QBUS6810: Statistical Learning and Data Mining

Lecture 10: Nonlinear Modelling

Semester 1, 2019

Discipline of Business Analytics, The University of Sydney Business School

Lecture 10: Nonlinear Modelling

1. Basis functions
2. Regression splines
3. Smoothing splines
4. Local regression
5. Generalised additive models

Nonlinear modelling

Our general framework for regression problems is the additive error model

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

where $f(\cdot)$ is an unknown regression function.

- In this lecture, we move beyond linear specifications for f to study methods that can approximate arbitrary functions f.
- We consider the case of a single predictor for most of this lecture, before extending the methodology to multiple predictors in the last section.

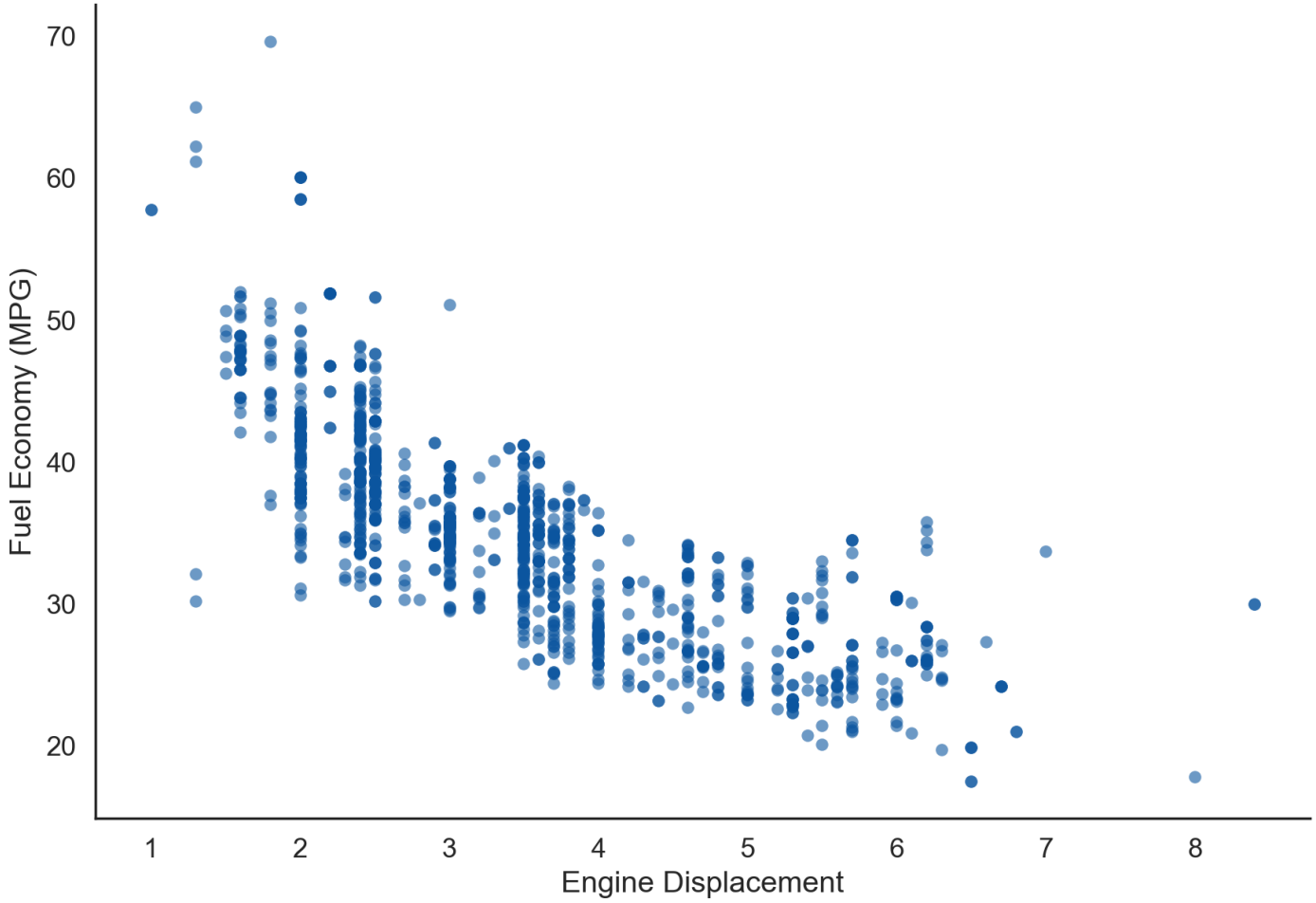
Example: Fuel Economy

- This example uses data extracted from the `fueleconomy.gov` website run by the US government, which lists different estimates of fuel economy for passenger cars and trucks.

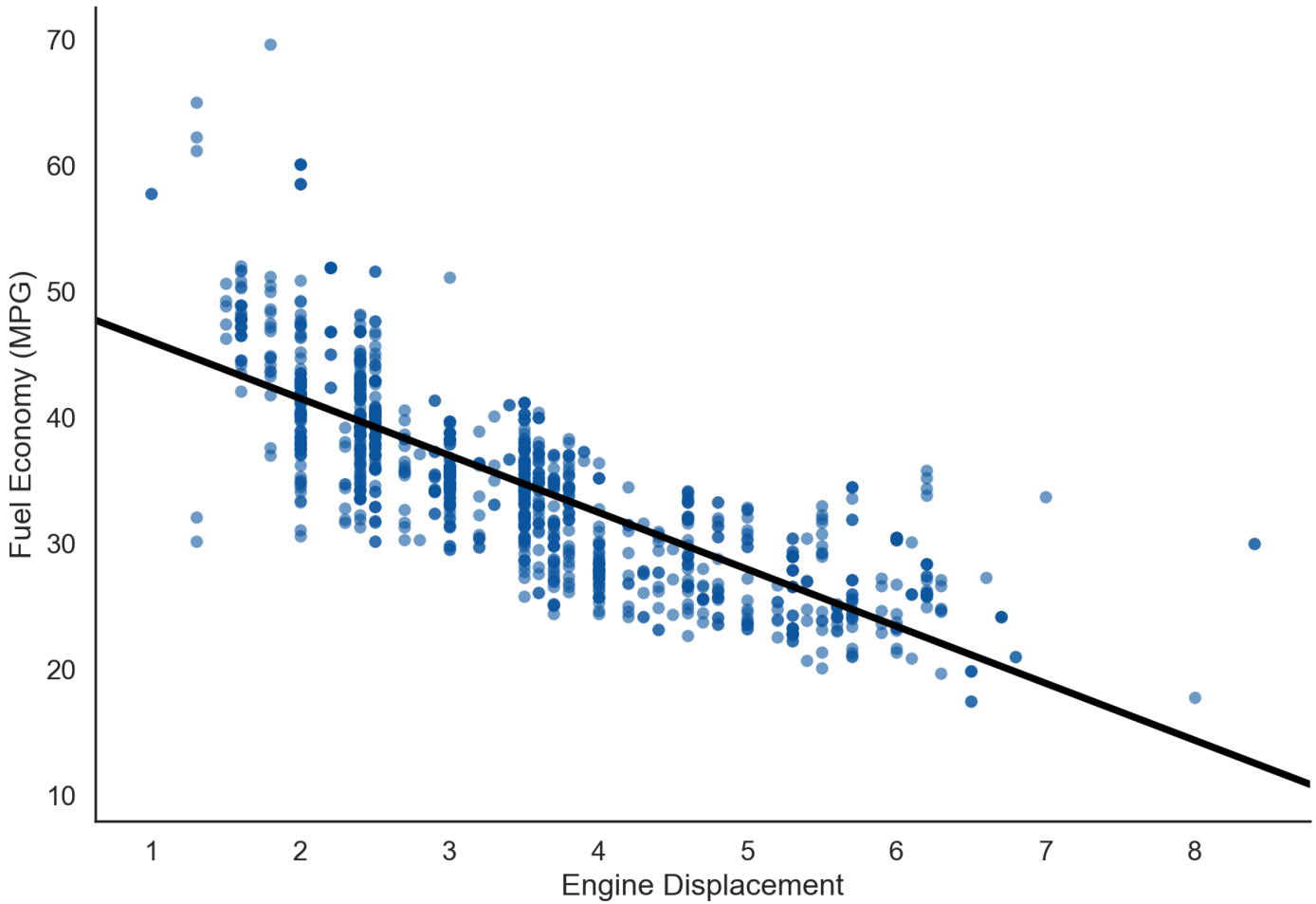
- For each vehicle in the dataset, we have information on various characteristics such as engine displacement and number of cylinders, along with laboratory measurements for the city and highway miles per gallon (MPG) of the car.

- We here consider the unadjusted highway MPG for 2010 cars as the response variable, and a single predictor, engine displacement.

