Topic 3
Generalized additive models (GAMs)

3.1 "Extending" the GLM

- In this topic we try to extend the GLM framework on the basis that it offers a unifying modelling and inferential framework for all distributions within the exponential family but is limited in modelling flexibility.

- In particular, we focus on extending the specification of how covariates $\boldsymbol{x}$ enter the model:

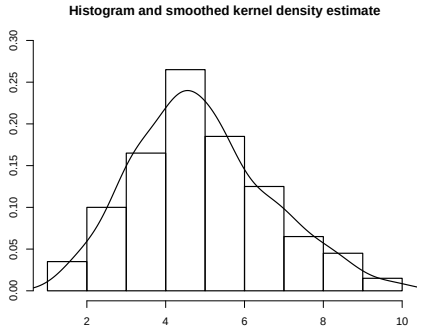
$$g(\mu(\boldsymbol{x})) = f(\boldsymbol{x};\boldsymbol{\theta}) \quad \text{rather than the old} \quad g(\mu(\boldsymbol{x})) = \boldsymbol{x}'\beta$$

  where $f(\cdot)$ is an unknown **smooth** function to be estimated **non-parametrically**.

3.1 Non-parametric models

- **Non-parametric** models are distinguished by not specifying in advance a parametric form for the probability distribution or the relationship of the response to the covariates $\boldsymbol{x}$. The object is to estimate things directly, rather than through imposing a parametric form—*we let the data decide!*

- A histogram or a smoothed density estimate (smoothed histogram) is one example of a non-parametric model that we have come across. We only need to choose the number of bins (or amount of smoothing).



3.1 Non-parametric models

- For quantifying associations, e.g. between a continuous variable and a factor, we could have a histogram for each level of a factor.
- For instance, could have a histogram of rainfall for each season.
- For continuous variables y and x, we could adopt the idea of **moving averages** to flexibly estimate or visualize the relationship.
- The general concept is to have a moving window over the range of x, so that the estimate of y at the center of the window is the (weighted) average of y's falling within that window.
- Two particularly useful methods based on the idea of moving averages are **kernel smoothing** and **loess** (LOcally Estimated Scatterplot Smoothing).
- See `topic3.R` for an implementation of these.

3.2 Non-parametric models

- Non-parametric models are appealing and seem more "objective" and flexible, e.g. we can have a histogram of y_i for each level of a categorical covariate x_i rather than having to decide what the distribution of y_i is and how x_i affects it.
- Many of the algorithms in the field of **machine learning** can be thought of as non-parametric models.
- There are however issues associated with non-parametric models:
  - They are data-intensive, i.e. require a lot of data for estimation;
  - The idea of "parameters" is not completely gone e.g., there is still the matter of choosing the number of bins in a histogram;
  - Inference is possible but computationally intensive (more on this later);
  - Not very useful for predicting out-of-sample data (e.g.probability of exceeding the maximum data point from the histogram?)

3.2 Parametric versus non-parametric models

- Nevertheless it is tempting to think that non-parametric models will automatically be 'better' than parametric models because they make less assumptions about the data.
- Certainly non-parametric approaches are valuable and will tend to explain the data better (in some sense). But:
  - Statistical modelling is about **explaining** data not just **reproducing** it—one needs to guard against **over-fitting**.
  - "parametric structures" in the data allow us to incorporate scientific knowledge and enable the explicit testing of scientific questions.
- E.g., the saturated model is basically the ultimate non-parametric model: it reproduces the data exactly without the need to specify how covariates enter the model. It is however useless at testing the difference in the mean response between two groups such as "treatment" and "control".

