BIG DATA ANALYTICS
Moving Beyond Linearity

Olga Klopp
klopp@essec.edu

Step Functions

Regression Splines

Smoothing Splines

Local Regression

Generalized Additive Models

Linear models

- Linear models:
  - relatively simple to describe and implement
  - have advantages in terms of interpretation and inference
  - significant limitations in terms of predictive power:
    - the linearity assumption is almost always an approximation (sometimes a poor one)
- Relax the linearity assumption while still maintaining as much interpretability as possible:
  - polynomial regression
  - step functions
  - splines
  - local regression
  - generalized additive models

Extensions

- *Polynomial regression*:
  - extends the linear model by adding extra predictors X^2, X^3, ...
  - a simple way to provide a nonlinear fit to data
- *Step functions*:
  - cuts the range of a variable into K distinct regions
  - fits a piecewise constant function

Extensions

- *Regression splines*:
  - more flexible than polynomials and step functions
  - an extension of both
  - divides the range of X into K distinct regions
  - within each region, a polynomial function is fitted to the data
  - polynomials are constrained: they join smoothly at the region boundaries
  - can produce an extremely flexible fit

Extensions

- *Smoothing splines*:
  - similar to regression splines
  - result from minimizing a residual sum of squares criterion subject to a smoothness penalty
- *Local regression*:
  - similar to splines
  - the regions are allowed to overlap
- *Generalized additive models*: allow to extend the methods above to deal with multiple predictors.

Outline

Step Functions

Regression Splines

Smoothing Splines

Local Regression

Generalized Additive Models

Outline

Step Functions

Regression Splines

Smoothing Splines

Local Regression

Generalized Additive Models

Step Functions

- Breaks the range of X into bins, and fits a different constant in each bin:
  - converting a continuous variable into an ordered categorical variable
  - we create cutpoints $c_1, c_2, ..., c_K$ in the range of X
  - construct $K + 1$ new variables:

$$
\begin{aligned}
C_0(X) &= I(X < c_1) \\
C_1(X) &= I(c_1 \leq X < c_2) \\
&\dots \\
C_K(X) &= I(c_K \leq X)
\end{aligned}
$$

  where $I(\cdot)$ is an indicator function

Step Functions

- $I(\cdot)$ returns a 1 if the condition is true, and returns a 0 otherwise
- For example, $I(c_K \leq X)$ equals 1 if $c_K \leq X$ and equals 0 otherwise
- Also called dummy variables
- For any value of X

$$C_0(X) + C_1(X) + ... + C_K(X) = 1$$

since X must be in exactly one of the $K + 1$ intervals

Step Functions

- We use least squares to fit a linear model using $C_1(X), C_2(X), ..., C_K(X)$ as predictors:

$$y_i = \beta_0 + \beta_1 C_1(x_i) + \beta_2 C_2(x_i) + \cdots + \beta_K C_K(x_i) + \epsilon_i$$

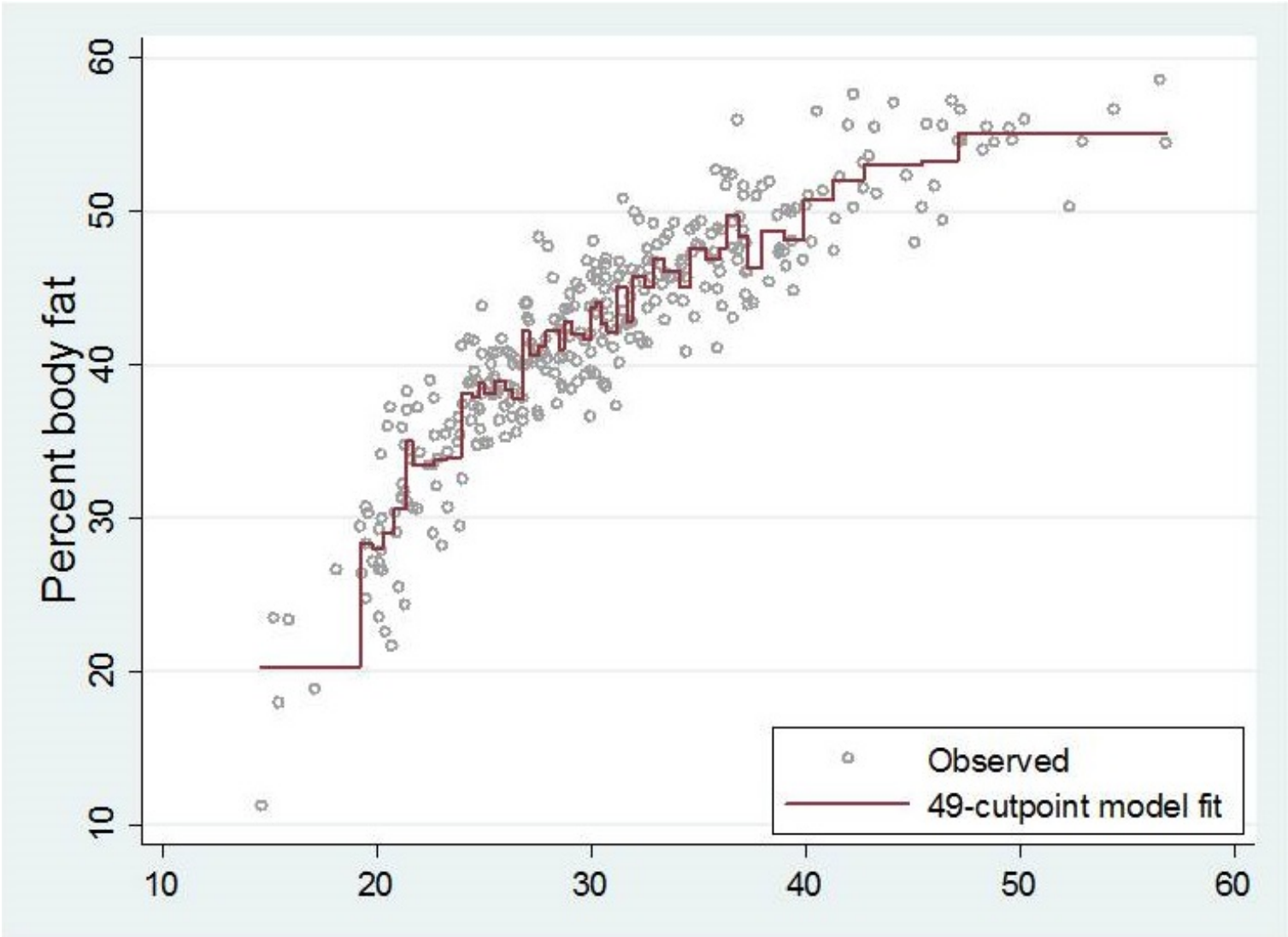
- For a given value of X, at most one of $C_1, C_2, ..., C_K$ can be non-zero
- When $X < c_1$, all of the predictors are zero $\implies$ β_0 is the mean value of Y for $X < c_1$
- For $c_j \leq X < c_{(j+1)}$ we get $\beta_0 + \beta_j \implies \beta_j$ is the average increase in the response for X in $c_j \leq X < c_{(j+1)}$ relative to $X < c_1$

Step Functions

$$y_i = \beta_0 + \beta_1 C_1(x_i) + \beta_2 C_2(x_i) + \cdots + \beta_K C_K(x_i) + \epsilon_i$$

- We exclude $C_0(X)$ as a predictor because it is redundant with the intercept
- The decision to exclude $C_0(X)$ instead of some other $C_k(X)$ is arbitrary
- Alternatively, we could include $C_0(X), C_1(X), \ldots, C_K(X)$ and exclude the intercept.

