihood is

$$p(\mathbf{y} = \mathbf{y} | \beta, \sigma^2) = (2\pi\sigma^2)^{-n/2} \exp\left\{ -\frac{1}{2\sigma^2} (\mathbf{y} - \mathbf{X}\beta)^\top (\mathbf{y} - \mathbf{X}\beta) \right\}.$$

Recall a prior $\beta \sim p(\beta)$ is *conjugate* to the likelihood $p(y \mid \beta, x, \sigma)$ if both have the same functional form with respect to β.

Prior on the coefficients

We have two unknown quantities to infer: β and σ^2. We focus on β for now and assume σ^2 is known.
Let β_0 and $\mathbf{\Sigma}_0$ be the prior mean and covariance:

$$
\begin{aligned}
p(\beta) &= (2\pi)^{-p/2}|\mathbf{\Sigma}_0|^{-1/2}\exp\left\{-\frac{1}{2}(\beta-\beta_0)^\top\mathbf{\Sigma}_0^{-1}(\beta-\beta_0)\right\} \\
&\propto \exp\left\{-\frac{1}{2}(\beta-\beta_0)^\top\mathbf{\Sigma}_0^{-1}(\beta-\beta_0)\right\} \\
&\propto \exp\left\{-\frac{1}{2}\beta^\top\mathbf{\Sigma}_0^{-1}\beta-\beta^\top\mathbf{\Sigma}_0^{-1}\beta_0\right\}
\end{aligned}
$$

Derivation of the full conditional

We apply Bayes' theorem to derive the full conditional

$$
\begin{aligned}
p(\beta|\mathbf{y}, \mathbf{X}, \sigma^2) &\propto p(\mathbf{y}|\beta, \sigma^2)p(\beta) \\
&\propto \exp\left\{-\frac{1}{2\sigma^2}(\mathbf{y} - \mathbf{X}\beta)^\top(\mathbf{y} - \mathbf{X}\beta\right\} \exp\left\{-\frac{1}{2}(\beta - \beta_0)^\top \mathbf{\Sigma}_0^{-1}(\beta - \beta_0)\right\} \\
&\propto \exp\left\{-\frac{1}{2}\beta^\top(\mathbf{X}^\top\mathbf{X}/\sigma^2 + \mathbf{\Sigma}_0^{-1})\beta - \beta^\top(\mathbf{\Sigma}_0^{-1}\beta_0 + \mathbf{X}^\top\mathbf{y}/\sigma^2)\right\}
\end{aligned}
$$

The expression above tells us that $p(\beta|\mathbf{y}, \mathbf{X}, \sigma^2)$ is MVN with

$$
\begin{aligned}
\text{Var}[\beta|\mathbf{y}, \mathbf{X}, \sigma^2] &= (\mathbf{\Sigma}_0^{-1} + \mathbf{X}^\top\mathbf{X}/\sigma^2)^{-1} \\
\text{E}[\beta|\mathbf{y}, \mathbf{X}, \sigma^2] &= (\mathbf{\Sigma}_0^{-1} + \mathbf{X}^\top\mathbf{X}/\sigma^2)^{-1}(\mathbf{\Sigma}_0^{-1}\beta_0 + \mathbf{X}^\top\mathbf{y}/\sigma^2)
\end{aligned}
$$

Relationship to Ordinary Least Squares

Suppose we set $\beta_0 = \mathbf{0}$ and $\mathbf{\Sigma}_0 = \tau_0^2 \boldsymbol{I}_p$ in our prior for β where $\boldsymbol{I}_p$ is a $p \times p$ identity matrix. Substituting this into the above:

$$\begin{aligned} \text{Var}[\beta|\mathbf{y}, \mathbf{X}, \sigma^2] &= (\mathbf{I}_p/\tau_0^2 + \mathbf{X}^\top \mathbf{X}/\sigma^2)^{-1} \\ \text{E}[\beta|\mathbf{y}, \mathbf{X}, \sigma^2] &= (\mathbf{I}_p/\tau_0^2 + \mathbf{X}^\top \mathbf{X}/\sigma^2)^{-1}(\mathbf{X}^\top \mathbf{y}/\sigma^2). \end{aligned}$$

Relationship to Ordinary Least Squares

If we we have little prior information about the values of β, then a way to express this is to have τ_0^2 be a really large number, e.g. $\to \infty$. Then we have:

$$\begin{aligned}\lim_{\tau_0^2 \to \infty} \mathrm{E}[\beta | \mathbf{y}, \mathbf{X}, \sigma^2] &= \lim_{\tau_0^2 \to \infty} (\mathbf{I}_p / \tau_0^2 + \mathbf{X}^\top \mathbf{X} / \sigma^2)^{-1} (\mathbf{X}^\top \mathbf{y} / \sigma^2) \\ &= (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y} = \beta^{\mathrm{OLS}}\end{aligned}$$

MCMC for the full posterior

If we want to also make inference on σ^2 we can assume a prior $p(\sigma^2)$, derive the conditional posterior $p(\sigma^2|\mathbf{y}, \mathbf{X}, \beta)$.
If we assume $p(\sigma^2)$ is an *inverse-gamma* distribution, it turns out we can compute $p(\sigma^2|\mathbf{y}, \mathbf{X}, \beta)$ analytically - in this case we can use Gibbs sampling (recall from previous lecture!).

1. Update β:
    1. Generate $\beta^{(s+1)} \sim p(\beta|\mathbf{y}, \mathbf{X}, \sigma^{(s)})$
2. Update σ^2:
    1. Generate $\sigma^{(s+1)} \sim p(\sigma^2|\mathbf{y}, \mathbf{X}, \beta^{(s+1)})$

Arbitrary priors and likelihoods

If we can't even analytically compute conditional likelihoods, not all is lost. We can use probabilistic programming languages, such as Stan, to sample from the posterior - (much) more on this later in the course!

Going Beyond Linearity

Introduction

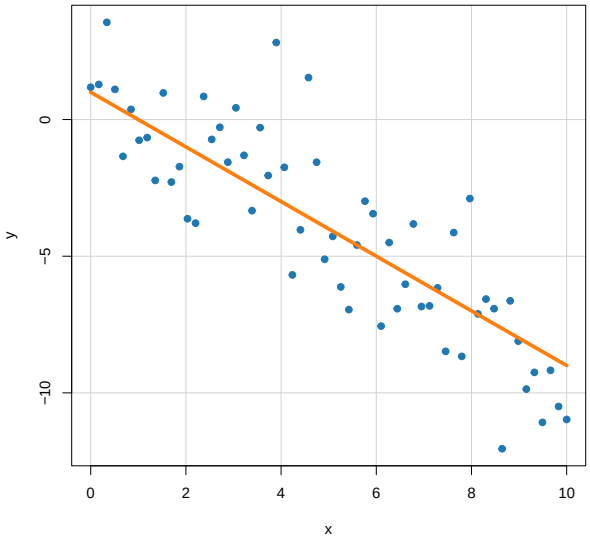
So we have assumed that the true regression function is a **linear function** of x:

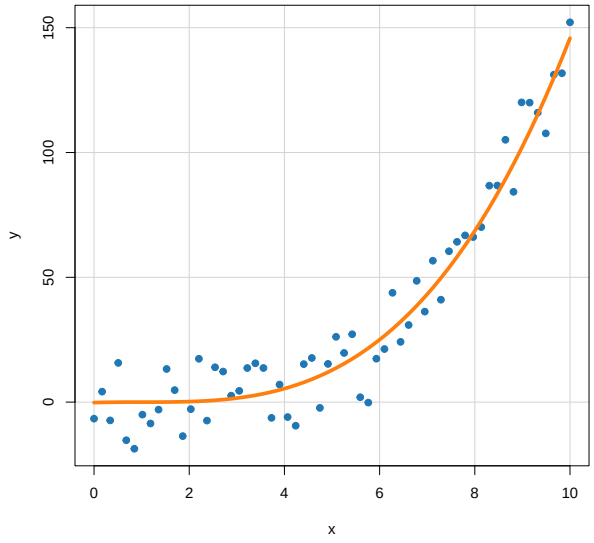
$$f(x) = \mathbb{E}[Y|X = x]. \tag{3}$$

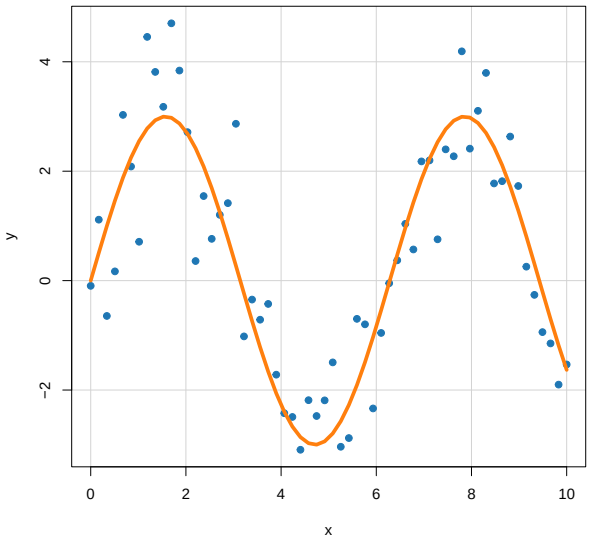
In practice, however, this is often **unrealistic**. Indeed, $f(x)$ will often be nonlinear and nonadditive in x.

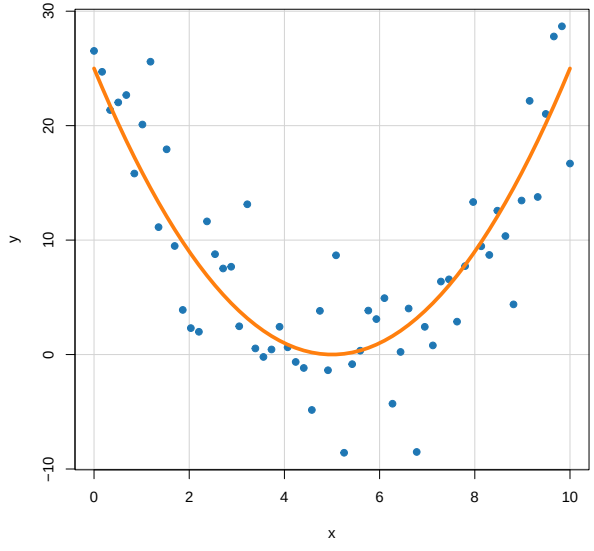
We'll now introduce several methods for **going beyond linearity**.

Introduction









Basis Functions

The basic idea is to transform or augment the inputs X with additional variables which are transformations of X.

We can then use linear models on this new space of derived input features.

Let $h_m(X) : \mathbb{R}^p \to \mathbb{R}$, denote a fixed sequence of transformations, for $m = 1, \ldots, M$. Then, for $X \in \mathbb{R}^p$, instead of a linear model we will use

$$f(X) = \sum_{m=1}^{M} \beta_m h_m(X). \tag{4}$$

This represents a **linear basis expansion** in X. The beauty of this approach is that, once we have chosen the basis functions h_m, the model is linear in these variables.

We can thus all of our existing inference tools for linear models.

Basis Functions

In particular, suppose that we observe $(x_1, y_1), \ldots, (x_N, y_N)$, where each $x_i = (x_{i1}, \ldots, x_{ip})^T \in \mathbb{R}^p$ and $y_i \in \mathbb{R}$. Suppose also we specify some basis functions $h_1, \ldots, h_M : \mathbb{R}^p \to \mathbb{R}$.

In vector-matrix notation, we can now write our model as

$$\mathbf{y} = \mathbf{N}\beta + \varepsilon \tag{5}$$

where now, instead of our original design matrix $\mathbf{X}$, we now have the matrix

$$\mathbf{N} = \begin{pmatrix} h_1(x_1) & \ldots & h_M(x_1) \\ \vdots & \vdots & \vdots \\ h_1(x_N) & \ldots & h_M(x_N) \end{pmatrix} \tag{6}$$

and all the other terms ($\mathbf{y}$, β, and ε) are defined as before.

Basis Functions

Similar to before, to estimate $\beta \in \mathbb{R}^p$, we can try to minimise the residual sum-of-squares:

$$\mathrm{RSS}(\beta) = ||\mathbf{y} - \mathbf{N}\beta||^2. \tag{7}$$

Following the same steps as before, we find that the OLS estimator is now just given by

$$\hat{\beta}^{\mathrm{OLS}} = (\mathbf{N}^\top \mathbf{N})^{-1} \mathbf{N}^\top \mathbf{y} \tag{8}$$

Similarly, our ridge regression estimator is now just given by

$$\hat{\beta}^{\mathrm{r}}_\lambda = (\mathbf{N}^\top \mathbf{N} + \lambda \mathbf{I})^{-1} \mathbf{N}^\top \mathbf{y} \tag{9}$$

Basis Functions

Some simple and widely used examples of the transformations h_m are the following.