3.2 Semi-parametric models

- It can then be argued that the most practically useful approach is one where the model is a mixture of non-parametric and parametric elements—so called **semi-parametric** models.
- Most traditional statistical models are **parametric** (e.g. GLMs).
- The form (linear, logistic etc.) of the function $\mu(\boldsymbol{x}_i)$ representing the mean is specified in advance and involves a number of unknown parameters whose values then need to be estimated from the data e.g.

$$\mu(x_i) = \beta_0 + \beta_1 x_i$$

- Similarly the form of the random component (Normal, Poisson, Binomial etc.) is also specified in advance and may (or may not) involve additional unknown auxiliary parameters which also need to be estimated from the data (e.g. its variance, if this can be specified independently of the mean) e.g.,

$$Y_i \sim \mathrm{N}(\mu(x_i), \sigma^2)$$

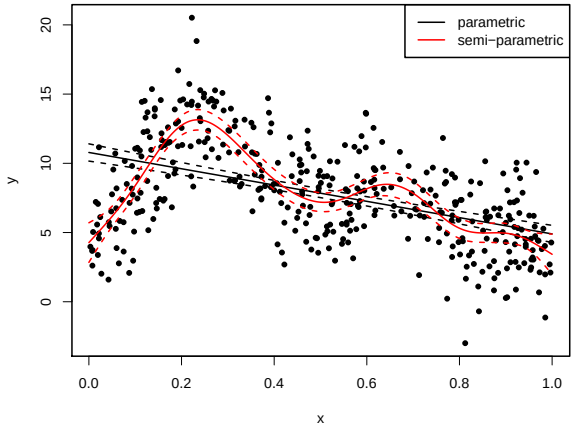
3.2 Semi-parametric models

- **Semi-parametric** models are distinguished by not specifying in advance a parametric form for $\mu(\boldsymbol{x}_i)$. The object is to estimate $\mu(\boldsymbol{x}_i)$ directly, rather than to estimate the relationship through imposing a parametric form.
- However, estimating an unknown function $\mu(\boldsymbol{x}_i)$ rather than a handful of parameters $\boldsymbol{\beta}$ is not a straightforward task. So we still need to assume something about $\mu(\boldsymbol{x}_i)$ in order to be able to come up with a method to fit the model.
- Most methods assume that $\mu(\boldsymbol{x}_i)$ is a **smooth** function and then proceed to estimate it using an appropriate fitting method (e.g. 'penalised likelihood').
- Note: a 'smooth' function is continuous and has continuous derivatives up to a certain order (it is differentiable and its derivative is a continuous function).
- In the methods that we will see in this topic, the idea is to represent the function $\mu(\boldsymbol{x}_i)$ in a low-dimension (in terms of number of parameters) while still treating it as an unknown function.

3.3 Semi-parametric models: Goal

- Example: consider the Gaussian model $Y_i \sim N(\mu(x_i), \sigma^2)$ with two possible formulations for the mean, parametric and semi-parametric respectively:

$$\mu(x_i) = \beta_0 + \beta_1 x_i \quad \text{and} \quad \mu(x_i) = \beta_0 + f(x_i) \quad \text{where } f() \text{ is smooth}$$



- The black line is "imposed" to the data whereas the smooth red line is allowed to flexibly explain the mean.

3.3 Additive models using splines

- Start by "extending" the linear model. An additive model is defined here as

$$\begin{aligned} Y_i &\sim N\left(\mu(\boldsymbol{x}_i), \sigma^2\right) \\ \mu(\boldsymbol{x}_i) &= \boldsymbol{z}'\beta + f_1(x_{1i}) + f_2(x_{2i}) + \cdots + f_p(x_{pi}) \end{aligned}$$

where $\boldsymbol{x} = (\boldsymbol{z}, x_1, \ldots, x_p)$, $\boldsymbol{z}'\beta$ is the familiar linear predictor and $f_j(\cdot)$ are smooth functions.

- E.g., suppose we want to model overall water demand for a particular UK county and interest lies on whether this changes for temperatures below 5° (cold causes breakage), and we have daily data for a number of years. A possible model might be

$$\begin{aligned} Y_i &\sim N\left(\mu(\boldsymbol{x}_i), \sigma^2\right) \\ \mu(\boldsymbol{x}_i) &= \beta_0 + \beta_1 z_i + f(x_i) \end{aligned}$$

where $z_i = 0$ or 1 if temperature is above or below 5°, and x_i is time in days. Interest lies on β_1, while the unknown function $f(x_i)$ allows for water demand fluctuations in time from factors that we haven't measured (such as other weather/climate information, holidays etc.)