Example



The relationship between percentage body fat and body mass index: $N = 326$, number of cut points 50

Step Functions

- Unless there are natural breakpoints in the predictors, piecewise-constant functions can miss the action
- Nevertheless, very popular in biostatistics, epidemiology ...

Basis Functions

- Polynomial and piecewise-constant regression models are special cases of a *basis function approach*
- We have at hand a family of functions or transformations that can be applied to a variable $X : b_1(X), b_2(X), \ldots, b_K(X)$
- We fit the model:

$$y_i = \beta_0 + \beta_1 b_1(x_i) + \beta_2 b_2(x_i) + \cdots + \beta_K b_K(x_i) + \epsilon_i$$

Basis Functions

- The basis functions $b_1(), b_2(), \ldots, b_K()$ are fixed and known
- For polynomial regression: $b_j(x_i) = x_i^j$
- For piecewise constant functions: $b_j(x_i) = I(c_j \leq x_i < c_{(j+1)})$
- A standard linear model with predictors $b_1(x_i), b_2(x_i), ..., b_K(x_i)$:
  - we can use, e.g., least squares to estimate the unknown regression coefficients
  - all the tools for linear models (e.g., standard errors and F-statistics for the model's overall significance) are available in this setting
- Other choices for basis functions: wavelets, Fourier series or **regression splines**

Outline

Step Functions

Regression Splines

Smoothing Splines

Local Regression

Generalized Additive Models

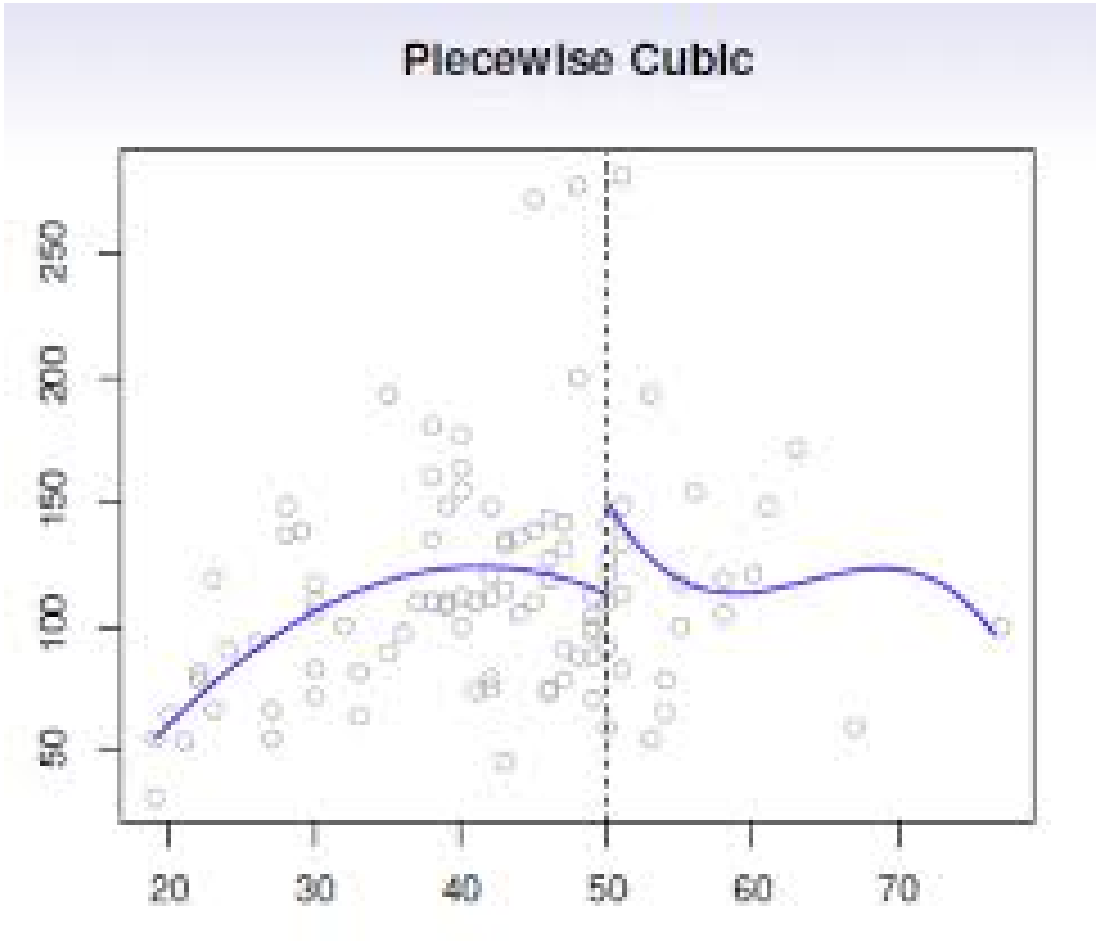
Piecewise Polynomials

- Piecewise polynomial regression involves fitting separate low-degree polynomials over different regions of X
- The points where the coefficients change are called *knots*
- Example, a piecewise cubic polynomial with a single knot at a point c:

$$y_i = \begin{cases} \beta_{01} + \beta_{11} x_i + \beta_{21} x_i^2 + \beta_{31} x_i^3 + \epsilon_i & \text{if } x_i < c \\ \beta_{02} + \beta_{12} x_i + \beta_{22} x_i^2 + \beta_{32} x_i^3 + \epsilon_i & \text{if } x_i \geq c \end{cases}$$

  - we fit two different polynomial functions to the data:
    - one on the subset of the observations with $x_i < c$
    - one on the subset of the observations with $x_i \geq c$
- Using more knots leads to a more flexible piecewise polynomial