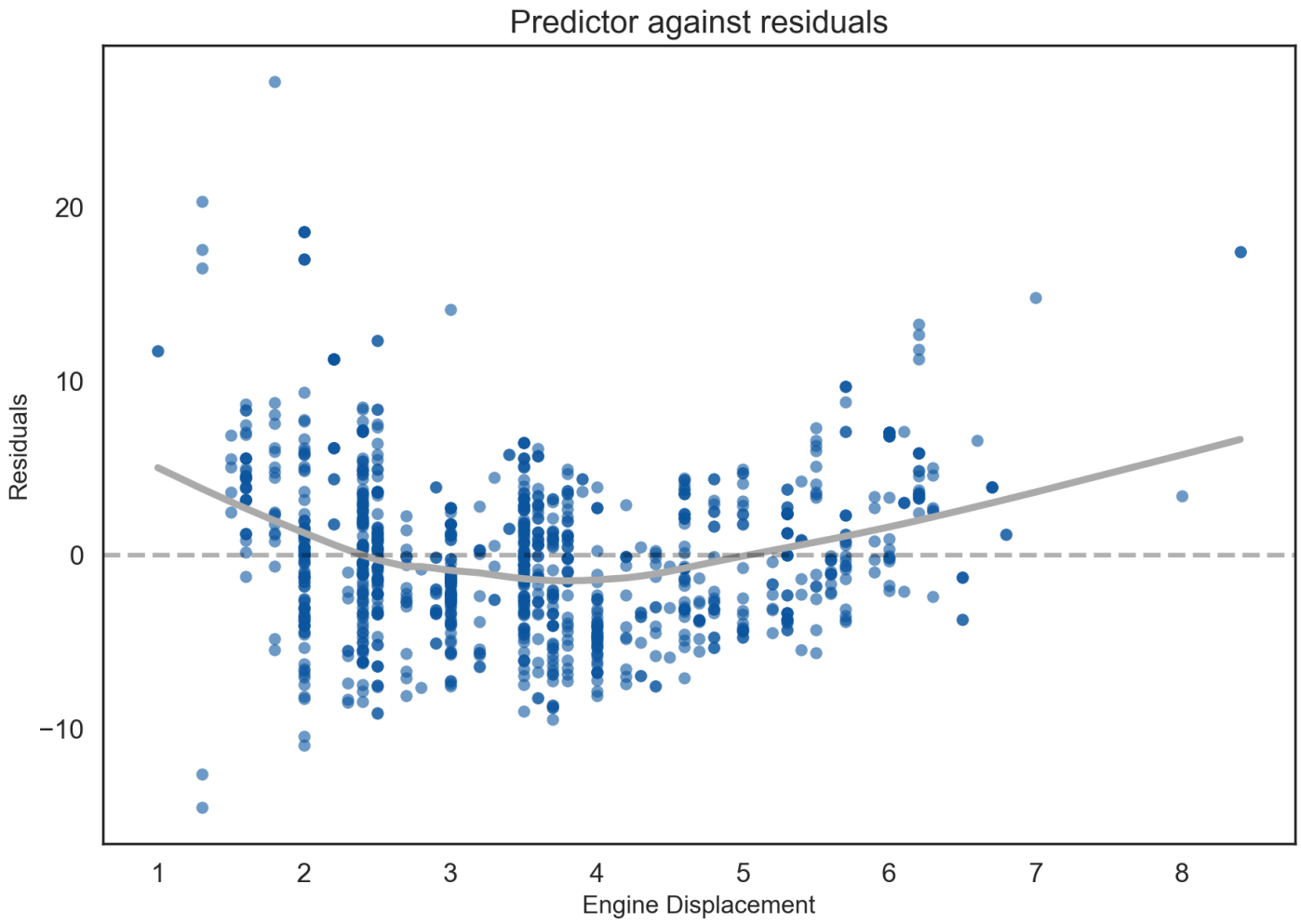
Example: Fuel Economy



Example: Fuel Economy



Example: Fuel Economy



Polynomial regression

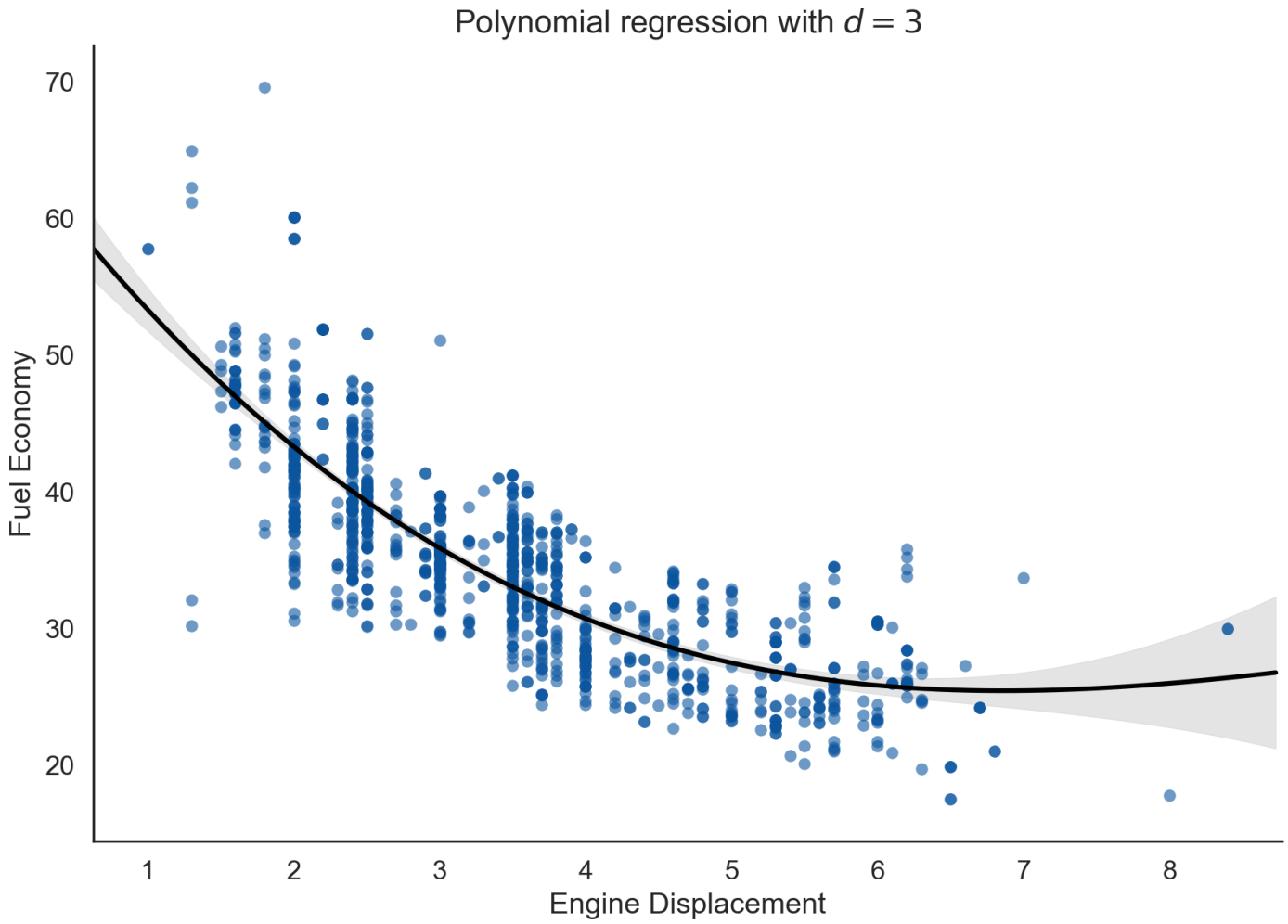
Earlier in the semester we discussed the polynomial regression model

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \ldots + \beta_d X^d + \varepsilon,$$

where d is the polynomial degree.

This simple approach can work well in some cases, and is a useful benchmark.

