Splines

Therese M-L Andersson

Biostatistics 1:Introduction to biostatistics

November 2024

Introduction

- Previously we have modelled **linear** relationships between Y (response) & X (predictor)
- The truth is almost never linear; but sometimes good enough
- One alternative is to categorise continuous variables
- **Continuous non-linear** relationships are more flexible, and sometimes preferable

Example data

- Let's say that we want to model the relationship between the predictor X and the response Y, plotted in the graph below

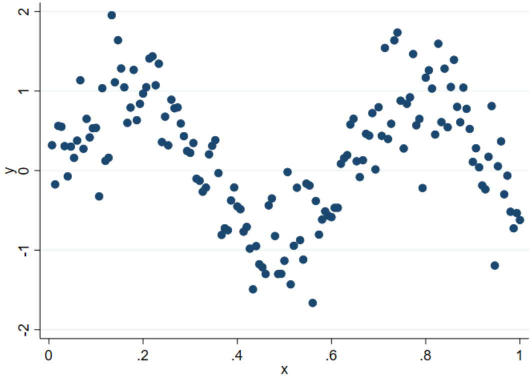


Figure 1: Scatter plot of example data, tmp

Linear function

Linearity gives the result below

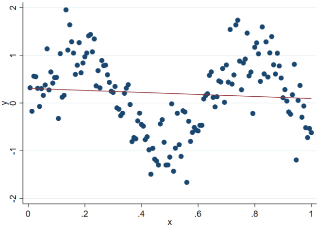


Figure 2: Linearity

```
Call:
lm(formula = y ~ x, data = tmp)

Coefficients:
             Estimate  Std. Error t value Pr(>|t|)
(Intercept)   0.3064     0.1274    2.405   0.0174
x            -0.2119     0.2196   -0.965   0.3362

R-squared:  0.00625
```

This can be written as:

- $y = f(x) = \beta_0 + \beta_1 \cdot x$
- $y = f(x) = \beta_0 \cdot b_0(x) + \beta_1 \cdot b_1(x)$
- $y = f(x) = \sum_{i=0}^{1} \beta_i \cdot b_i(x)$,
  where $b_0(x) = 1$, $b_1(x) = x$ are called **basis** functions

Polynomials

Alternatively (and in this case better), we can fit a non-linear model using polynomials

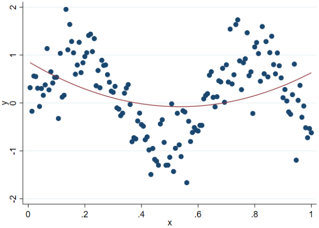


Figure 3: Polynomial with degree 2

```
Call:
lm(formula = y ~ x + I(x^2), data = tmp)

Coefficients:
            Estimate  Std. Error t value Pr(>|t|)
(Intercept)  0.8655     0.1834    4.721 5.43e-06
x           -3.5223     0.8409   -4.189 4.82e-05
I(x^2)       3.2885     0.8092    4.064 7.82e-05

R-squared:  0.1066
```

This can be written as:

$$f(x) = \sum_{i=0}^{2} \beta_i \cdot b_i(x)$$

What are the basis functions?

Polynomials

Alternatively (and in this case better), we can fit a non-linear model using polynomials

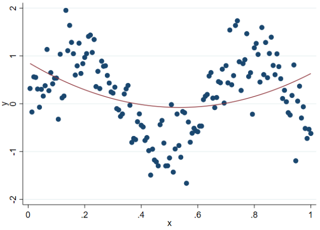


Figure 3: Polynomial with degree 2

```
Call:
lm(formula = y ~ x + I(x^2), data = tmp)

Coefficients:
             Estimate  Std. Error t value Pr(>|t|)
(Intercept)   0.8655     0.1834    4.721 5.43e-06
x            -3.5223     0.8409   -4.189 4.82e-05
I(x^2)        3.2885     0.8092    4.064 7.82e-05

R-squared:  0.1066
```

This can be written as:

$$f(x) = \sum_{i=0}^{2} \beta_i \cdot b_i(x)$$

What are the basis functions?