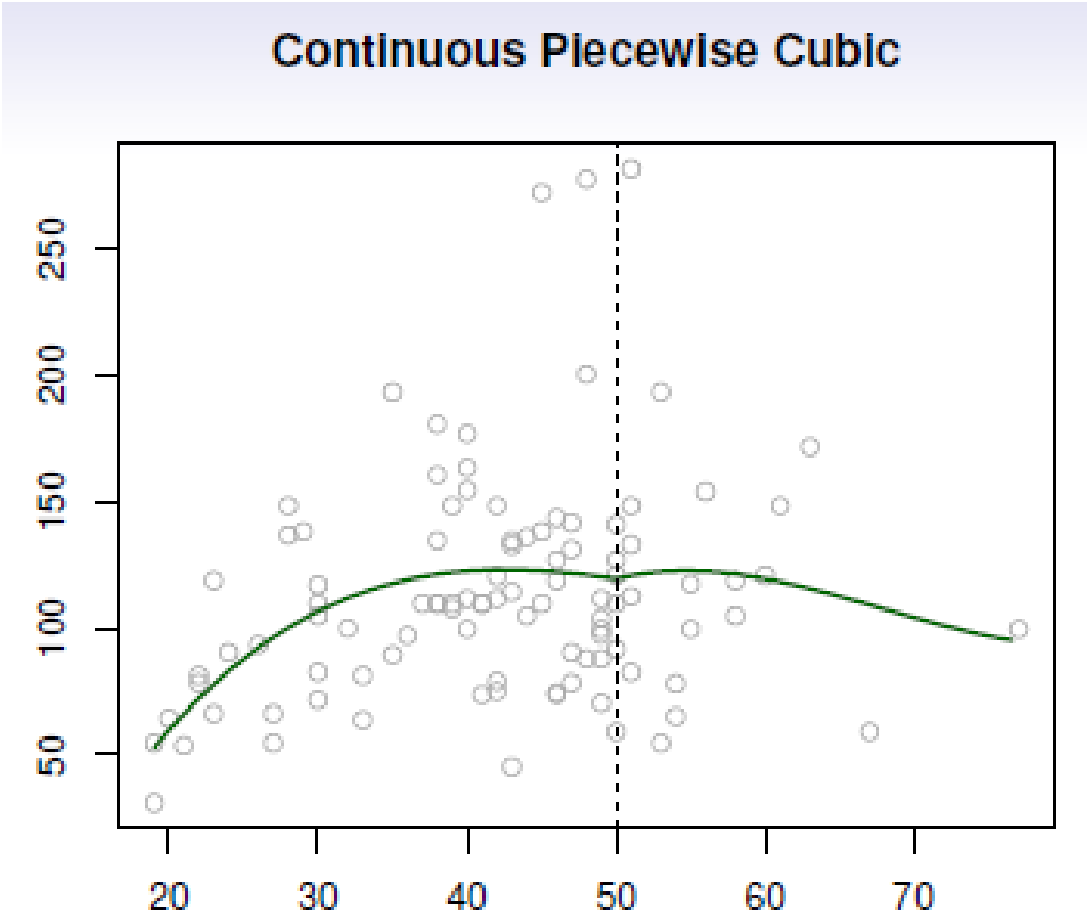
Piecewise Polynomials: example



The function is discontinuous

Constraints and Splines

- Fit a piecewise polynomial under the constraint that the fitted curve must be continuous:

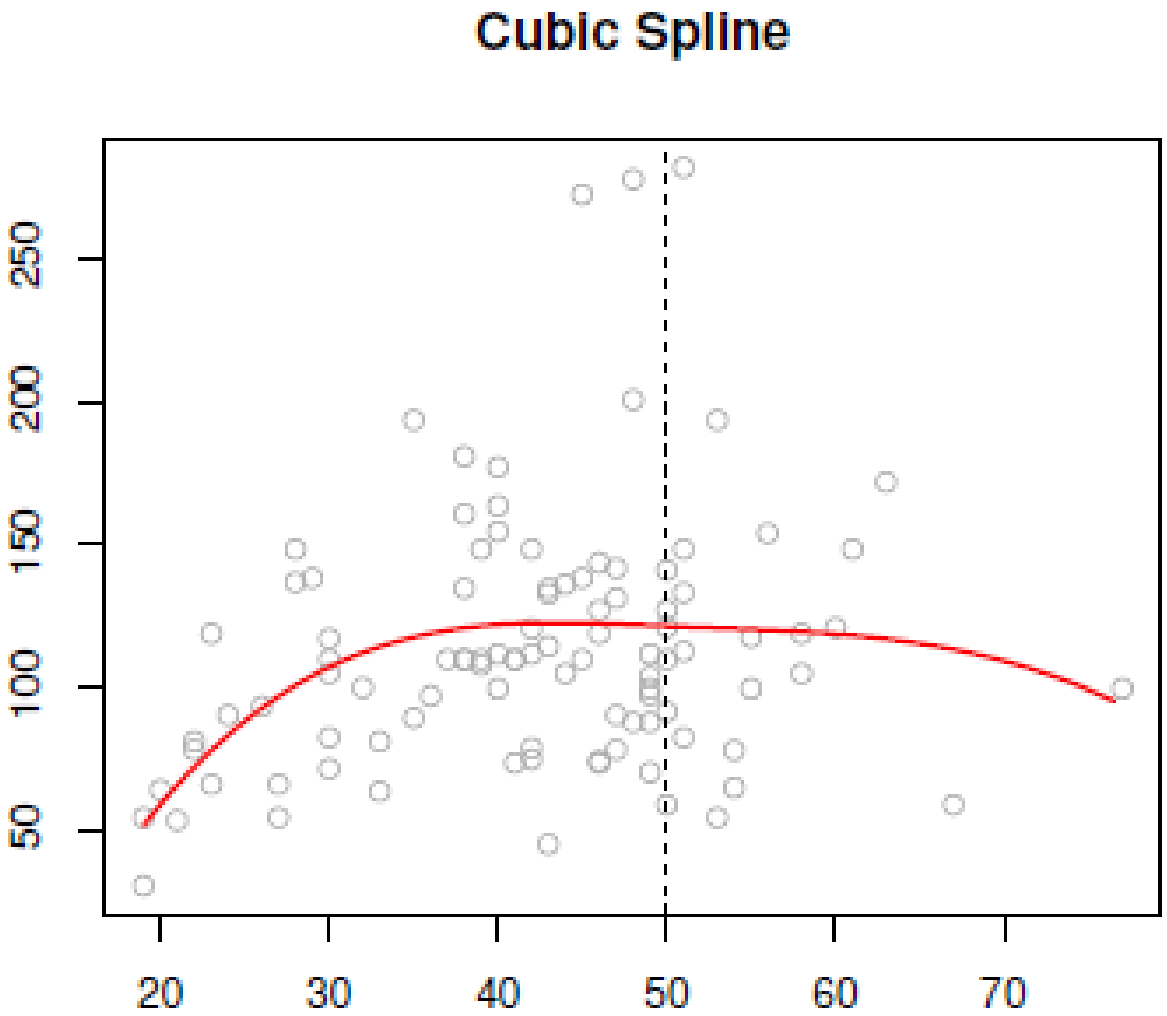


Better, but the V-shaped join looks unnatural

Constraints and Splines

- Additional constraints: both the first and second derivatives of the piecewise polynomials are continuous
- We are requiring that the piecewise polynomial be not only continuous, but also very smooth $\Longrightarrow$ a smoother form of the curve in the join points

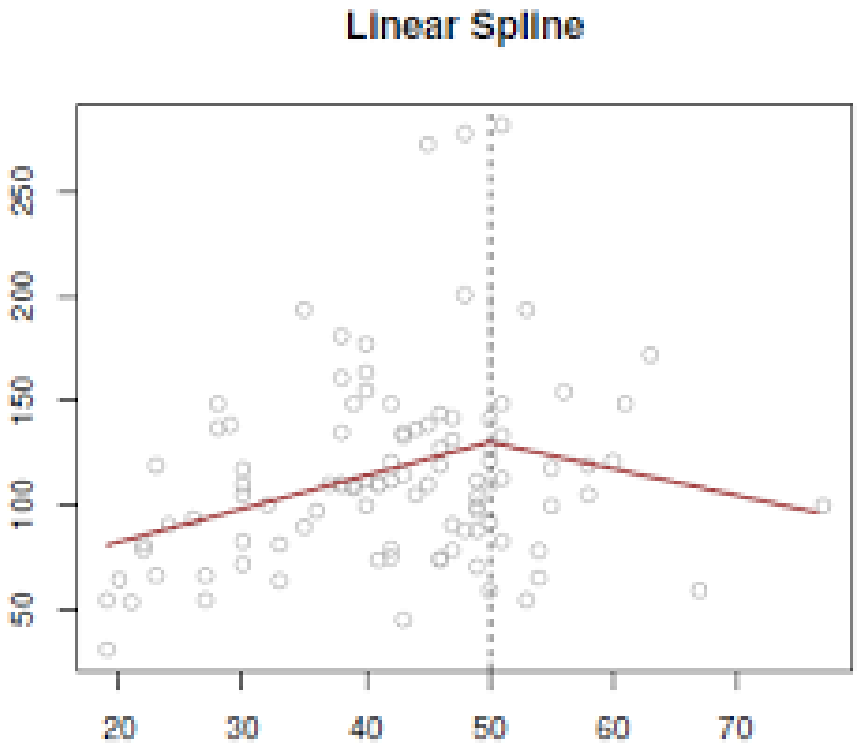
Splines



Splines

A degree-d spline is a piecewise degree-d polynomial, with continuity in derivatives up to degree $d-1$ at each knot