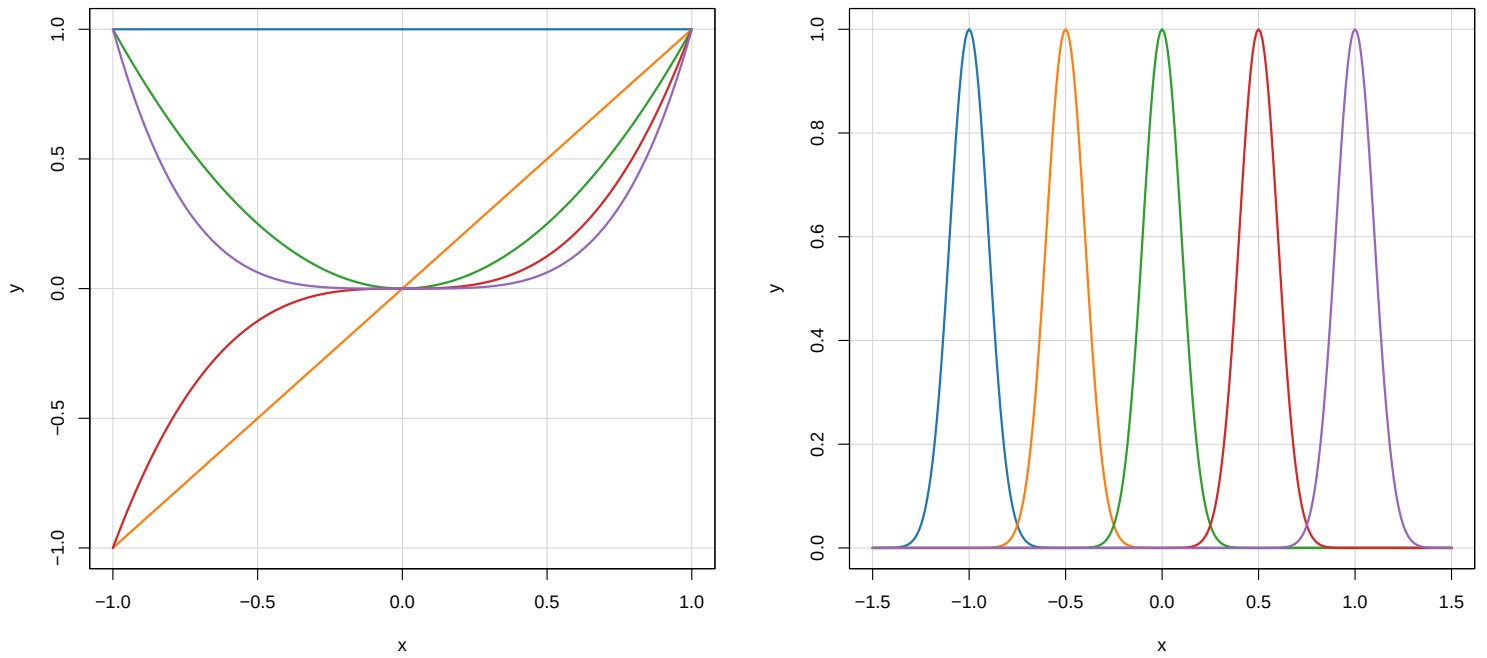
(i) $h_m(X) = X_m$, for $m = 1, \ldots, p$. This recovers the original linear model.

(ii) $h_m(X) = X_j^2$ or $X_j X_k$, for some $j, k \in 1, \ldots, p$. This allows us to model second-order effects.

(iii) $h_m(X) = \log(X_j), \sqrt{X_j}, \ldots$. This permits nonlinear transformations of single inputs. More generally, we could use non-linear transformations of multiple inputs (or all of the inputs).

(iv) $h_m(X) = 1(L_m \leq X_j < U_m)$, an indicator function for some interval $[L_m, U_m)$. This allows us to break up the range of X_j into nonoverlapping regions, and thus fit a model with **piecewise constant** contribution for X_j.

Basis Functions



(a) Polynomial Basis Functions

(b) Radial Basis Functions

Figure: Some examples of commonly used basis functions. Here we assume that $X \in \mathbb{R}$.

Basis Functions

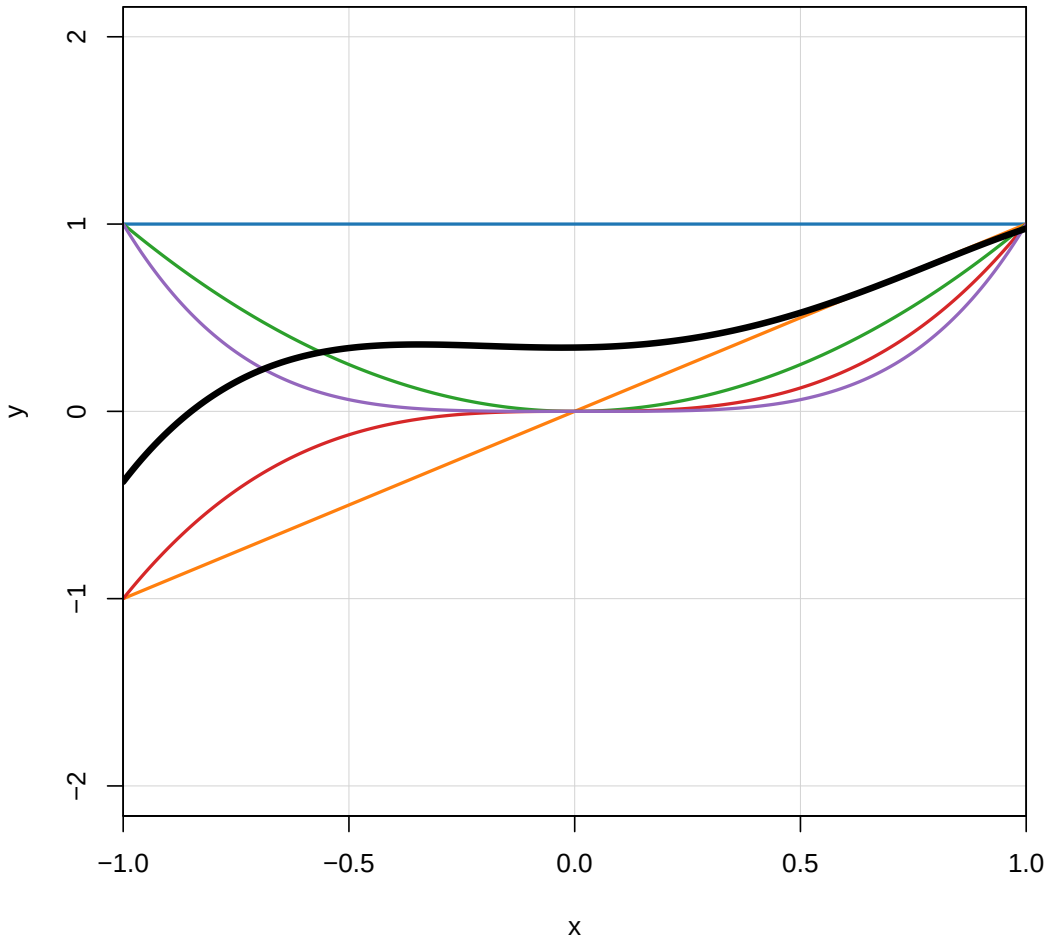


Figure: **A linear combination of the first five polynomial basis functions:** $f(X) = \sum_{i=1}^{5} \beta_i X^{i-1}$**, for** $X \in \mathbb{R}$. The parameters $\beta_1, \ldots, \beta_5$ are randomly sampled from a $\mathcal{N}(0, 1)$ distribution.