Example: Fuel Economy



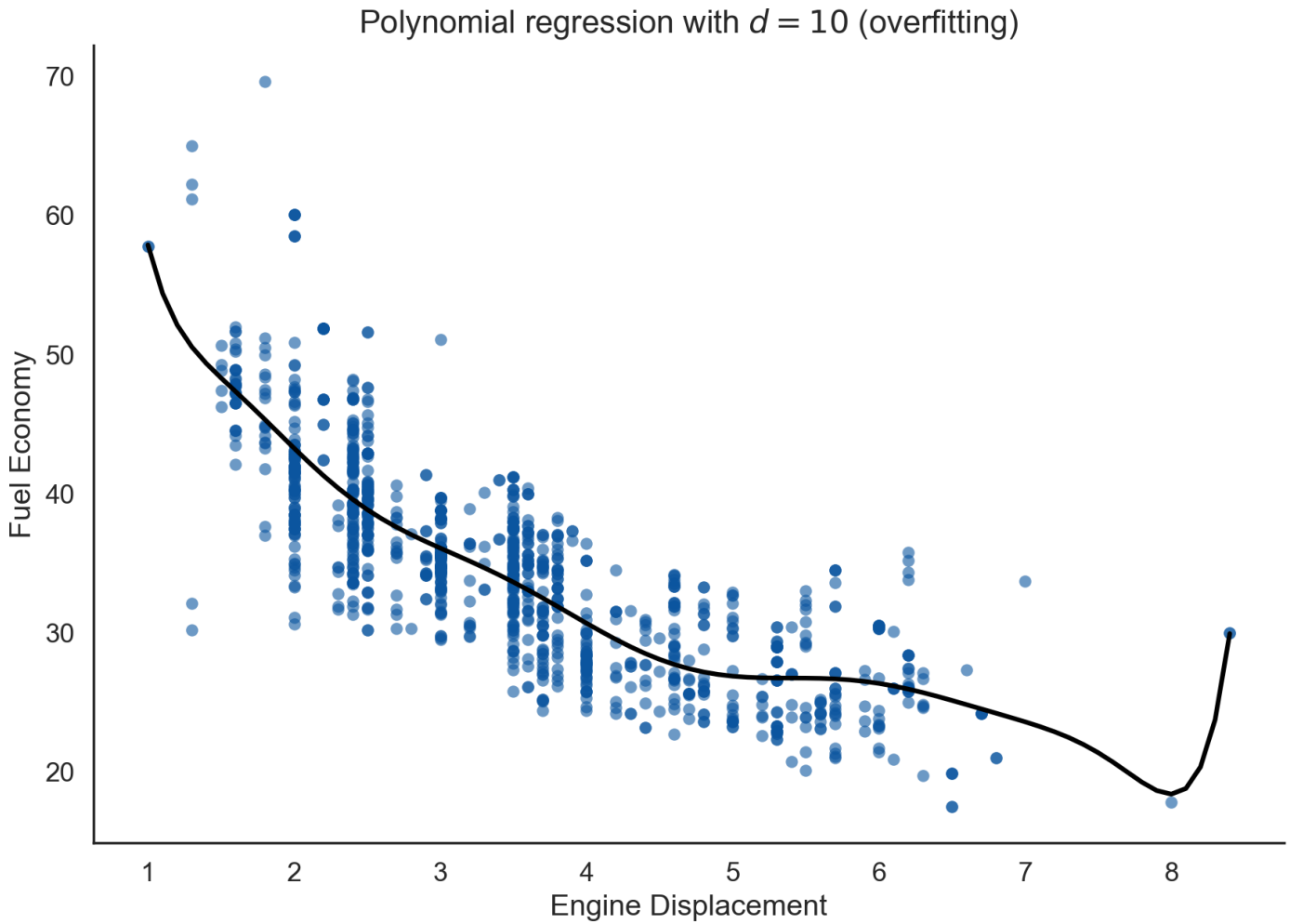
Limitations of polynomial regression (review)

- Polynomials can overfit. A polynomial of degree d can approximate d points exactly, so that increasing d produces a wiggly curve that gets close to the data, but predicts poorly.

- Polynomials display a highly **nonlocal fit**: observations in one region, especially outliers, can seriously affect the fit in another region.

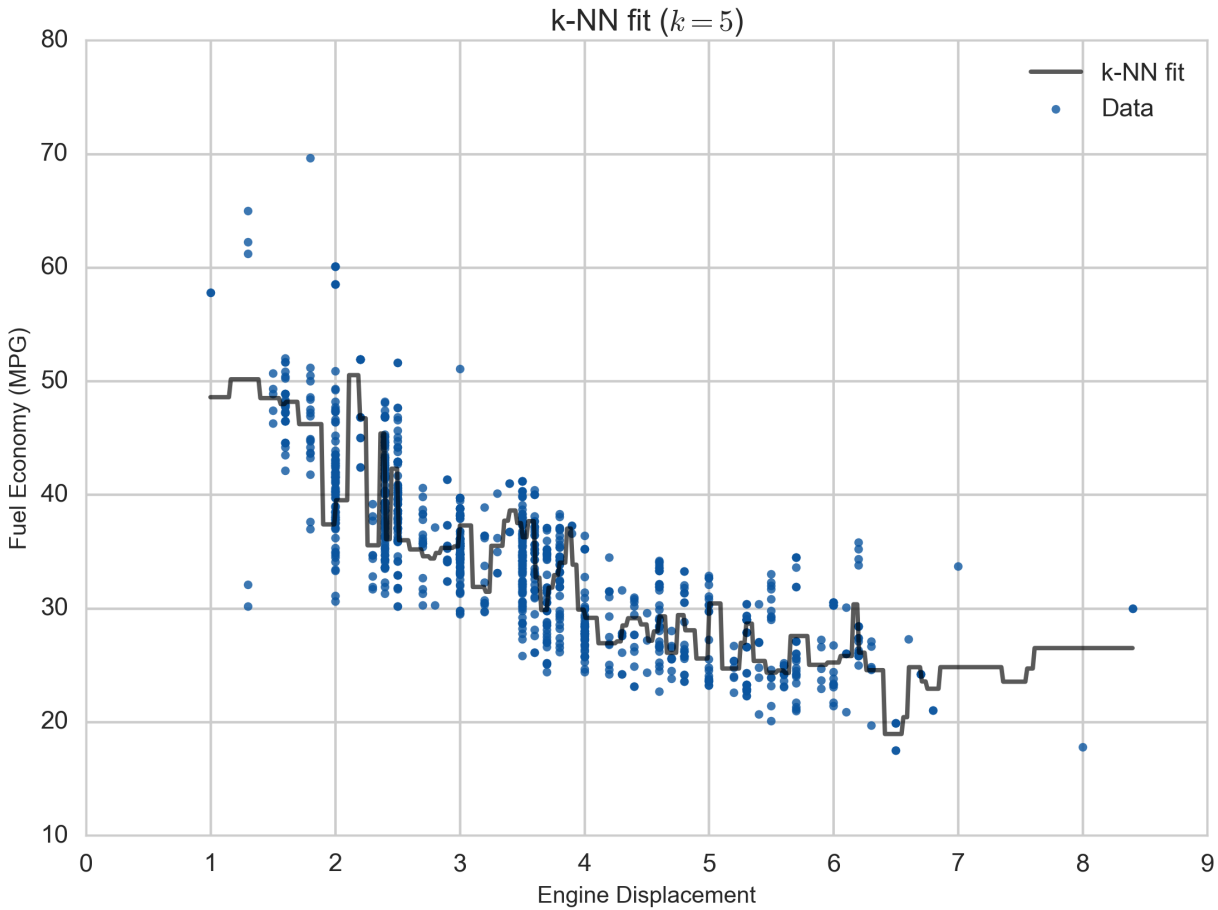
- Polynomials are unstable near the boundaries of the data. You should never extrapolate polynomial regressions to generate predictions outside the observed range of the predictor.

Example: polynomial overfitting



K-Nearest Neighbours

Another nonlinear model that we have seen is the KNN method.



The KNN method can lead to an unnecessarily "jumpy" (non-smooth) fit.

Overview of the methods to be discussed

Regression splines divide the range of X into different regions and fit polynomial regressions within each, with suitable smoothness restrictions when crossing between regions.

Smoothing splines are similar to regression splines, but arise from regularised empirical risk minimisation with a penalty that encourages smoothness.

Local regression To make a prediction at point x_0, fits a weighted linear regression in the neighbourhood x_0, where the weight of the training points decreases as they move further away from x_0.

Generalised additive models extend these methods to the case of multiple predictors.

Basis functions

Basis functions

The key idea of several nonlinear methods (including polynomial regression, regression splines, and smoothing splines) is to augment X with additional variables that are transformations of X, and then fit linear models with these derived features.

Basis functions

For $m = 1, \ldots, M$, let $h_m(x)$ be a transformation of x. We call h_m a **basis function** and model f as a linear combination of the basis functions:

$$f(x) = \beta_0 + \sum_{m=1}^{M} \beta_m h_m(x)$$

Since this specification is linear in the basis functions h_m, all the estimation and inference tools from linear regression immediately apply to this basis function model.

Basis functions: examples

$$f(x) = \beta_0 + \sum_{m=1}^{M} \beta_m h_m(x)$$

- $h_1(x) = x$ and $M = 1$ gives the original linear model.
- $h_1(x) = x$, $h_2(x) = x^2$, $h_3(x) = x^3$ and $M = 3$ leads to polynomial regression.
- $h_m(x) = \log(x)$, $h_m(x) = \sqrt{x}$, etc, permit other nonlinear transformations of the predictor.