- Example: a linear spline is obtained by fitting a line in each region of the predictor space defined by the knots, requiring continuity at each knot



The Spline Basis Representation

- We can use the basis model to represent a regression spline
- E .g., a cubic spline with K knots can be modeled as

$$y_i = \beta_0 + \beta_1 b_1(x_i) + \beta_2 b_2(x_i) + \cdots + \beta_{K+3} b_{K+3}(x_i) + \epsilon_i$$

for an appropriate choice of basis functions $b_1, b_2, ..., b_{(K+3)}$

- This model can then be fitted using least squares

The Spline Basis Representation

- There are many equivalent ways to represent cubic splines using different choices of basis functions
- The most direct way:
  - start with a basis for a cubic polynomial: x, x^2, x^3
  - add one truncated power basis function per knot:
    - a truncated power basis function:

$$\eta(x,\xi) = (x-\xi)_+^3 = \begin{cases} (x-\xi)^3 & \text{if } x > \xi \\ 0 & \text{otherwise} \end{cases}$$

where ξ is a knot

- There is rarely any need to go beyond cubic splines, which are the most common type of splines in practice.

The Spline Basis Representation

- To fit a cubic spline to a data set with K knots, we perform least squares regression with an intercept and $3 + K$ predictors, of the form

$$X, X^2, X^3, h(X, \xi_1), h(X, \xi_2), ..., h(X, \xi_K)$$

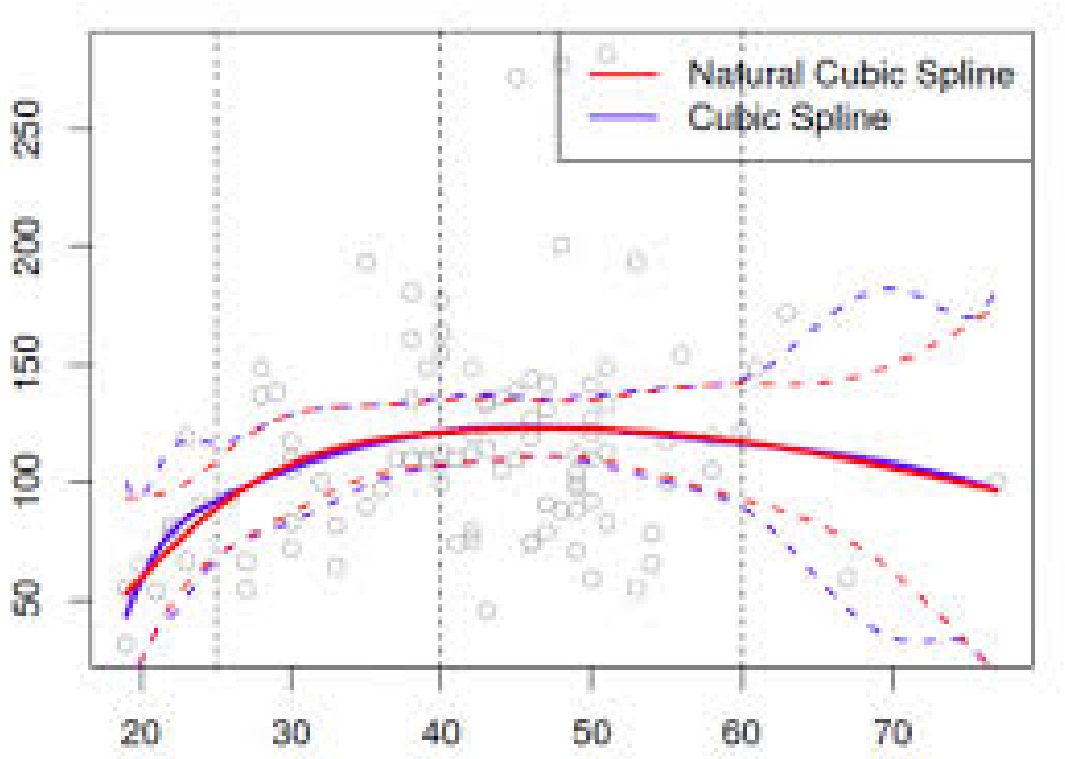
  where $\xi_1, ..., \xi_K$ are the knots

- This amounts to estimating a total of $K + 4$ regression coefficients
- Fitting a cubic spline with K knots uses $K + 4$ degrees of freedom.

Natural Spline

- Splines can have high variance close to the boundary
  - Boundary: the region where X is smaller than the smallest knot, or larger than the largest knot
- *A natural spline* is a regression spline with additional boundary constraints:
  - the function is required to be linear near the boundary
- Natural splines generally produce more stable estimates near the boundaries

Natural Spline



- In blue, a cubic spline: confidence bands in the boundary region appear fairly wild
- In red, a natural cubic spline: the corresponding confidence intervals are narrower

Choosing the Locations of the Knots

- Where should we place the knots?
- The regression spline is most flexible in regions that contain a lot of knots
  - the polynomial coefficients can change rapidly
- One option: to place more knots in places where the function might vary most rapidly, and to place fewer knots where it seems more stable
- In practice, it is common to place knots in a uniform fashion:
  - we specify the desired degrees of freedom
  - the software automatically place the corresponding number of knots at uniform quantiles of the data

Choosing the Number of the Knots

- How many knots should we use (equivalently how many degrees of freedom)?
- One option: try out different numbers of knots and see which produces the best looking curve

Choosing the Number of the Knots

- More objective approach: to use cross-validation:
  - we remove a portion of the data (say 10 %)
  - fit a spline with a certain number of knots to the remaining data
  - use the spline to make predictions for the held-out portion
  - repeat this process multiple times until each observation has been left out once
  - compute the overall cross-validated RSS
  - this procedure can be repeated for different numbers of knots K
  - the value of K giving the smallest RSS is chosen.

Comparison to Polynomial Regression

- Regression splines often give superior results to polynomial regression:
  - polynomials must use a high degree to produce flexible fits
  - splines introduce flexibility by increasing the number of knots but keeping the degree fixed
  - generally, this approach produces more stable estimates
- Splines allow to place more knots, and hence flexibility, over regions where the function f seems to be changing rapidly, and fewer knots where f appears more stable.

Outline

Step Functions

Regression Splines

Smoothing Splines

Local Regression

Generalized Additive Models

Smoothing Splines

- Regression splines:
  - specifying a set of knots
  - producing a sequence of basis functions
  - using least squares to estimate the spline coefficients
- A different approach that also produces a spline:

Smoothing Splines

Smoothing Splines

- Fitting a smooth curve to a set of data:
  - we want to find some function, g, that fits the observed data well $\Longleftrightarrow$
  - find function g such that

$$RSS = \sum (y_i - g(x_i))^2 \quad \text{is small}$$

- If we don't put any constraints on g, we can always make RSS zero:
  - by choosing g such that it interpolates all of the y_i
- Such function would overfit the data - it would be far too flexible
- **A function g that makes RSS small, but that is also smooth**

Smoothing Splines

- Find the function g that minimizes

$$\sum_{i=1}^{n} (y_i - g(x_i))^2 + \lambda \int (g''(t))^2 \, \mathrm{d}t$$

  where λ is a nonnegative tuning parameter

- The function g that minimizes this criteria is known as *a smoothing spline*
- It takes the "Loss+Penalty" formulation (as the ridge regression and the lasso)
- The term $\sum_{i=1}^{n} (y_i - g(x_i))^2$ is a loss function that encourages g to fit the data well
- The term $\lambda \int (g''(t))^2 \, \mathrm{d}t$ is a penalty term that penalizes the variability in g.