3.3 Additive models using splines

- This additive model allows flexible specification of the dependence of the response on the covariates, however it comes at the cost of two theoretical questions:
  1. How to specify the smooth functions?
  2. How smooth should they be?
- A third challenge is whether we can answer the first question in a way as to end up with an inferential framework to parallel the familiar normal theory linear model.
- Well, one way to go is to use **penalised regression splines** to represent the functions whereas the appropriate degree of smoothness can be estimated from the data using **cross validation**. For brevity we restrict discussion to a single covariate $\mu(x_i) = f(x_i)$ and return to the general case later.
- Cross-validation is a technique where a subset of the data is left out when fitting the model, and then using the fitted model to predict those data points.

3.4 Basis functions

- A key idea is to represent the unknown smooth function in terms a convolution of much simpler functions by assuming a particular **basis**. Choosing a **basis** for a function $f(\cdot)$ implies that one is defining the space of functions for which $f(\cdot)$ is an element.
- Selecting a basis essentially means that function $f(\cdot)$ can be represented as a sum of **basis functions** $b_k(\cdot)$:

$$f(x) = \sum_{j=1}^{q} \beta_k b_k(x)$$

for particular values of the unknown parameters β_k.

- We have actually come across **polynomial basis** before, e.g. if we believe that $f(\cdot)$ is a 4th order polynomial then

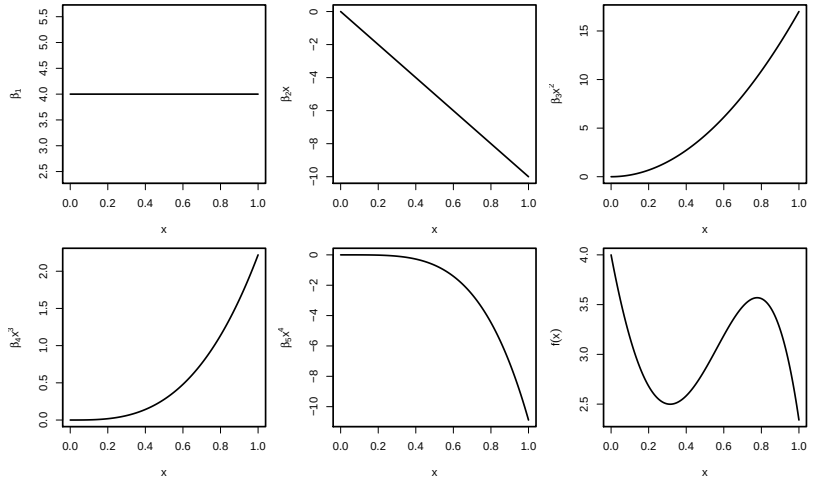
$$f(x) = \beta_1 + \beta_2 x + \beta_3 x^2 + \beta_4 x^3 + \beta_5 x^4$$

so that the basis functions are $b_1(x) = 1$, $b_2(x) = x$, $b_3(x) = x^2$ and $b_4(x) = x^3$, $b_5(x) = x^4$.

3.4 Polynomial basis functions

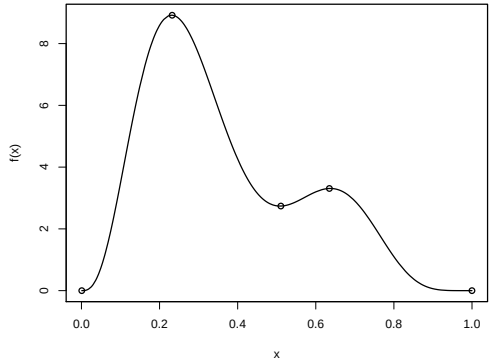
$$f(x) = \beta_1 + \beta_2 x + \beta_3 x^2 + \beta_4 x^3 + \beta_5 x^4$$

with $\beta_1 = 4, \beta_2 = -10, \beta_3 = 17, \beta_4 = 2.22$ and $\beta_5 = -10.88$.



3.4 Cubic spline basis functions

- Although a polynomial basis is straightforward, they can have undesirable properties in regions with sparse data and can become numerically unstable for large orders.
- A cubic spline is a curve made up of sections of cubic polynomials (the splines), joined together at points known as **knots**. At the knots, the splines are continuous in value as well as first and second derivatives.