Regression splines

Linear spline

A **linear spline** is a linear regression in which the slopes are allowed to change at certain fixed points called knots. If there is one knot at ξ, the model is:

$$f(x) = \beta_0 + \beta_1 x + \beta_2 (x - \xi)_+$$

where we define $(x - \xi)_+$ as:

$$(x - \xi)_+ = \begin{cases} 0 & \text{if } x - \xi \leq 0 \\ x - \xi & \text{if } x - \xi > 0 \end{cases}$$

Linear Splines

The model

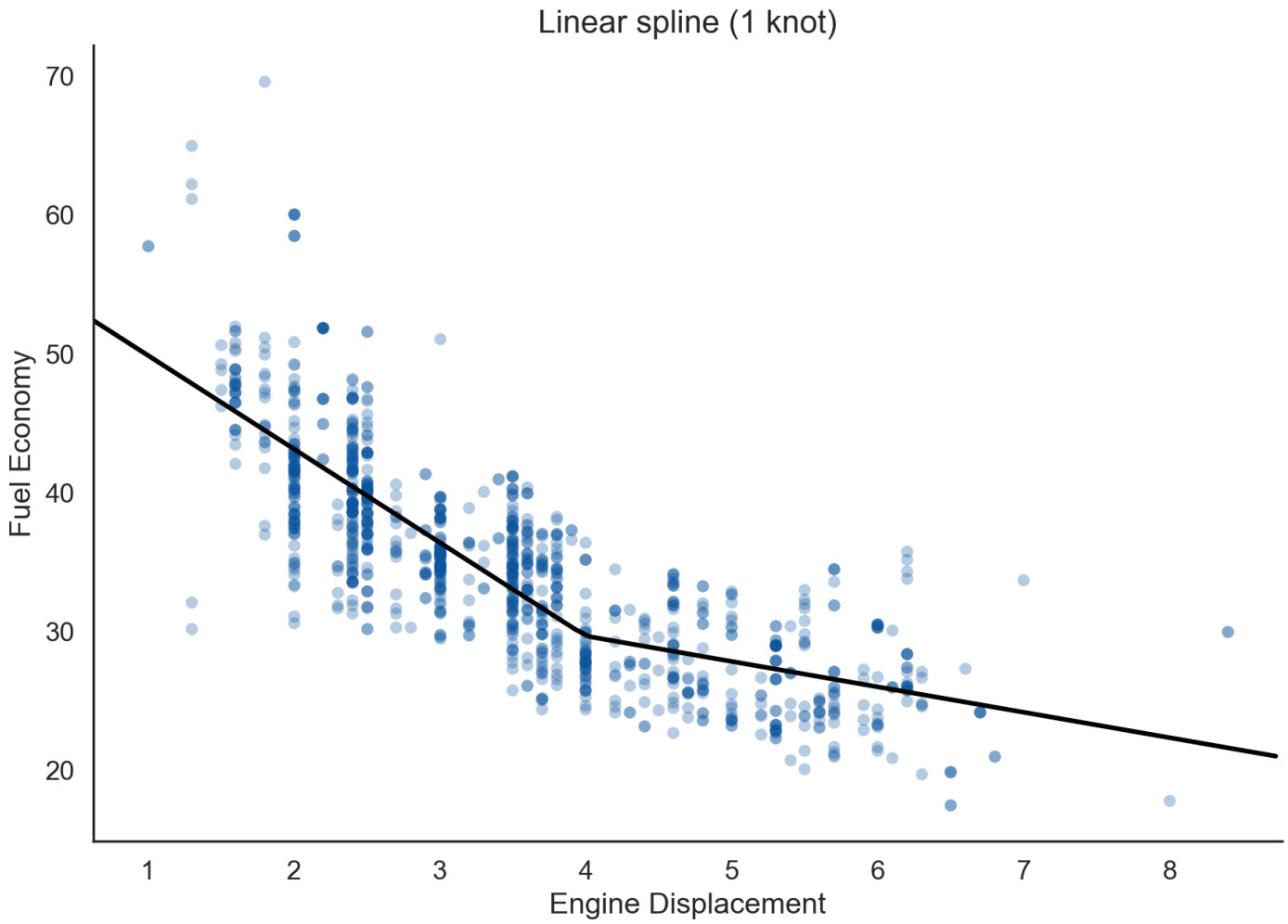
$$f(x) = \beta_0 + \beta_1 x + \beta_2 (x - \xi)_+$$

can be written as:

$$f(x) = \begin{cases} \beta_0 + \beta_1 x & \text{if } x \leq \xi \\ \beta_0 + \beta_1 x + \beta_2 (x - \xi) & \text{if } x > \xi \end{cases}$$

Note that at point $x = \xi$ the two linear components take the *same* value. Thus, the resulting function f is *continuous*.

Example: linear spline with one knot



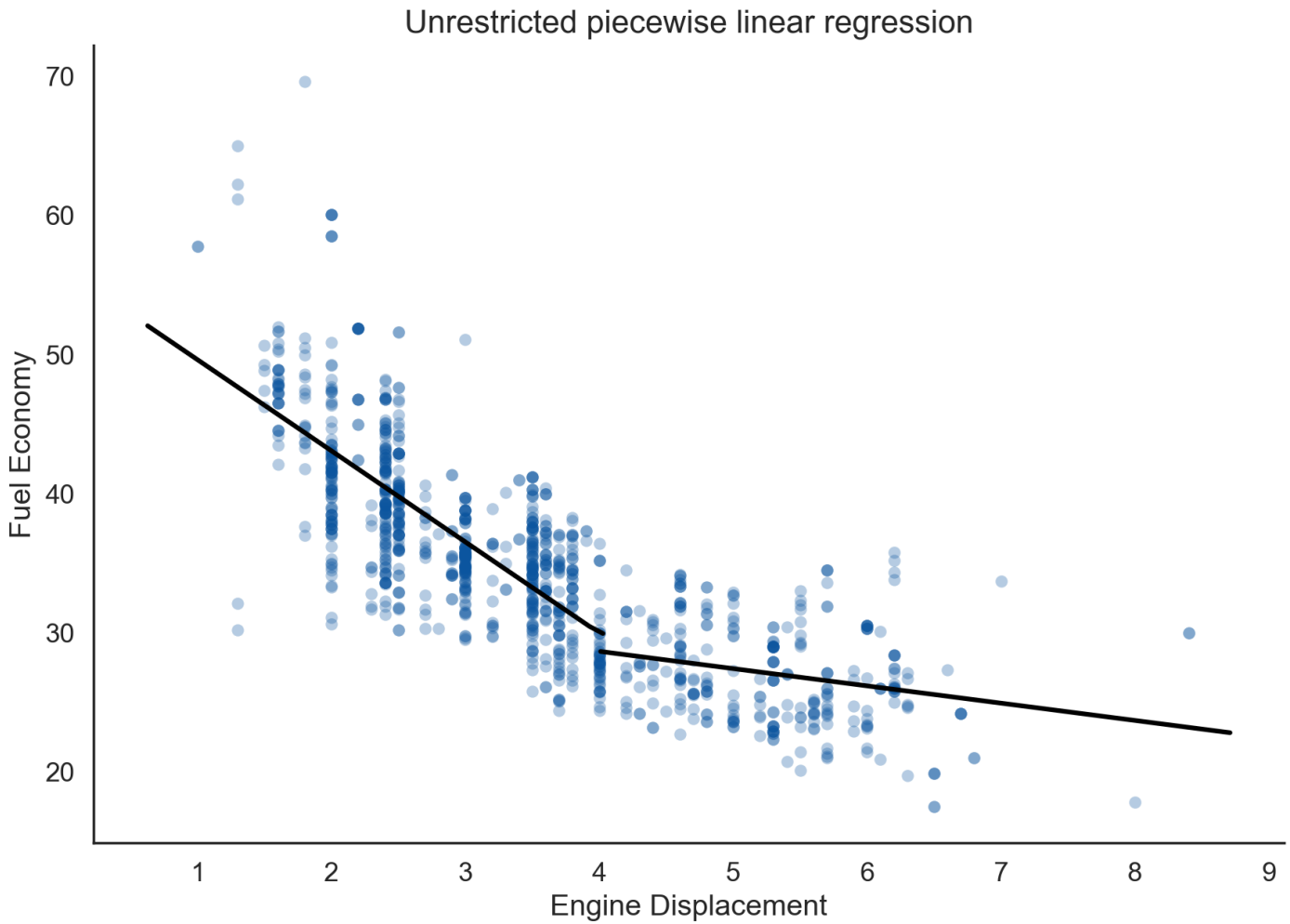
Piecewise linear regression

Linear spline is a special case of a **piecewise linear regression** model

$$f(x) = \begin{cases} \beta_{01} + \beta_{11}x & \text{if } x \leq \xi \\ \beta_{02} + \beta_{12}x & \text{if } x > \xi \end{cases}$$

However, a linear spline restricts the coefficients to ensure that the regression function is continuous.