Smoothing Splines: penalty

$$\lambda \int (g''(t))^2 \, \mathrm{d}t$$

- $g''(t)$ is the second derivative of the function g
- The first derivative $g'(t)$ measures the slope of a function at t
- The second derivative corresponds to the speed at which the slope is changing
- It is a measure of its roughness:
  - it is large in absolute value if g is very wiggly near t
  - it is close to zero otherwise
  - the second derivative of a straight line is zero

Smoothing Splines: penalty

$$\lambda \int (g''(t))^2 \, \mathrm{d}t$$

- Integral: a summation over the range of t
- The penalty is a measure of the total change in the function $g''(t)$ over its entire range
  - if g is close to an affine fit, then $g'(t)$ will be close to constant and $g''(t)^2$ will take a small value
  - if g is jumpy and variable than $g''(t)$ will vary significantly and the $\int$ will take a large value
- **The penalty encourages g to be smooth**
- The larger is the value of λ, the closer g will be to an affine fit (straight line)

Smoothing Splines: penalty

- When $\lambda = 0$ the penalty term has no effect $\implies$ g will be very jumpy and will exactly interpolate the training observations
- When $\lambda \to \infty$, g will be perfectly affine: a straight line that passes as closely as possible to the training points
  - g will be the linear least squares line, since the loss function=RSS
- For an intermediate value of λ, g will approximate the training observations but will be somewhat smooth
- **λ controls the bias - variance trade - off of the smoothing spline**

Smoothing Splines

$$\sum_{i=1}^{n} (y_i - g(x_i))^2 + \lambda \int (g''(t))^2 \, \mathrm{d}t$$

- The function $g(x)$ that minimizes this criteria has special properties:
  1. it is a piecewise cubic polynomial with knots at $x_1, \ldots, x_n$
  2. it has continuous first and second derivatives at each knot
  3. it is linear in the region outside of the extreme knots
- (1) - (3) $\Longrightarrow$ **the function $g(x)$ is a natural cubic spline with knots at $x_1, ..., x_n$!**

Smoothing Splines

$$\sum_{i=1}^{n} (y_i - g(x_i))^2 + \lambda \int (g''(t))^2 \, \mathrm{d}t$$

- It is not the same natural cubic spline that we would get applying the basis function approach with knots at $x_1, ..., x_n$
- A shrunken version, where the value of the tuning parameter λ controls the level of shrinkage.

Choosing the Smoothing Parameter λ

- A smoothing spline: a natural cubic spline with knots at every unique value of x_i
- It might seem that a smoothing spline will have far too many degrees of freedom: a knot at each data point allows a large flexibility
- The tuning parameter λ controls the roughness of the smoothing spline
- We do not need to select the number or the locations of the knots but we need to choose the value of λ

Choosing the Smoothing Parameter λ

- Using cross-validation:
  - find the value of λ that makes the cross-validated RSS as small as possible
- The leave-one-out cross-validation error can be computed very efficiently for smoothing splines with essentially the same cost as computing a single fit

Outline

Step Functions

Regression Splines

Smoothing Splines

Local Regression

Generalized Additive Models

Local Regression

Local regression is an approach for fitting flexible non-linear functions, which involves computing the fit at a target point x_0 *using only nearby training observations*

Step Functions

Regression Splines

Smoothing Splines

Local Regression

Generalized Additive Models

Local Regression

Algorithm: Local Regression at $X = x_0$

1. Gather the fraction $s = k/n$ of training points whose x_i are closest to x_0
2. Assign a weight $K_{(i0)} = K(x_i, x_0)$ to each point in this neighborhood. All but k nearest neighbors get weight zero.
3. Fit a weighted least squares regression of the y_i on the x_i using the weights:

$$\sum_{i=1}^{n} K_{(i0)}(y_i - \beta_0 - \beta_1 x_i)^2$$

4. The fitted value at x_0 is given by $\hat{f}(x_0) = \hat{\beta}_0 + \hat{\beta}_1 x_0$

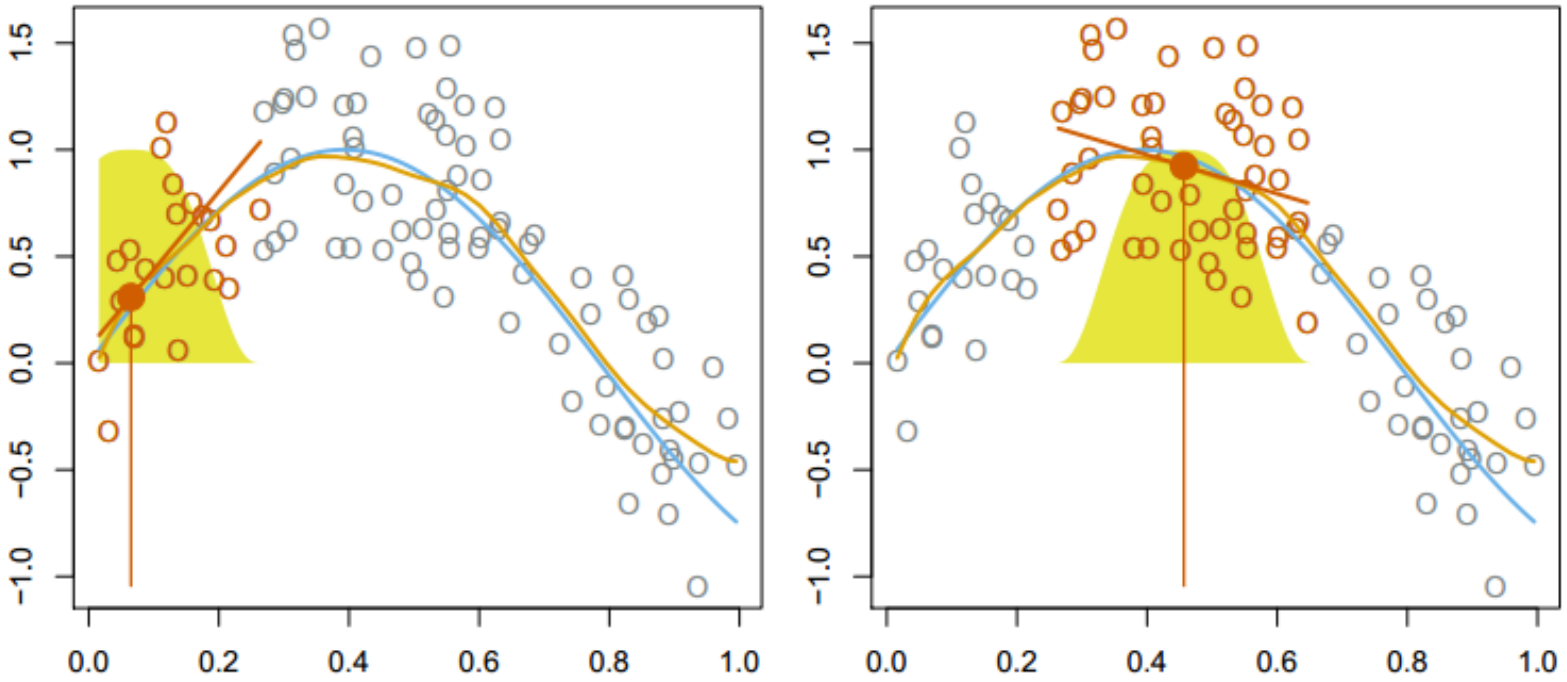
Local Regression

- In Step 3 of Algorithm the weights $K_{(i0)}$ will differ for each value of x_0
- The weight function gives the most weight to the data points nearest the point of estimation and the least weight to the data points that are furthest away:
  - e.g., the tri-cube weight function:

$$w(x) = (1 - |d|^3)^3$$

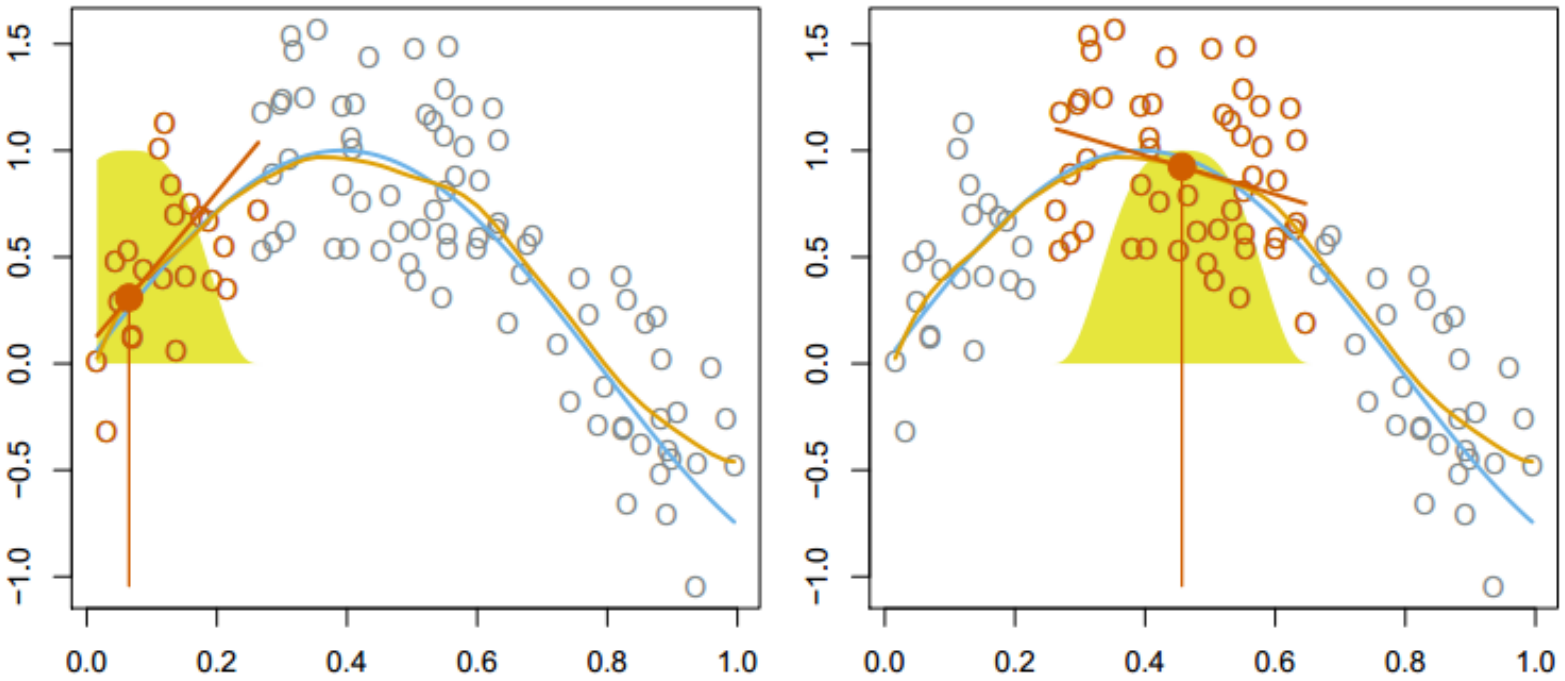
    where d is the distance of a given data point to the point on the curve being fitted, scaled to lie in $[0, 1]$
- Local regression is referred to as a *memory-based procedure*: we need all the training data each time we wish to compute a prediction.

Local Regression: example



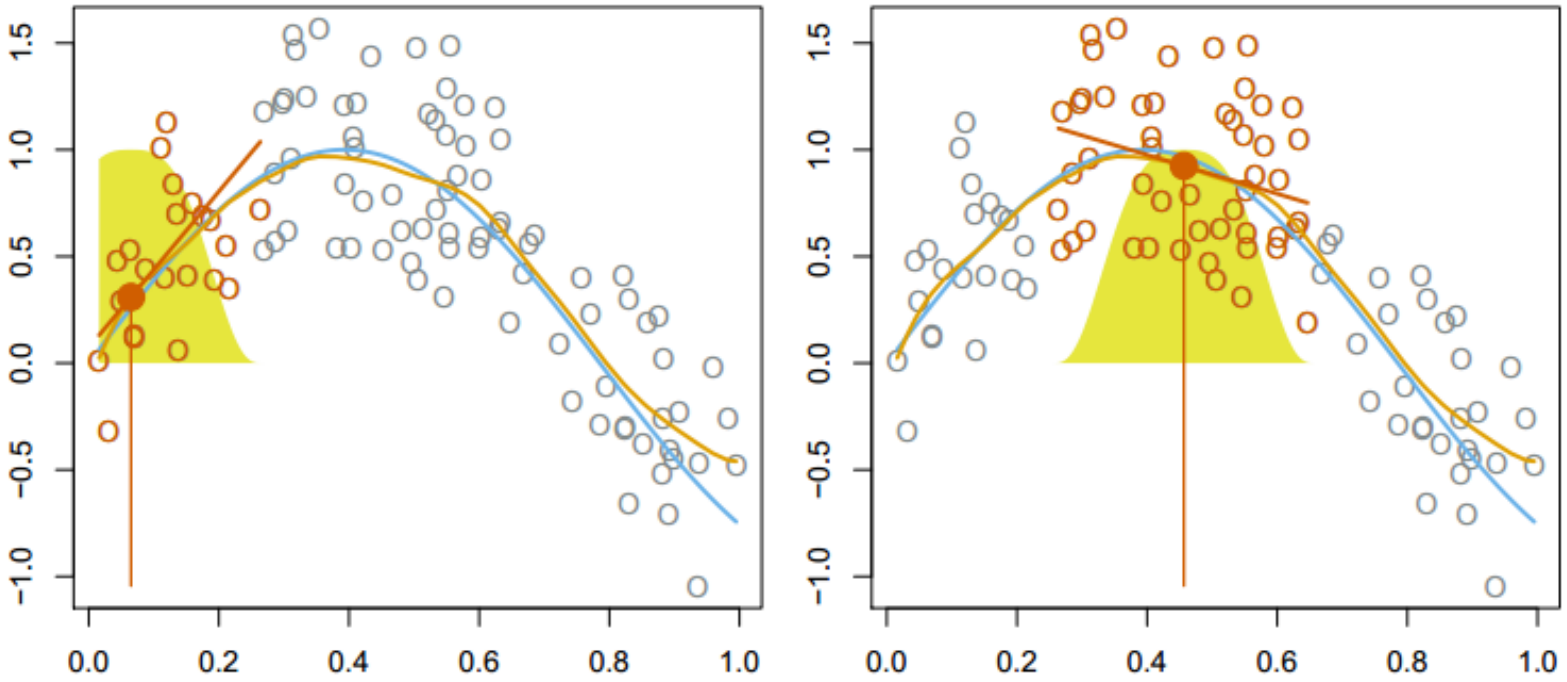
- one target point near 0.4, and another near the boundary at 0.05
- the blue line represents the function $f(x)$ from which the data were generated
- the orange line corresponds to the local regression estimate $\hat{f}(x)$