3.4 Cubic spline basis functions

- Conventionally the knots are placed at each data point, however for the regression splines that we are aiming for, the knots will have to be chosen—typically at equidistant points along the range x or at empirical quantile values of x.
- Given knot values at x_j^*, $j = 1, \ldots, q$, there many possibilities for the choice of basis functions $b_j(\cdot)$ to represent cubic splines. Noting that these are essentially equivalent, a simple basis to use is:

$$b_1(x) = 1, \;\; b_2(x) = x \text{ and } b_{k+2}(x) = |x - x_k^*|^3 \text{ for } k = 1, \ldots, q-2$$

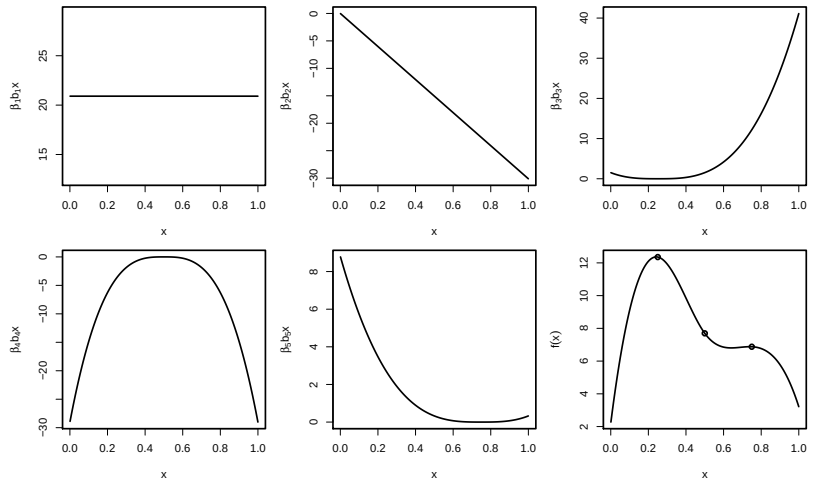
  where q is the basis dimension or **rank**.
- Note that the model is still linear in the parameters so that a row of the model matrix $\boldsymbol{X}$ is

$$\boldsymbol{x}_i = \big(1, x_i, b_3(x_i), \ldots, b_q(x_i)\big)$$

  and least squares can be used to estimate the β_k in $\mu(\boldsymbol{x}_i) = \boldsymbol{x}_i'\boldsymbol{\beta}$.

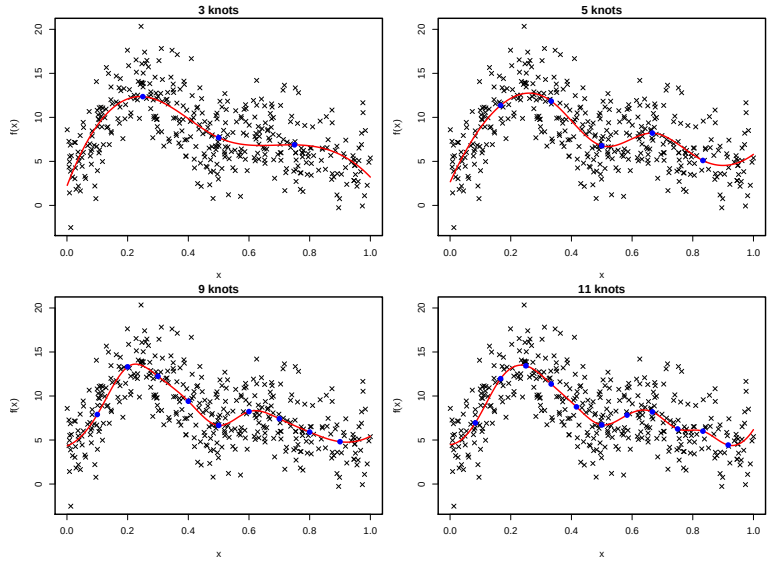
3.4 Cubic spline basis functions

- A rank 5 example of the cubic basis defined earlier, with knots at $x = 0.25, 0.5, 0.75$:



3.5 Cubic spline basis functions

- As we might expect, increasing the basis dimension q (or equivalently the number of knots) will decrease the smoothness or increase the "**wiggliness**" of the estimated function.