Example: piecewise linear regression



Linear Splines (general definition)

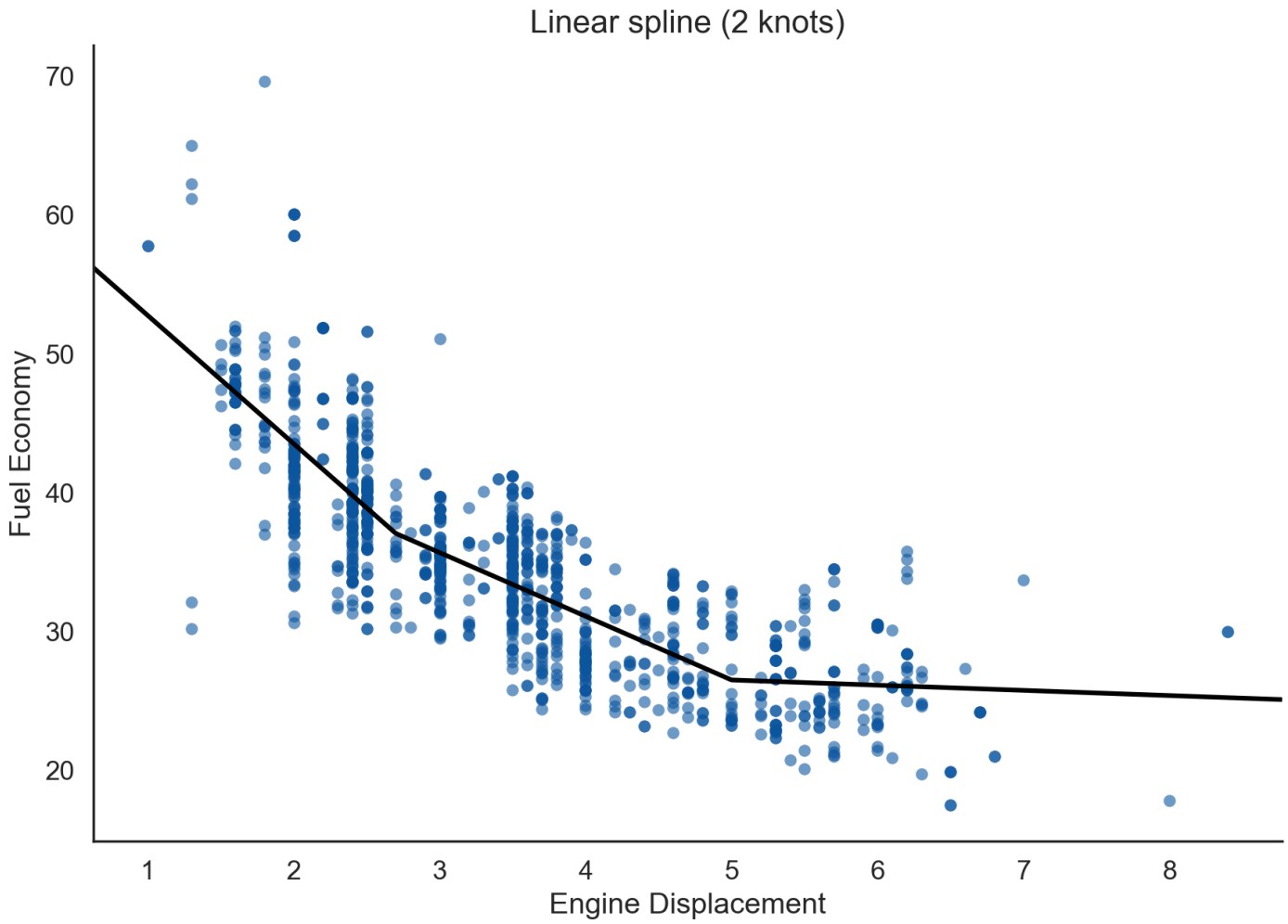
A linear spline model with K knots $\xi_1, \xi_2, \ldots, \xi_K$ is:

$$f(x) = \beta_0 + \beta_1 x + \beta_2 (x - \xi_1)_+ + \ldots + \beta_{K+1} (x - \xi_K)_+$$

Note that this representation uses $K + 1$ basis functions:

$$h_1(x) = x, \quad h_2(x) = (x - \xi_1)_+, \ \ldots \ , h_{K+1}(x) = (x - \xi_K)_+$$

Example: linear spline with two knots



Piecewise linear regressions

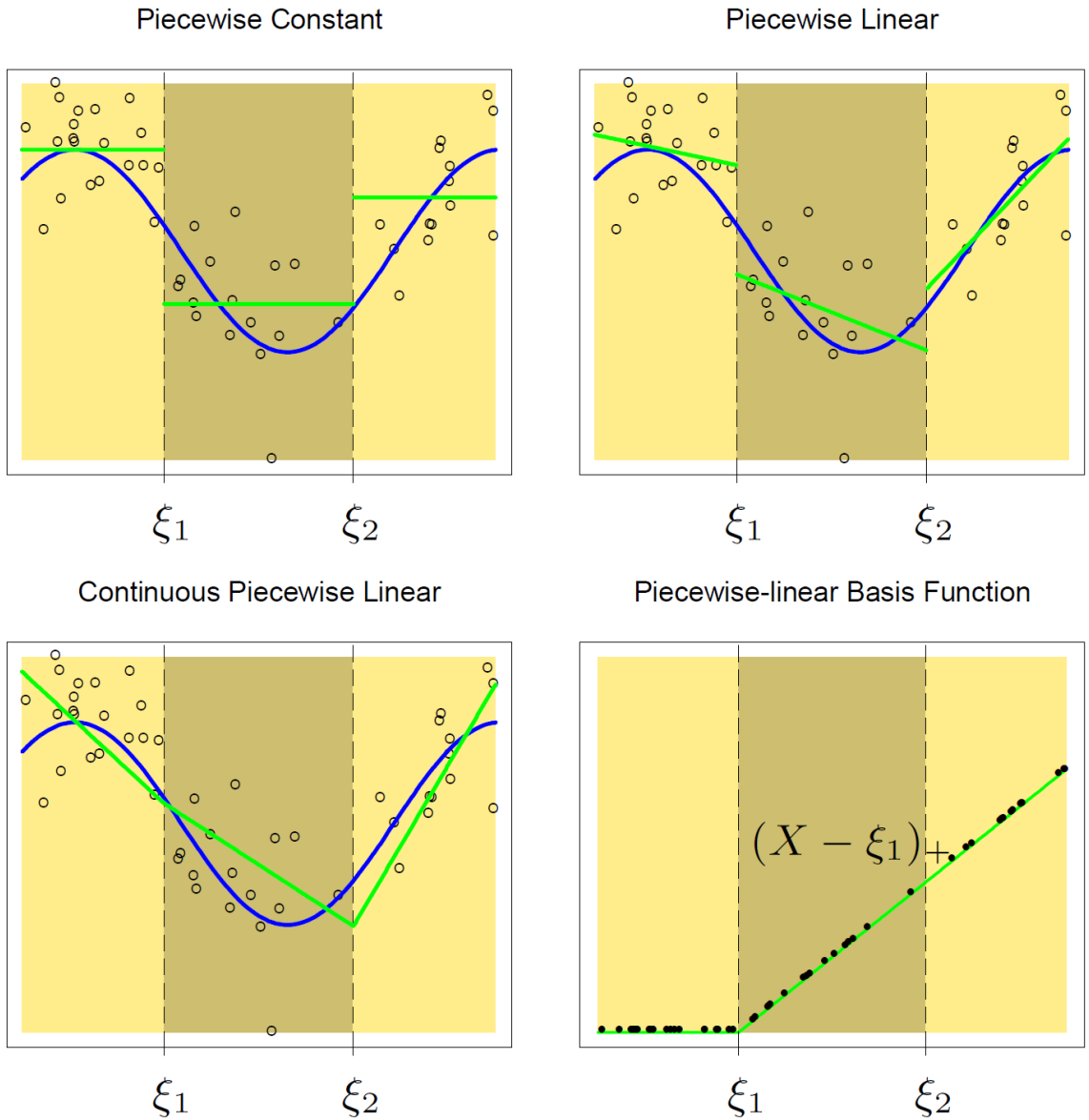


Figure from ESL

Regression splines

A **regression spline** is a piecewise degree-d polynomial regression that restricts the regression function $f(x)$ to be continuous and have continuous first $d-1$ derivatives.

This general approach extends the idea of linear splines by fitting polynomials instead of linear functions in each region.