Basis Functions

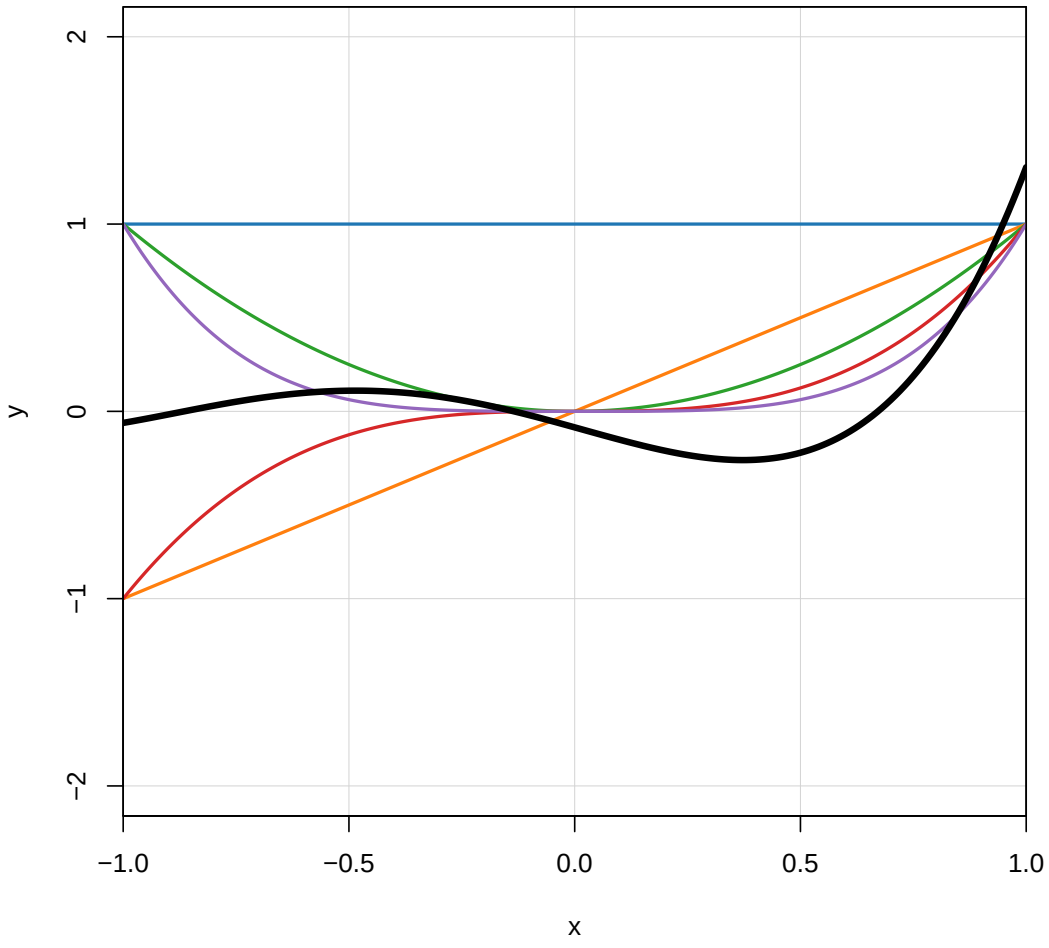


Figure: **A linear combination of the first five polynomial basis functions:** $f(X) = \sum_{i=1}^{5} \beta_i X^{i-1}$**, for** $X \in \mathbb{R}$. The parameters $\beta_1, \ldots, \beta_5$ are randomly sampled from a $\mathcal{N}(0, 1)$ distribution.

Basis Functions

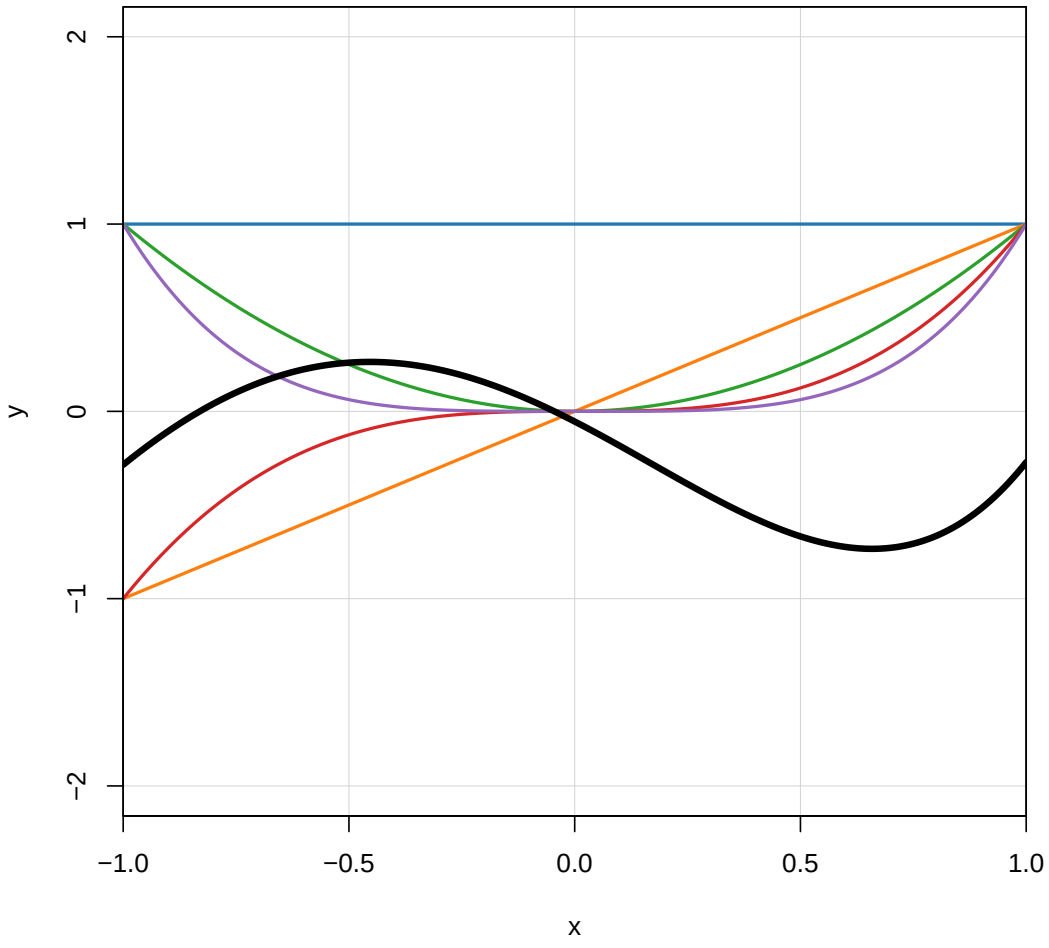


Figure: **A linear combination of the first five polynomial basis functions:** $f(X) = \sum_{i=1}^{5} \beta_i X^{i-1}$**, for** $X \in \mathbb{R}$. The parameters $\beta_1, \ldots, \beta_5$ are randomly sampled from a $\mathcal{N}(0, 1)$ distribution.