3.5 Penalising the degree of smoothing

- So, representing the unknown function $f(x)$ using splines is very convenient as we can essentially do it using linear model machinery (i.e. least squares).

- However, the choice of rank (i.e. the number of knots $+2$) seems arbitrary, and this is a problem:
  - Choosing q to be too small implies the function is not flexible enough to capture the true relationship;
  - Choosing q to be too large could lead to **over-fitting**.

- The problem is effectively one of model selection, so we might try an approach based on the likelihood ratio test where we have a grid of descending values of q and try **backwards selection**.

- However, this is more difficult than is seems, because models with different numbers of equidistant knots are **not nested**. So an alternative approach is needed!

3.5 Penalising the degree of smoothing

- Alternatively we could keep the basis dimension fixed at a size larger than we think reasonable, and control the smoothness by adding a "wiggliness" penalty to the least squares objective function. For instance, rather than minimising $\sum_{i=1}^{n}(y_i - \boldsymbol{x}_i'\beta)^2$, we minimise **penalised least squares**

$$\sum_{i=1}^{n}\left(y_i - \boldsymbol{x}_i'\beta\right)^2 + \lambda\int_x \left[f''(x)\right]^2 dx$$

  which penalises functions $f(x)$ that are too wiggly through their second derivative.

- The trade-off between model fit and smoothness is controlled by the **smoothing parameter** λ (similar to the bandwidth and span in scatter plot smoothers):
  - $\lambda \to \infty$ leads to a straight line;
  - $\lambda \to 0$ results in an un-penalised regression spline estimate.

- Note that this is a special case of **penalised likelihood**, which is written as $L(\boldsymbol{\theta}; \boldsymbol{y}) + \lambda\int_x \left[f''(x)\right]^2 dx$.

3.5 Penalising the degree of smoothing

- Although the integral looks "nasty", the fact that $f(x)$ is linear in the parameters implies that

$$\int_x \left[f''(x)\right]^2 dx = \beta'\boldsymbol{S}\beta$$

  where $\boldsymbol{S}$ is a $q \times q$ matrix involving the spline basis and its first two derivatives.

- In fact, it can be shown that given a value for the smoothing parameter λ, the penalised least squares problem can be solved using ordinary least squares with an amended model matrix $\boldsymbol{X}$.

- So we can still fit the additive model using linear regression machinery and we have a single parameter λ to control overfitting. But how do we estimate λ?

3.6 Estimation of the penalty in additive models

- The ideal value of λ, should be such that the estimated value of the function at $\hat{f}(x_i)$ at x_i, is as close as possible to the true function $f(x_i)$. A suitable criterion might then be to choose λ to minimise

$$M = \frac{1}{n}\sum_{i=1}^{n}\left[f(x_i) - \hat{f}(x_i)\right]^2$$

- However, $f(\cdot)$ is of course unknown so M cannot be used directly, but we can consider the expected squared error in predicting a new data y_s:

$$\frac{1}{m}\sum_{s=1}^{m}\left[y_s - \hat{y}_s\right]^2 = \frac{1}{m}\sum_{s=1}^{m}\left[f(x_s) + \epsilon_s - \hat{f}(x_s)\right]^2 = \mathbb{E}(M) + \sigma^2$$

- It turns out that if we consider the **ordinary cross validation score**

$$V = \frac{1}{n}\sum_{i=1}^{n}\left[y_i - \hat{f}^{[-i]}(x_i)\right]^2$$

where $\hat{f}^{[-i]}(x_i)$ is the prediction from the model fitted to all data except y_i, then for a large enough sample $\mathbb{E}(V) \approx \mathbb{E}(M) + \sigma^2$.

3.6 Estimation of the penalty in additive models

- Of course V is the score that results from leaving one data point out in turn and refitting the model, which provides a practical measure that we can use to estimate λ—i.e. choose λ that minimises V.

- Leave-one-out techniques are known to be computationally expensive, so in practice V is approximated from the fitted model rather than computed directly. The R package `mgcv` that we will use, employs what is called a Generalised Cross Validation (GCV) measure to produce estimates of λ.