Special cases: linear splines $(d = 1)$, cubic splines $(d = 3)$

Piecewise cubic polynomials

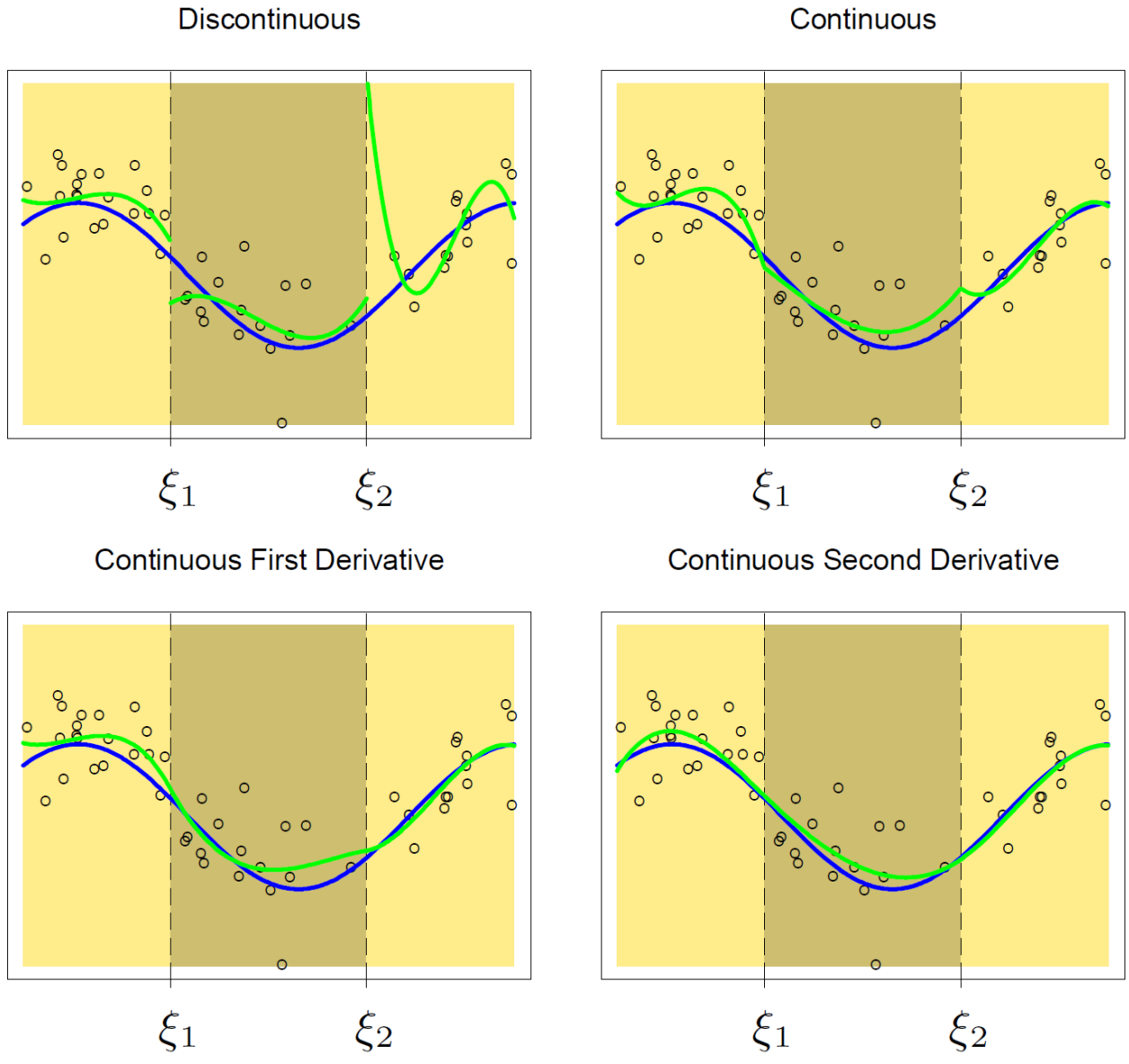


Figure from ESL

Cubic spline

A **cubic spline** model with K knots is

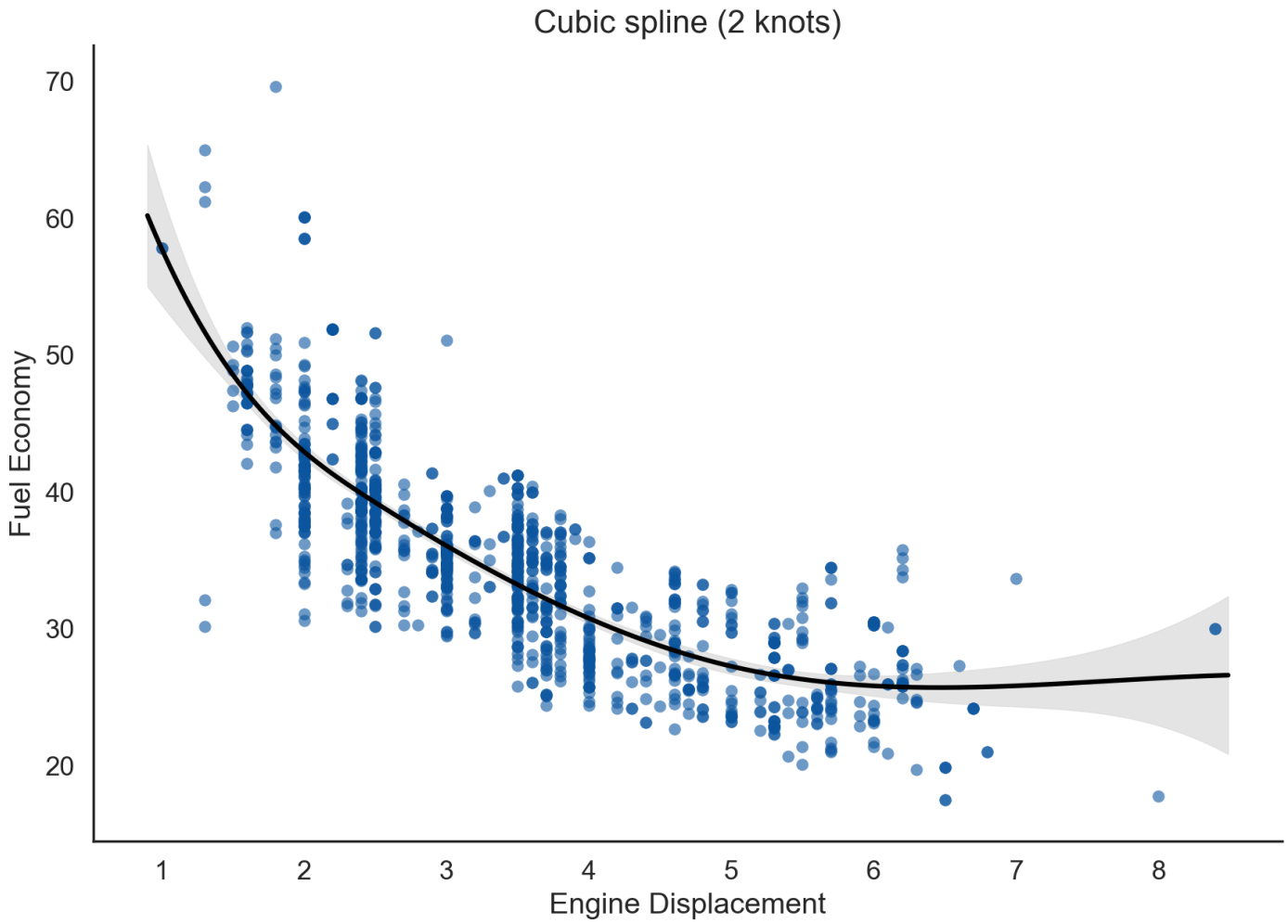
$$f(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \beta_4 (x - \xi_1)_+^3 \ldots + \beta_{K+3} (x - \xi_k)_+^3$$

The number of basis functions here is $K + 3$

For example, when there is $K = 1$ knot:

$$f(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \beta_4 (x - \xi_1)_+^3$$

Example: Cubic Spline



Regression splines vs polynomial regression

- Regression splines are preferable to polynomial regressions in most applications.

- The reason is that we can increase the flexibility of a spline by increasing the number of knots (without increasing the degree d of the polynomials). This leads to more stable estimates compared to increasing d, which is the only option for polynomial regressions.

Natural cubic splines

Natural cubic splines are a modification of cubic splines; they add the constraint that the regression function is linear in the two boundary regions (i.e to the left of the first knot and to the right of the last knot). This increases the stability of the fit.

Modelling choices

- Polynomial order: linear and, especially, cubic splines are the most common choices.

- Placement of the knots: typically at uniformly spaced percentiles of the data. For example, if there is one knot then we would place it at the sample median. If there are three, we would place them at the sample quantiles, i.e. the 25-th, the 50-th and the 75-th percentiles.

- Number of knots: model selection (e.g. cross-validation).

Degrees of freedom

It is useful to parameterise regression splines by their **degrees of freedom**, which equal the number of parameters that are free to vary, i.e. 1 plus the number of basis functions.

For example, a cubic spline with K knots uses $K + 4$ degrees of freedom (there are $K + 3$ basis functions).

We can also calculate the degrees of freedom for regression splines by taking the total number of parameters in the model, and then subtracting the number of linear constraints on these parameters.

For example, a natural cubic spline with K knots uses K degrees of freedom. This is because, when compared to cubic splines, natural splines impose two additional constraints for each of the boundary regions (coefficients for x^3 and x^2 must be zero), thus reducing the degrees of freedom by 4.