Basis Functions

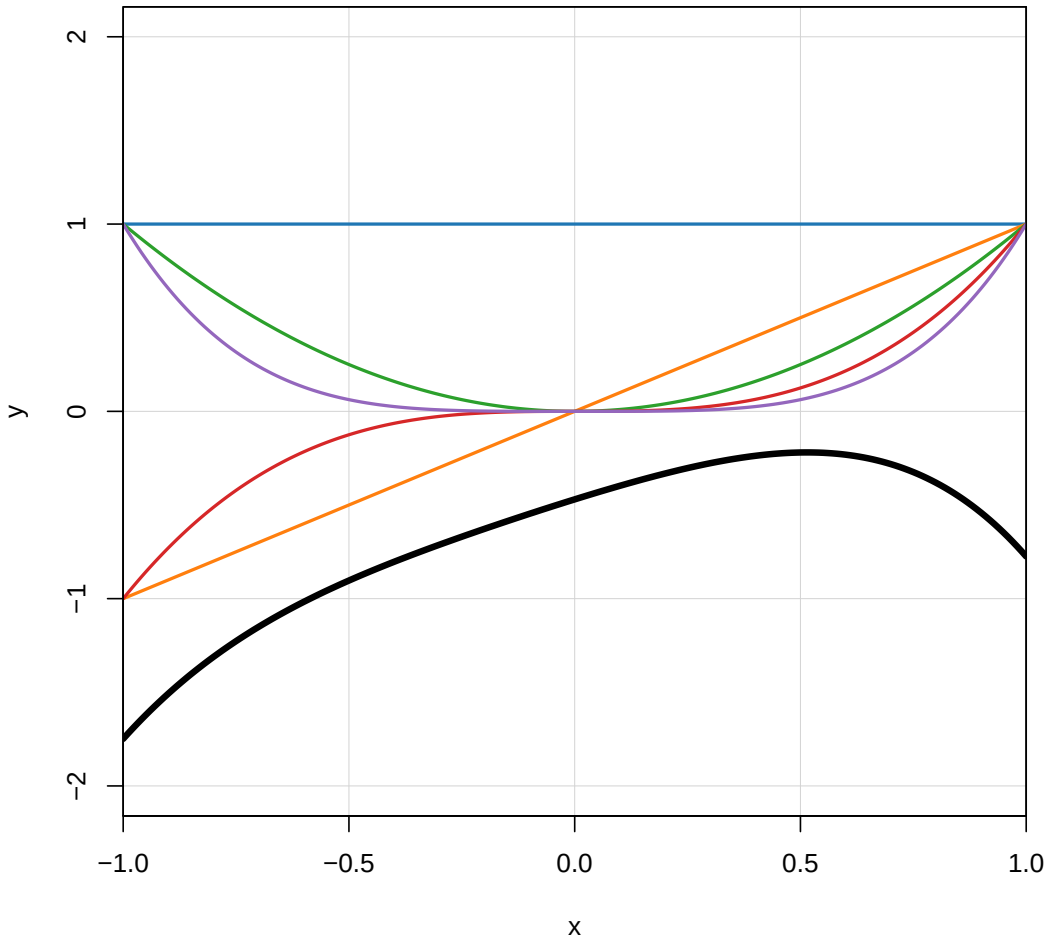


Figure: **A linear combination of the first five polynomial basis functions:** $f(X) = \sum_{i=1}^{5} \beta_i X^{i-1}$**, for** $X \in \mathbb{R}$. The parameters $\beta_1, \ldots, \beta_5$ are randomly sampled from a $\mathcal{N}(0, 1)$ distribution.

Basis Functions

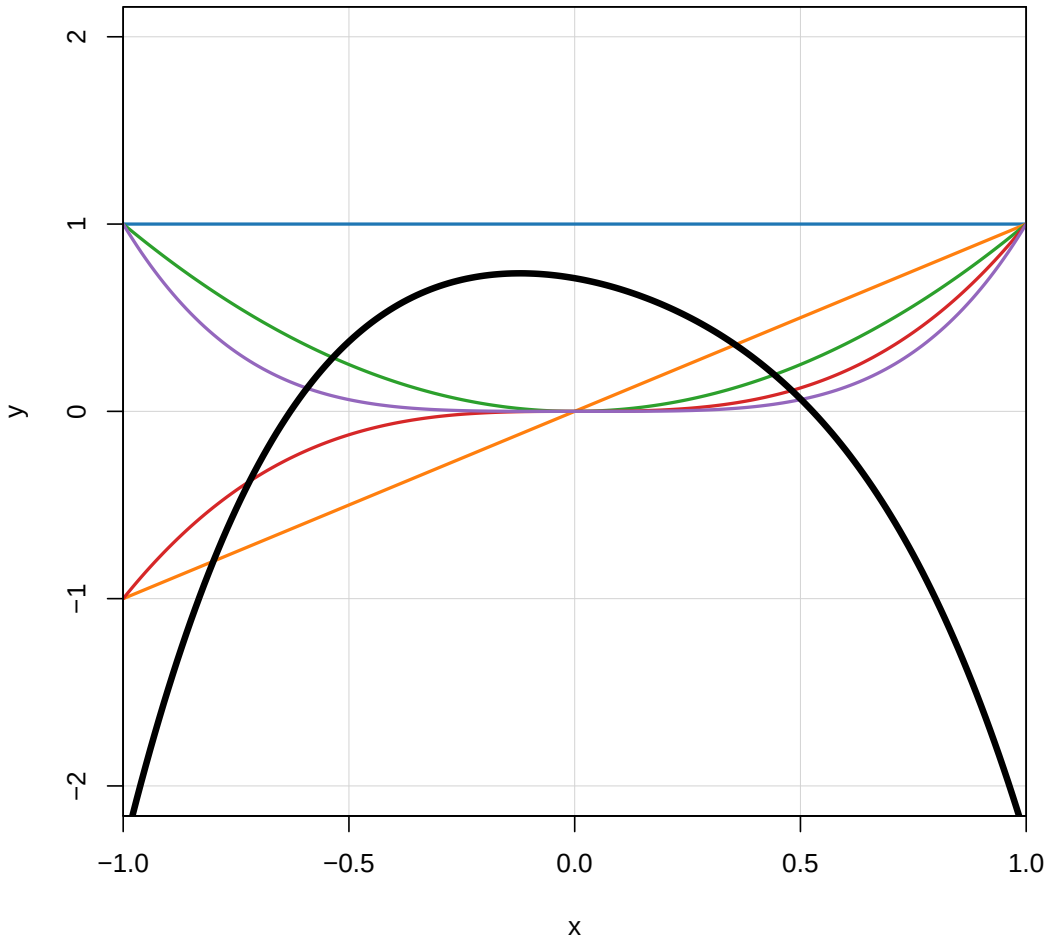


Figure: **A linear combination of the first five polynomial basis functions:** $f(X) = \sum_{i=1}^{5} \beta_i X^{i-1}$**, for** $X \in \mathbb{R}$. The parameters $\beta_1, \ldots, \beta_5$ are randomly sampled from a $\mathcal{N}(0, 1)$ distribution.

Basis Functions

In some cases, the problem will suggest a **specific choice** of the basis functions $(h_m)_{m=1}^M$, e.g., power functions, logarithms.

In most cases, however, the purpose of using a basis expansion will be achieving **greater flexibility** in our representation of $f(X)$.