Local Regression: example



- the orange points are local to the target point x_0
- the green bell-shape indicates weights assigned to each point, decreasing to zero with distance from the target point

Local Regression: example



- the fit at x_0 is obtained by fitting a weighted linear regression (orange line segment)
- the fitted value at x_0 (orange solid dot) is the estimate $\hat{f}(x_0)$

Local Regression

- In order to perform local regression, there are a number of choices to be made:
  - how to define the weighting function K
  - whether to fit a linear, constant, or quadratic regression in Step 3
- The most important choice is the fraction s (Step 1)
  - it plays a role of the tuning parameter λ in smoothing splines: it controls the flexibility of the fit
  - the smaller the value of s, the more local and wiggly will be our fit
  - a very large value of s will lead to a global fit to the data using all of the training observations
  - we can use cross-validation to choose s, or we can specify it directly.

Local Regression: generalizations

Multiple features $X_1, X_2, ..., X_p$:

- fitting a multiple linear regression model that is global in some variables, but local in another, such as time:

Varying coefficient models

- allows to adapt the model to the most recently gathered data

Local Regression: generalizations

Models that are local in a pair of variables X_1 and X_2 rather than one:

- use two-dimensional neighborhoods to fit bivariate linear regression models using the observations that are near each target point in two-dimensional space
- the same approach can be implemented in higher dimensions, using linear regressions fit to p-dimensional neighborhoods
  - local regression can perform poorly if p is large as there will generally be very few training observations close to x_0 (similar to Nearest-neighbors regression)

Outline

Step Functions

Regression Splines

Smoothing Splines

Local Regression

Generalized Additive Models

Generalized Additive Models

- Flexibly predicting a response Y on the basis of a single predictor X: extensions of simple linear regression
- Flexibly predicting Y on the basis of several predictors $X_1, ..., X_p$:

Generalized additive models (GAMs)

- an extension of multiple linear regression
- a general framework for extending a standard linear model by allowing non-linear functions of each of the variables, while maintaining additivity
- GAMs can be applied with both quantitative and qualitative responses.

Generalized additive models

Replace each linear component $\beta_j X_j$ with a (smooth) non-linear function $f_j(X_j)$:

$$y_i = \beta_0 + f_1(x_{i1}) + f_2(x_{i2}) + \cdots + f_p(x_{ip}) + \epsilon_i$$

- It is an additive model because we calculate a separate f_j for each X_j and then add together all of their contributions
- Methods for fitting functions to a single variable can be used as building blocks for fitting an additive model.

Example: Wage data

Predicting wage as function of year, age and education:

$$\text{wage} = \beta_0 + f_1(\text{year}) + f_2(\text{age}) + f_3(\text{education}) + \epsilon$$

- year and age are quantitative variables
- education is the level of high school or college education that an individual has completed:
  - a qualitative variable with five levels: $<$HS, HS,$<$Coll, Coll, $>$Coll
- We fit f_1 and f_2 using natural splines
- We fit f_3 using a separate constant for each level, via the usual dummy variable approach

Example: Wage data

Predicting wage as function of years in working force, age and education:

$$\text{wage} = \beta_0 + f_1(\text{year}) + f_2(\text{age}) + f_3(\text{education}) + \epsilon$$

- We can use least squares:
  - natural splines can be constructed using an appropriately chosen set of basis functions
  - the entire model is just a big regression onto spline basis variables and dummy variables

Example: Wage data

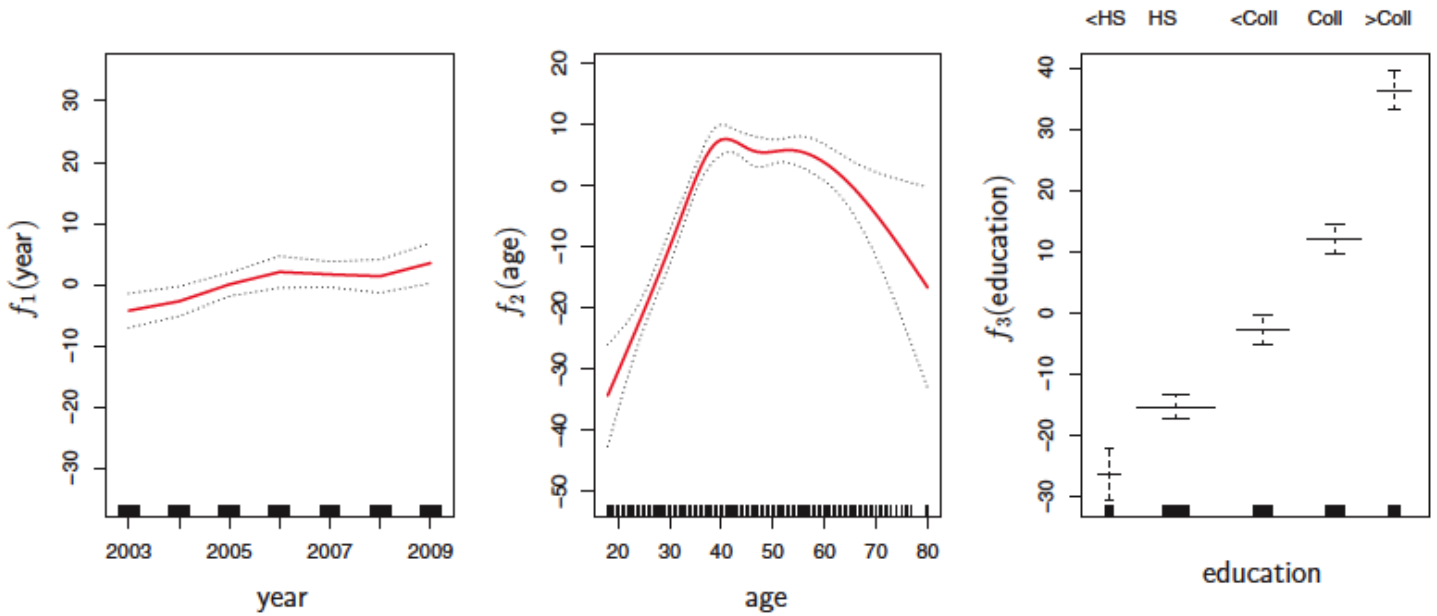


FIGURE 7.11. *For the* `Wage` *data, plots of the relationship between each feature and the response,* `wage`*, in the fitted model (7.16). Each plot displays the fitted function and pointwise standard errors. The first two functions are natural splines in* `year` *and* `age`*, with four and five degrees of freedom, respectively. The third function is a step function, fit to the qualitative variable* `education`*.*

- The left-hand panel: holding age and education fixed, wage tends to increase slightly with year