Smoothing splines

Smoothing splines

A **smoothing spline** solves the following regularised empirical risk minimisation problem:

$$\min_f \sum_{i=1}^{n} [y_i - f(x_i)]^2 + \lambda \int [f''(x)]^2 dx$$

where $\lambda \geq 0$ is a tuning parameter, $f''(x)$ is second derivative of the function, and $\int [f''(x)]^2 dx$ measures the overall "roughness" of the function. The penalty encourages the regression function to be smooth.

Special cases

- $\lambda = 0$: the solution f can be any function that interpolates (passes through) the data points.
- $\lambda = \infty$: the simple least squares line fit, since no second derivative can be tolerated ($f''(x) = 0$ for all x, hence the model is linear).
- $\lambda \in (0, \infty)$ allows functions that vary from very rough to very smooth.

Smoothing spline

- Remarkably, when $\lambda > 0$, the unique solution is a natural cubic spline with knots at $x_1, x_2, \ldots, x_n$.

- The penalty term becomes a penalty on the spline coefficients, which get shrunk towards the linear fit.

- Similarly to OLS and ridge regression, there is a short-cut for computing the LOOCV MSE, in which you only need to fit the model once, on all the data.

Effective Degrees of freedom

Effective degrees of freedom (or effective number of parameters) of a regression model are defined as:

$$\text{df}(\widehat{Y}) = \frac{\sum_{i=1}^{n} \text{Cov}(\widehat{Y}_i, Y_i)}{\sigma^2}$$

For regression splines, $\text{df}(\widehat{Y})$ just gives the degrees of freedom discussed earlier (we will use this fact without proof). The following holds:

for OLS: $\text{df}(\widehat{Y}) = p + 1$

for K-NN regression: $\text{df}(\widehat{Y}) = n/K$

for smoothing splines, as λ increases from zero (perfect fit) to infinity (linear fit), $\text{df}(\widehat{Y})$ decreases from n to 2.

Local regression

Local regression

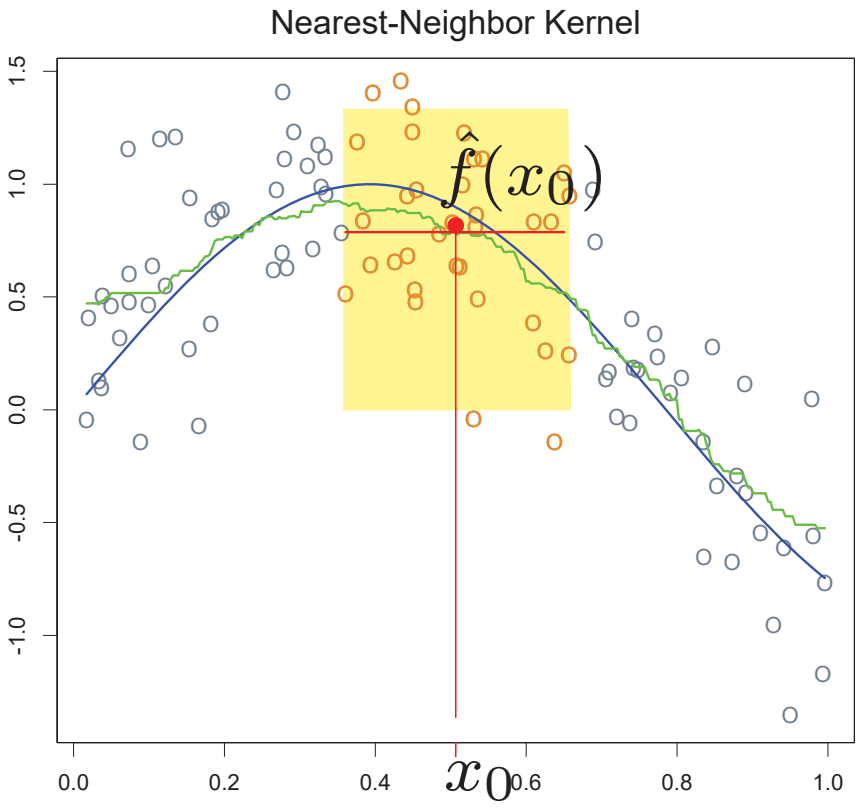
To make a prediction at a point x_0, **local regression** or **kernel smoothing** methods fit a regression model using only nearby observations, where the training cases are assigned weights that die off smoothly as the distance from x_0 increases.

Like the KNN method, local regression uses all the training data every time it needs to compute a prediction.

Local average (local regression is without the predictor, just with the intercept)

Left: KNN (average neighbours' y values) Right: weighted average

Yellow curve shows weights (Left: all equal Right: decreasing with distance to x_0)



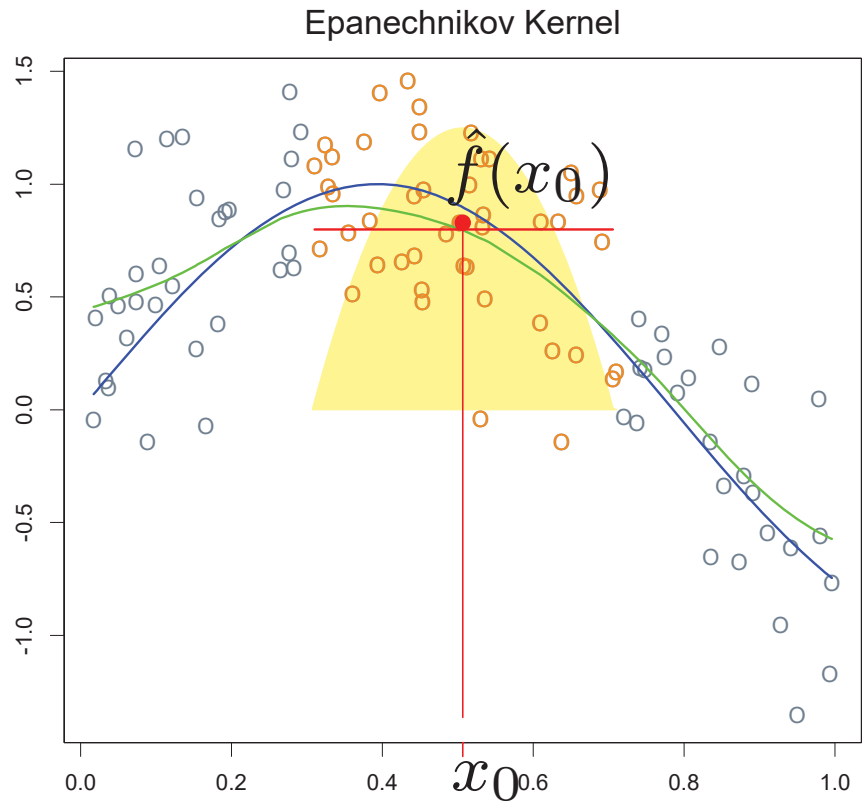


Figure from ESL

Local linear regression

Here instead of averaging we fit a weighted linear regression using just the neighbours