We'll now consider two concrete examples: **piecewise polynomials** and **splines**. Tomorrow you'll encounter **Gaussian processes**.

Piecewise Polynomials

For simplicity, for the rest of this section we will assume the inputs are one-dimensional: that is, $X \in \mathbb{R}$.

A **piecewise polynomial** function $f(X)$ is obtained by dividing the domain of X into continuous intervals, and representing f by a separate polynomial on each interval.

The simplest case is when the polynomial on each region is degree 0, that is, a constant. This is sometimes referred to as **histogram regression**.

Piecewise Polynomials

For example, if we were to specify three basis functions, we would have

$$h_1(X) = I(X < \xi_1), \qquad h_2(X) = I(\xi_1 \leq X < \xi_2), \qquad h_3(X) = I(\xi_2 \leq X),$$

where $\xi_1 < \xi_2$ are the points of discontinuity, referred to as the **knots**.

In this case, since the basis functions are positive over disjoint regions, the least squares estimate of the model

$$Y = \sum_{m=1}^{3} \beta_m h_m(X) + \varepsilon \tag{10}$$

is simply given by

$$\hat{\beta}_m = \bar{Y}_m, \tag{11}$$

the mean of the response Y in the mth region.

Piecewise Polynomials

We can, of course, go beyond **piecewise constant** functions, and consider, say, **piecewise linear** functions.

In the simple example considered above, where we specified two knots $\xi_1 < \xi_2$, this would mean using an additional three basis functions, namely

$$h_4(X) = I(X < \xi_1)X$$
$$h_5(X) = I(\xi_1 \leq X < \xi_2)X$$
$$h_6(X) = I(\xi_2 \leq X)X.$$

Piecewise Polynomials

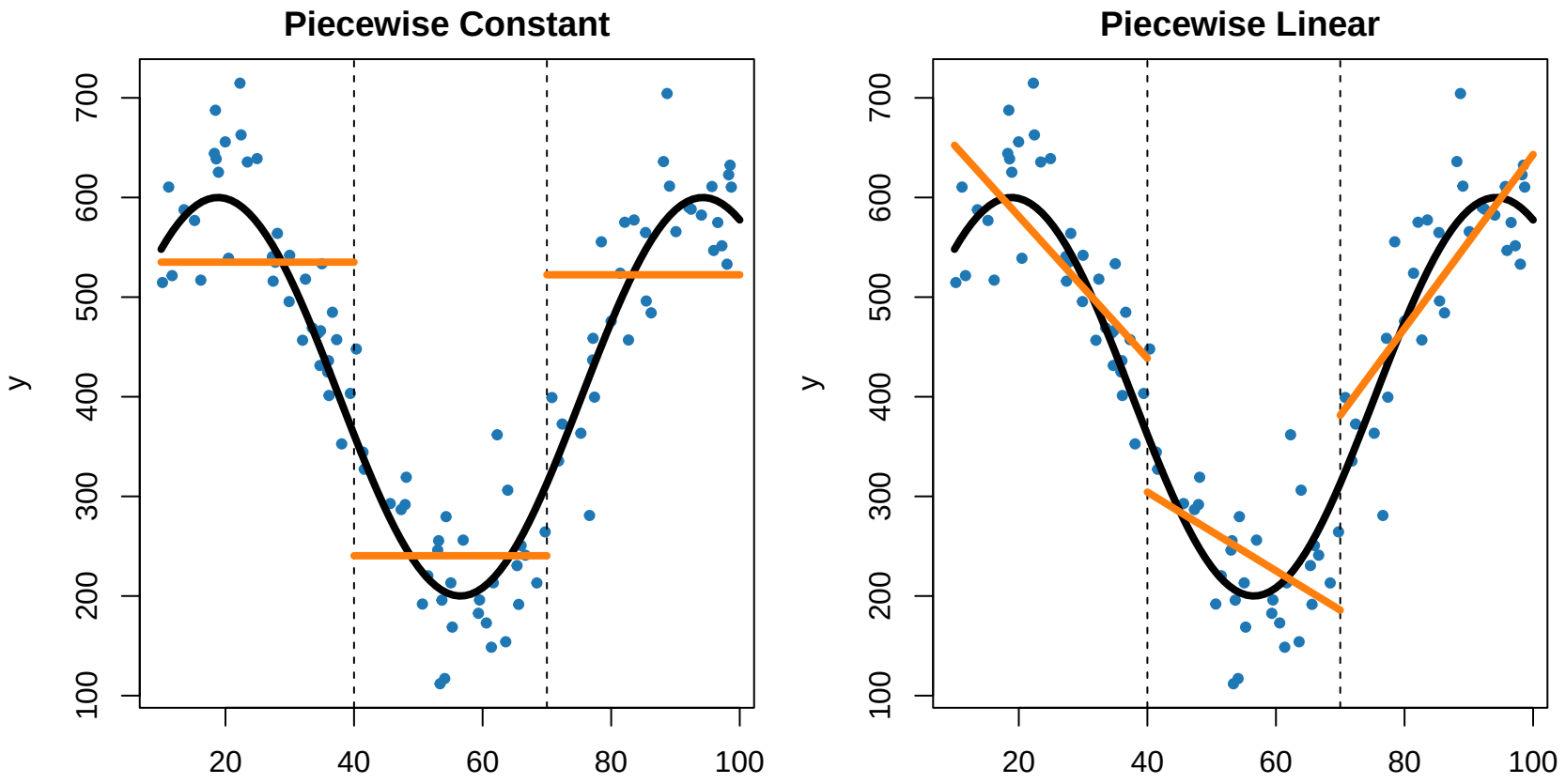


Figure: **Estimation using piecewise constant (left) and piecewise linear (right) basis functions.** The two panels show the piecewise constant (left) and piecewise linear (right) function fits to some artifical data. The artifical data (blue) were generated by adding Gaussian noise to the true curve (black).

Regression Splines

In general, it is typical to require that the fitted function f has some level of **smoothness** (e.g., continuity) at each knot.

In general, these continuity constraints imply a set of **linear constraints** on the model parameters.

For example, for the piecewise linear basis with two knots $\xi_1 < \xi_2$ considered above:

- the constraint that $f(\xi_1^-) = f(\xi_1^+)$ implies that

$$\beta_1 + \xi_1\beta_4 = \beta_2 + \xi_1\beta_5, \tag{12}$$

- the constraint that $f(\xi_2^-) = f(\xi_2^+)$ implies that

$$\beta_2 + \xi_2\beta_5 = \beta_3 + \xi_2\beta_6. \tag{13}$$

Thus, we now only have four free parameters, as opposed to six in the unconstrained case.

Regression Splines

In many cases, we may prefer smoother functions, which can be achieved by further increasing the order of the local polynomial (e.g., quadratic, cubic, etc.).

In the case of a **piecewise cubic** function, which also has continuous **first** and **second derivatives** at each knot, the function is known as a **cubic spline**.

It is often said that cubic splines are the lowest-order spline for which the knot-discontinuity (i.e., discontinuity in the third derivative) is not visible to the human eye. It is rare to go beyond cubic functions.