- Note also that minimising the expected predictive error is rather intuitive, when the goal is to avoid overfitting—we came across this when discussing the AIC as a measure for model selection. In fact the AIC can also be used to estimate λ on the basis that is an estimate of out-of-sample predictive power of a model.

3.7 Additive models

- It is straightforward to extend the discussion so far to the additive model $\mu(\boldsymbol{x}_i) = \boldsymbol{z}'\beta + f_1(x_{1i}) + \cdots + f_p(x_{pi})$, which can be estimated using penalised least squares, assuming each smooth function $f_j(\cdot)$ is represented by a spline basis and where there is one smoothing parameter λ_j for each.

- Note that to avoid having multiple intercepts in the model, we set the first term of each spline representation to zero, $b_1(x) = 0$ and allow the model to only have one intercept, β_0. This constrains each $f_j(\cdot)$ to be centred at zero.

- Note also that additive models make a strong assumption in that the complex smooth function $f(x_1, \ldots, x_p)$ can be represented as $f_1(x_1) + \cdots + f_p(x_p)$. We return to this point later.

3.7 Generalised Additive models (GAMs)

- **Generalized Additive Model** (GAMs) extend the additive model as the GLM extends the linear model: a link function relates the mean to a non-linear smooth function of the covariates and the response may follow any distribution from the exponential family, or simply have a known mean-variance relationship permitting use of the quasi-likelihood approach.
- The linear predictor is of the form:

$$g(\mu_i) = \eta_i = \boldsymbol{z}'\beta + f_1(x_{1,i}) + \cdots + f_p(x_{p,i})$$

- A GAM can be fitted by penalised likelihood which however requires its own penalized iterative re-weighted least squares (P-IRLS) algorithm—iteratively fitting weighted additive models in an analogous way as the IRLS procedure relates to ordinary least squares.
- Smoothing parameters can be estimated using the GCV mentioned earlier (or other equivalent measures such as the AIC).

3.7 Interactions in GAMs

- In theory, GAMs can be easily extended to allow for interactions (or equivalently more complex functions):

$$g(\mu_i) = \eta_i = \boldsymbol{z}'\beta + f_{1,2}(x_{1,i}, x_{2,i}) + f_3(x_{3,i}) + \cdots$$

- In practice this is more difficult and one needs to choose the basis functions carefully, as high dimensional functions need a huge amount of data to be estimated (**curse of dimensionality**).

- A particularly useful basis function are so-called **thin-plate splines**, which actually avoid the need to place knots and have desirable properties when estimating high-dimensional functions.

- We do not go into detail here, but note that the function `gam()` in package `mgcv` uses thin-plate splines by default, although other spline functions are also implemented (such as the cubic splines used earlier).

3.8 Inference in GAMs

- Conditional on the estimated values of the smoothing parameter(s), inference is essentially the same from GLMs, i.e. based on likelihood theory.

- Note that inference will not be *exactly* the same since conditioning on the obtained values of the smoothing parameters ignores the uncertainty of the having to estimate them. This is similar to ignoring the uncertainty in estimating the dispersion parameter in GLMs and thus using z-tests rather than t-tests.

- The same holds for deviance tests. The R function `gam()` in package `mgcv` will produce the **effective degrees of freedom** (EDF) derived as the number of parameters minus the number of constraints, and use these to derive approximate χ^2 or F tests for the significance of each smooth function.

- EDF can be then used to perform goodness of fit tests and model selection based on the scaled deviance (see Topic 2)—again conditional on the smoothing parameters.

3.8 Discussion - parametric versus non-parametric models

- We have extended the GLM to the GAM, so we now have a "suite" of very flexible models to apply to real-world problems, noting that smooth functions of covariates will in general improve model fit.

- Smooth functions in the linear predictor are very useful in flexibly allowing for "nuisance" influence from covariates and also in investigating unknown relationships between the response and various covariates.

- Unlike parametric models such as GLMs, inference in non-parametric models is generally more difficult and cumbersome. E.g., for the GAM we can do likelihood inference but only conditional on the estimates of the smoothing parameters.

- To quantify uncertainty in the estimation of the smoothing parameters would require more sophisticated techniques, beyond the scope of this module.