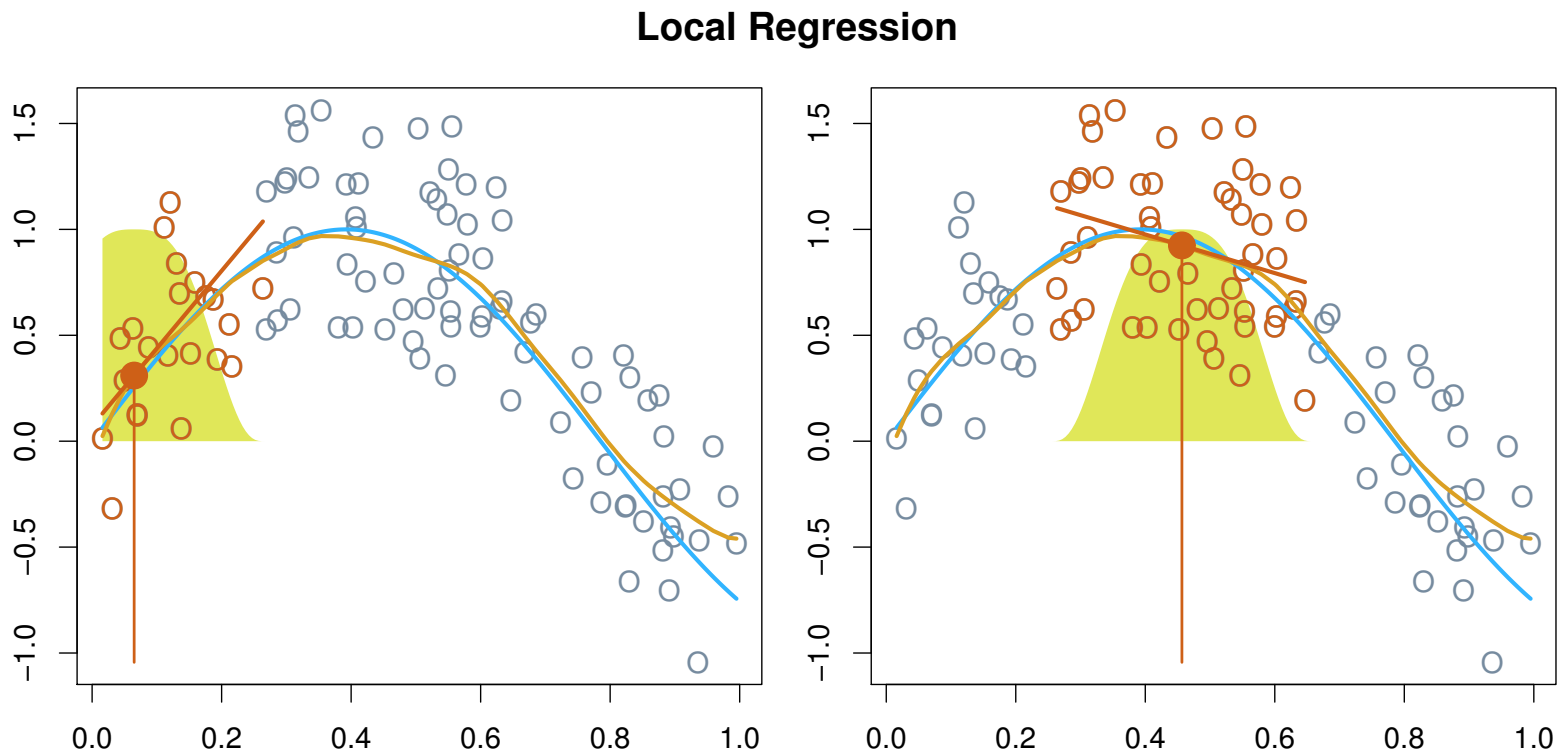


Figure from ISL

Local linear regression

Algorithm Local linear regression at point x_0

1: Take a neighbourhood of x_0 with the width determined by the tuning parameter λ.

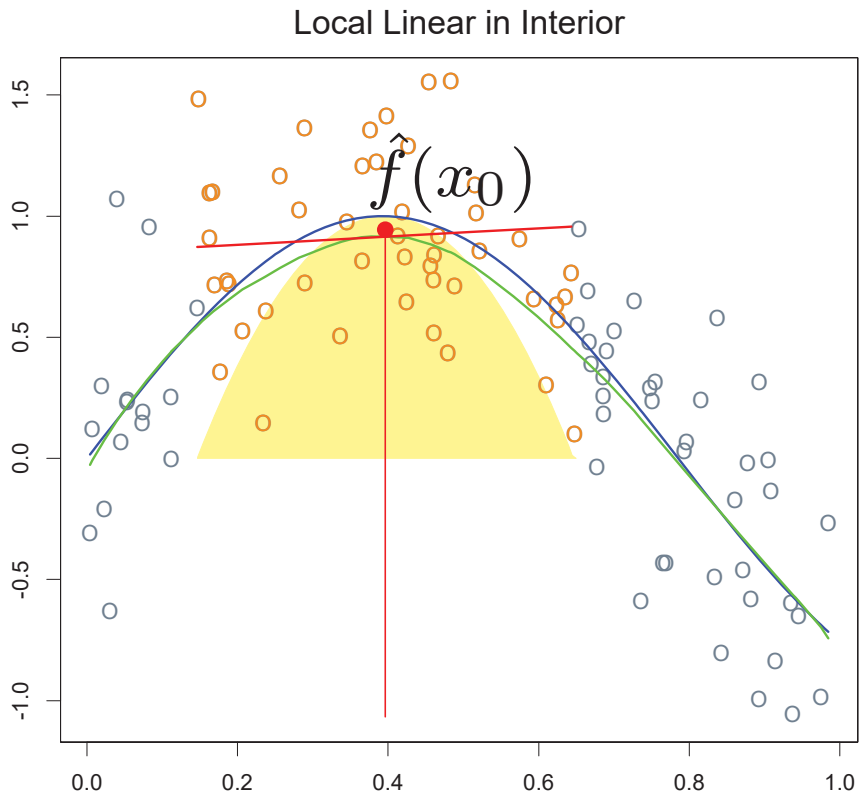
2: Assign a weight $K_\lambda(x, x_0)$ to each point x in the neighbourhood, with the highest weight given to the point x closest to x_0. As x moves towards the boundary of the neighbourhood, the weight smoothly decreases all the way to zero.

3: Fit the weighted least squares regression

$$\widehat{\beta}_0, \widehat{\beta}_1 = \underset{\beta_0, \beta_1}{\text{argmin}} \sum_{i=1}^{n} K_\lambda(x_i, x_0)\Big(y_i - \beta_0 - \beta_1 x_i\Big)^2$$

4: The prediction is $\widehat{f}(x_0) = \widehat{\beta}_0 + \widehat{\beta}_1 x_0$.

Can also fit a local quadratic regression



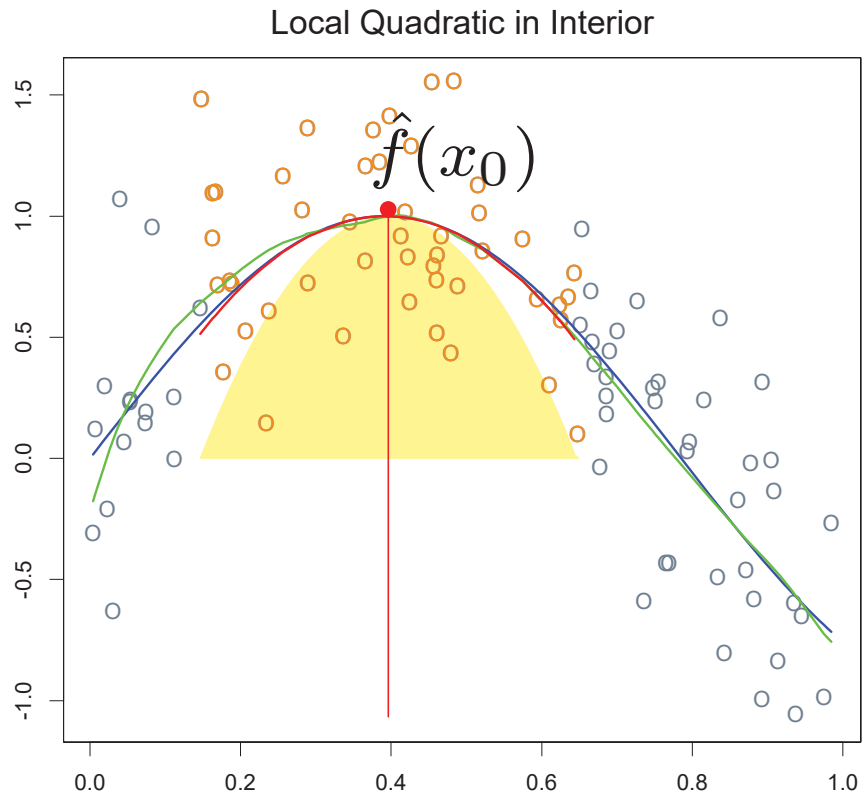


Figure from ESL

Modelling choices

1. The **kernel function** (weight function) K_λ.

2. Whether to fit an average, linear regression, or polynomial regression in the neighbourhood.

3. The tuning parameter λ, which controls how local the method is.

Generalised additive models

Generalised additive model

The **generalised additive model** (GAM) is

$$Y = f_1(X_1) + f_2(X_2) + \ldots + f_p(X_p) + \varepsilon$$

where functions f_j are generally some smooth nonlinear functions of each individual predictor.

Methods from earlier in this lecture can be used as building blocks for fitting GAMs.

Generalised additive models: advantages

- GAMs allow us to fit *nonlinear* functions $f_j(X_j)$ for each predictor, so that we can automatically model nonlinear relationships missed by linear regression. The nonlinear fits can potentially lead to higher predictive accuracy.

- GAMs leverage the advantages of its building blocks into a convenient framework for multiple predictors.

- Because of the additive structure of the model, it is still easy for us to interpret the relationship between each X_j and Y conditional on the other predictors.

Generalised additive models: limitations

An important limitation of GAMs is that they do **not** account for interactions. Interactions arise when we consider general multivariate functions such as $f(X_1, X_2)$.

One option in this case is to use a bivariate polynomial regression as an approximation to $f(X_1, X_2)$; a more flexible approach is to use 2-dimensional splines or local regression.

Interactions

- We could go even further and consider functions of many variables $f(X_1, \ldots, X_p)$, but doing so leads to the **curse of dimensionality**.

- In this setting, the curse of dimensionality refers to problem that the number of parameters in flexible approximations to $f(X_1, \ldots, X_p)$ grows exponentially in p.

Review questions

- What are the main disadvantages of polynomial regression?
- What are regression splines and how do we calculate their degrees of freedom?
- What is a smoothing spline and how is the fit affected by the tuning parameter λ?
- What is a local regression and how is it different from KNN?
- What is a GAM and what are its limitations?