Regression Splines

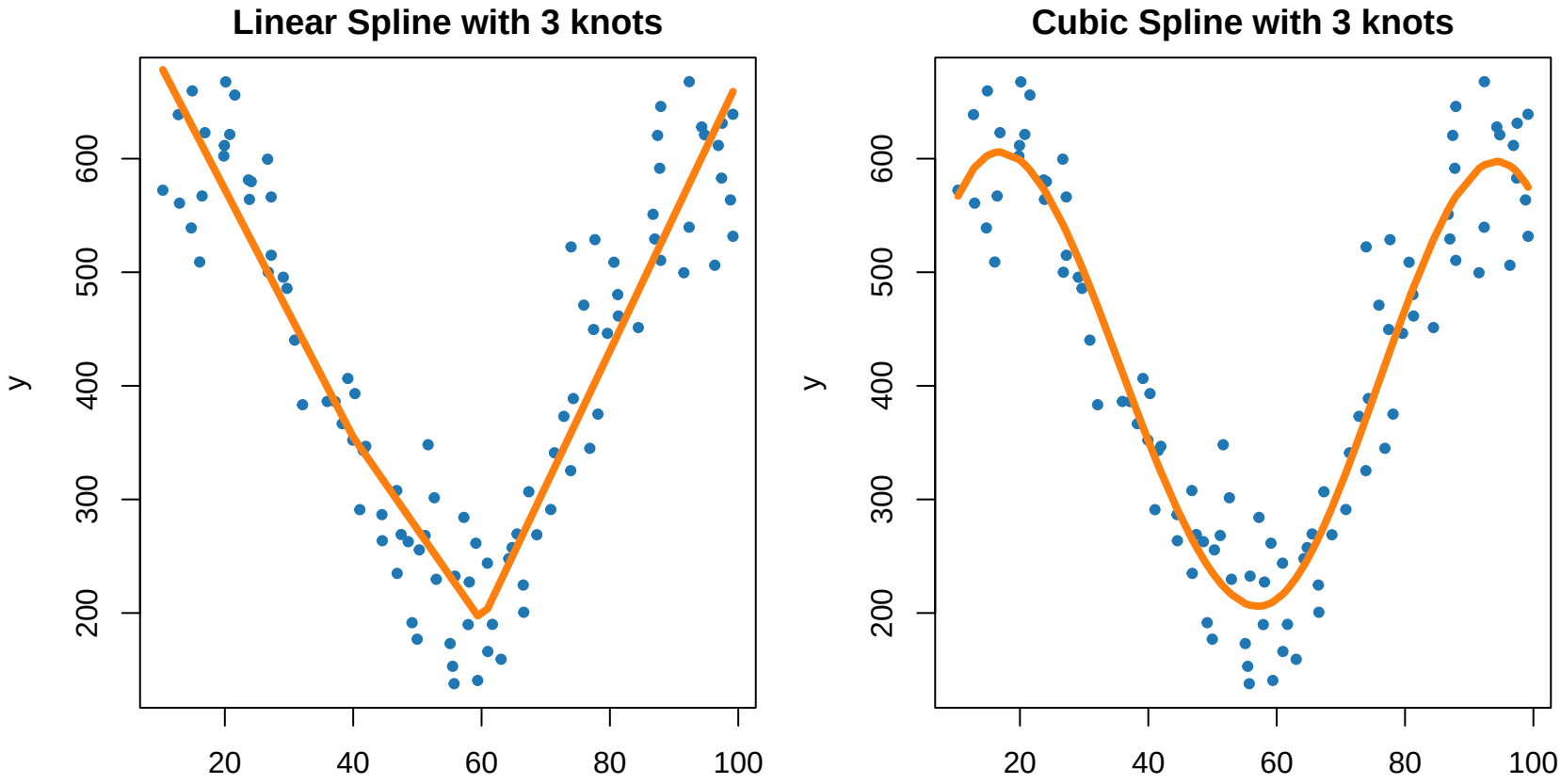


Figure: **Estimation using a linear spline (left) and a cubic spline (right).** The two panels show the linear spline (left) and cubic spline (right) fitted to the same artificial data used in the previous figure.

Recap and further questions

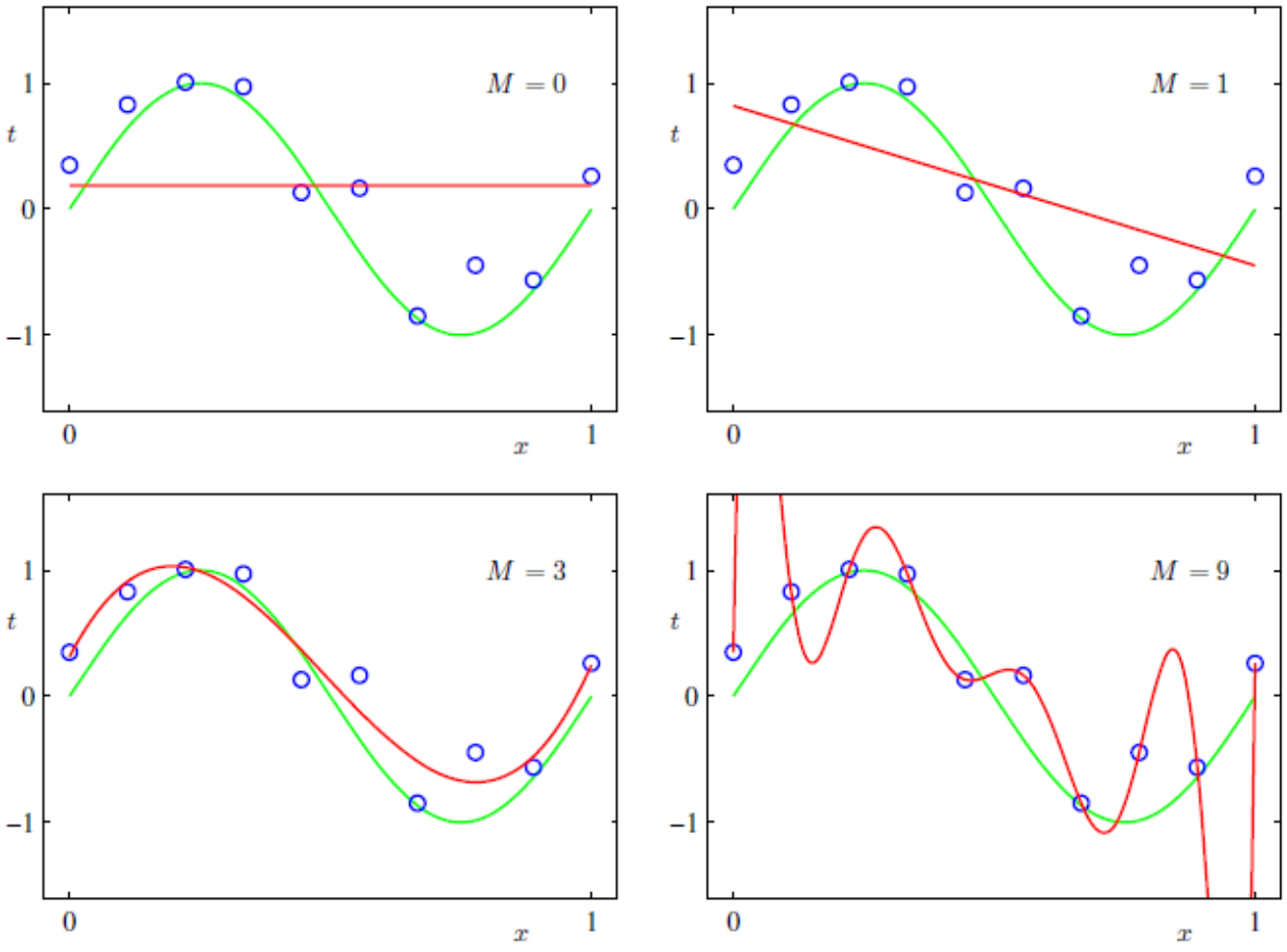
Recap:

1. Linear models
2. Least squares estimation
3. Bayesian regression
4. Basis expansions (beyond linearity)
5. Increasing flexibility

What haven't we answered yet?:

1. 'Infinitely' flexible, i.e. $M \to \infty$?
2. priors on coefficient β to priors on the function f?
3. How flexible is too flexible?

Model selection and regularisation



Penalisation methods

A typical strategy is to add a 'penalisation' term to the objective function (such as the least suares or likelihood function) which encourages certain characteristics of the solution. From a Bayesian viewpoint, one can encode this in the choice of prior. I'll now briefly cover some examples.

Penalise the number of parameters (elements of β)

These methods aim to select $k \leq p$ variables to include in the model, by minimisation of an appropriately penalised function. Some common choices including the **Akaike information criterion** (AIC), the **Bayesian information criterion** (BIC), and **Mallow's** C_p.

In the linear Gaussian case, these three criteria are given by

$$\mathrm{AIC}(k) = N \log\left(\frac{\mathrm{RSS}}{N}\right) + 2k \tag{14}$$

$$\mathrm{BIC}(k) = N \log\left(\frac{\mathrm{RSS}}{N}\right) + k \log(N) \tag{15}$$

$$C_p(k) = \mathrm{RSS} + 2k\hat{\sigma}^2_{\mathrm{full}} \tag{16}$$

where k denotes the number of variables included in the current model, and $\hat{\sigma}^2_{\mathrm{full}}$ denotes an estimate of the residual variance based on the full model.

Penalise the size of the parameters: shrinkage estimators

Ridge regression:

$$\hat{\beta}^{\mathrm{r}}_{\lambda} = \underset{\beta \in \mathbb{R}^p}{\arg\min} \left(\|\mathbf{y} - \mathbf{X}\beta\|_2^2 + \lambda \|\beta\|_2^2 \right),$$

where $\lambda \geq 0$ is a user specified **tuning parameter** which controls the degree of shrinkage. This is a **penalised** form of our RSS objective.

The larger the value of λ, the larger the amount of shrinkage. In particular:

- In the case that $\lambda = 0$, the penalty term has no effect, and we recover the standard RSS.
- Meanwhile, when $\lambda \to \infty$, all coefficients are shrunk towards 0.
- Bayesian approach: set prior mean of β to 0 and control the prior variance.

Penalise 'roughness' (or encourage smoothness!)

In fitting a smooth curve to a set of data, our general goal is to find some function, say $f(x)$, which fits the observed data well. That is, we want to find $f(x)$ which minimises

$$\mathrm{RSS} = \sum_{i=1}^{n} (y_i - f(x_i))^2 \tag{17}$$

There is, of course, a problem with this naive approach: if we don't include any constraints on $f(x)$, then we can always make the $\mathrm{RSS} = 0$ by choosing a function which **interpolates** the data. That is, $f(x)$ such that $y_i = f(x_i)$ at each x_i.

Instead, what we would like is a function that makes the RSS small, but that is also **smooth**.

Smoothing Splines

One approach is to consider the following optimisation problem.

Among all functions with two continuous derivatives, find the function f that minimises the penalised residual sum of squares

$$\mathrm{PRSS}(f, \lambda) = \sum_{i-1}^{N} (y_i - f(x_i))^2 + \lambda \int f''(t)^2 \mathrm{d}t, \tag{18}$$

where $\lambda \geq 0$ is a fixed **smoothing parameter**.

Smoothing Splines

The first term measures closeness to the data, while the second term penalises too much curvature in the function. The parameter λ controls the trade-off between these two terms:

- If $\lambda = 0$, f can be any function that interpolates the data. This may be detrimental, since such an f is likely to **overfit** the data.
- If $\lambda = \infty$, then f coincides with the OLS fit, since the second derivative must be identically zero.

Between these two extremes, the minimiser of $\mathrm{PRSS}(f, \lambda)$ will vary from very rough to vary smooth. The hope is that $\lambda \in (0, \infty)$ will index an interesting class of functions.

Bayesian perspective?

How can we achieve this from a Bayesian viewpoint with priors over function spaces? See tomorrow!

Bayesian Hierarchical Models

So far we have considered Bayesian problems of the form:

$$\theta \sim p(\theta),$$
$$y \sim p(y \mid \theta),$$

And are interest has been in computing, or sampling, from the posterior:

$$p(\theta \mid y) \propto p(y \mid \theta)p(\theta).$$

In some situations there may be structure in the model which suggests another 'level' to our model:

$$\gamma \sim p(\gamma)$$
$$\theta \sim p(\theta \mid \gamma),$$
$$y \sim p(y \mid \theta).$$

Bayesian Hierarchical Models

- Suppose y_i region specific data for a given country, with i indexing region. (For example y_1 denotes a rainfall reading in Capetown, y_2 in Pretoria, y_3 in Johannesburg etc...)
- Then we could take θ_i to be the mean of y_i.
- We model θ_i to vary with i to capture variation over the country and place a prior $\theta_i \sim p(\gamma)$

Bayesian Hierarchical Models

Our model then becomes:

$$\begin{aligned}
\gamma &\sim p(\gamma) \\
\theta_i &\sim p(\theta \mid \gamma), \text{ for } i = 1, \ldots, n, \\
y_i &\sim p(y \mid \theta_i), \text{ for } i = 1, \ldots, n.
\end{aligned}$$

The posterior for θ, γ is:

$$p(\theta, \gamma) = p(y \mid \theta) p(\theta \mid \gamma) p(\gamma).$$

Bayesian Hierarchical Models

- Bayesian hierarchical modeling allows us borrowing strength between related groups, giving to more stable and reliable inferences.
- Handling of Complex Data Structures: It allows us to effectively model nested, grouped, or multi-level data (e.g., patients within hospitals, students within schools), capturing both individual and group-level variations.
- Can be efficiently implemented using probabilistic programming languages (PPLs), such as Stan or Numpyro to estimate marginal distributions and posterior quantities. More on PPLs throughout this course!

Thanks!

Any questions?