Example: Wage data

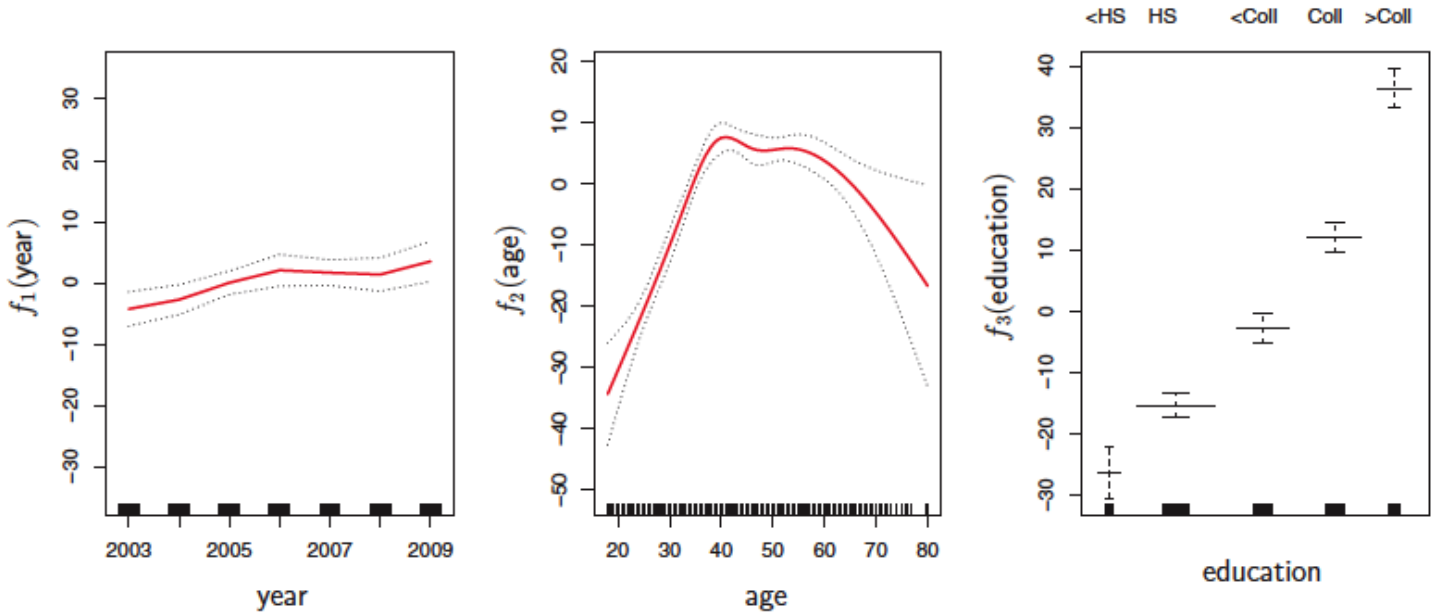


FIGURE 7.11. *For the* `Wage` *data, plots of the relationship between each feature and the response,* `wage`*, in the fitted model (7.16). Each plot displays the fitted function and pointwise standard errors. The first two functions are natural splines in* `year` *and* `age`*, with four and five degrees of freedom, respectively. The third function is a step function, fit to the qualitative variable* `education`*.*

- The center panel: holding education and year fixed, wage tends to be highest for intermediate values of age, and lowest for the very young and very old.

Example: Wage data

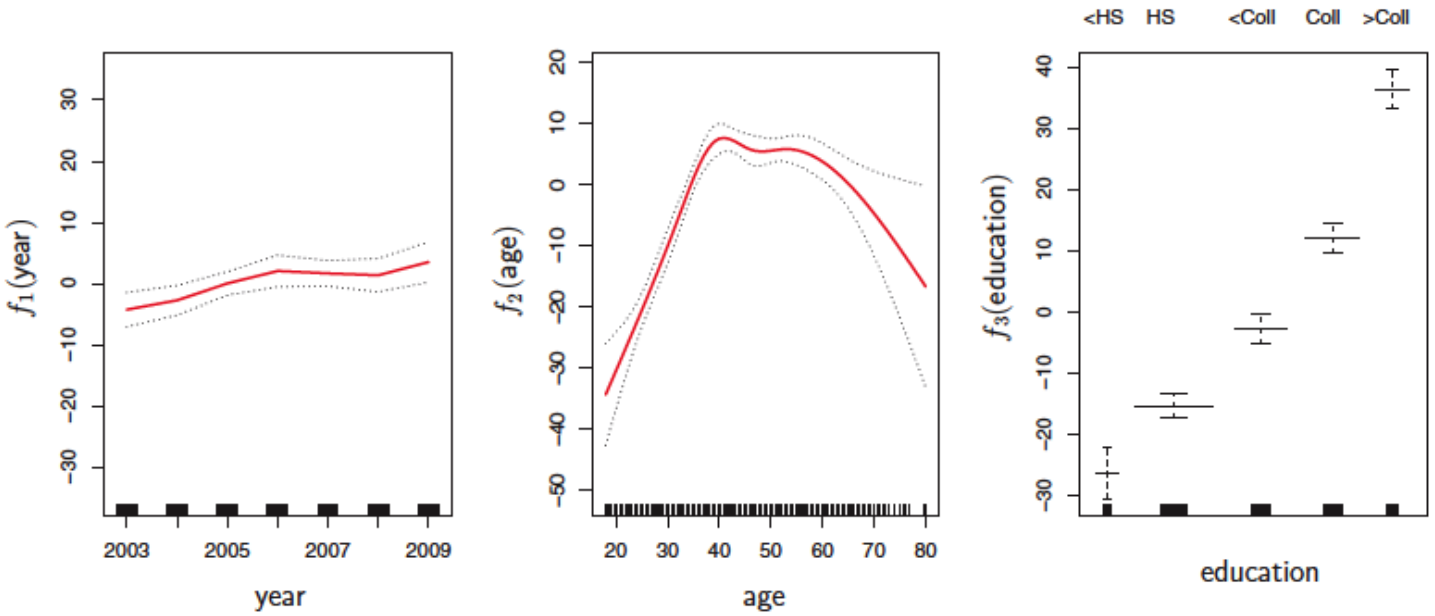


FIGURE 7.11. *For the* `Wage` *data, plots of the relationship between each feature and the response,* `wage`*, in the fitted model (7.16). Each plot displays the fitted function and pointwise standard errors. The first two functions are natural splines in* `year` *and* `age`*, with four and five degrees of freedom, respectively. The third function is a step function, fit to the qualitative variable* `education`*.*

- The right-hand panel: holding year and age fixed, wage tends to increase with education

GAMs

- As the building blocks for GAMs we can use local regression, polynomial regression, or any combination of these approaches
- Advantages:
  - GAMs allow us to fit a non-linear f_j to each X_j
  - We do not need to manually try out many different transformations on each variable individually
  - The non-linear fits can potentially give more accurate predictions for the response Y
  - As the model is additive, we can still examine the effect of each X_j on Y individually while holding all of the other variables fixed
  - If we are interested in inference, GAMs provide a useful representation

Limitation of GAMs

- The model is restricted to be additive
- With many variables, important interactions can be missed
- We can manually add interaction terms to the GAM model by including additional predictors of the form $X_j \times X_k$
- We can add low-dimensional interaction functions of the form $f_{(jk)}(X_j, X_k)$ into the model
  - can be fitted using two-dimensional local regression, or two-dimensional splines
- For fully general models, we have to look for even more flexible approaches such as random forests and boosting.

References

- Trevor Hastie, Robert Tibshirani, Jerome Friedman. *The elements of statistical learning - Data Mining, inference, and prediction.* Springer.

- Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani. *An Introduction to Statistical Learning with applications in R.* Springer.