$$b_0(x) = 1, b_1(x) = x, b_2(x) = x^2$$

Polynomials

More flexibility with higher degree polynomials

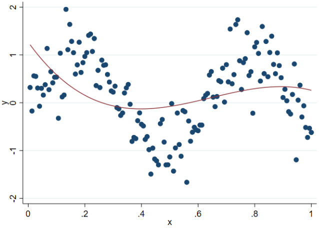


Figure 4: Polynomial with degree 3

```
Call:
lm(formula = y ~ x + I(x^2) + I(x^3), data = tmp)

Coefficients:
            Estimate  Std. Error t value Pr(>|t|)
(Intercept)  1.2678     0.2432   5.212 6.25e-07
x           -8.2389     2.0854  -3.951 0.000121
I(x^2)      14.9631     4.8052   3.114 0.002222
I(x^3)      -7.7315     3.1383  -2.464 0.014918

R-squared:  0.1423
```

This can be written as:

$$f(x) = \sum_{i=0}^{3} \beta_i \cdot b_i(x)$$

What are the basis functions?

Polynomials

More flexibility with higher degree polynomials

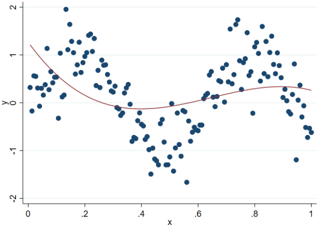


Figure 4: Polynomial with degree 3

```
Call:
lm(formula = y ~ x + I(x^2) + I(x^3), data = tmp)

Coefficients:
             Estimate  Std. Error t value Pr(>|t|)
(Intercept)   1.2678     0.2432    5.212 6.25e-07
x            -8.2389     2.0854   -3.951 0.000121
I(x^2)       14.9631     4.8052    3.114 0.002222
I(x^3)       -7.7315     3.1383   -2.464 0.014918

R-squared:  0.1423
```

This can be written as:

$$f(x) = \sum_{i=0}^{3} \beta_i \cdot b_i(x)$$

What are the basis functions?

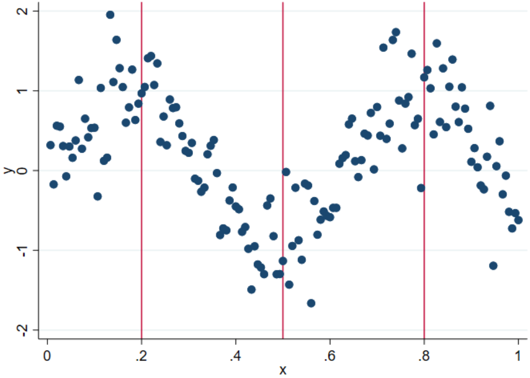
$$b_0(x) = 1, b_1(x) = x, b_2(x) = x^2, b_3(x) = x^3$$

Piecewise

If *polynomials* do not work, we can:

- divide the total function into parts using **knots**

  Here: $m_1 = 0.2, m_2 = 0.5, m_3 = 0.8$
- fit various polynomials for each part



Piecewise constant

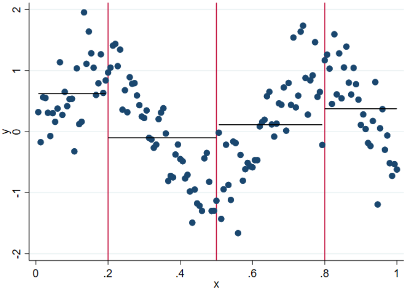


Figure 5: Piecewise constant

```
Call:
lm(formula = y ~ 0 + xp1 + xp2 + xp3 + xp4,
    data = tmp)

Coefficients:
    Estimate Std. Error t value Pr(>|t|)
xp1   0.6335     0.1339   4.731 5.22e-06
xp2  -0.1247     0.1093  -1.140   0.2560
xp3   0.1364     0.1093   1.248   0.2142
xp4   0.3479     0.1339   2.598   0.0103

R-squared:  0.1797
```

$$f(x) = \sum_{i=0}^{3} \beta_i \cdot b_i(x)$$

Basis functions:

$$b_0(x) = 1 \text{ if } (x < m_1),$$
$$b_1(x) = 1 \text{ if } (m_1 \geq x < m_2),$$
$$b_2(x) = 1 \text{ if } (m_2 \geq x < m_3),$$
$$b_3(x) = 1 \text{ if } (x \geq m_3).$$

Piecewise cubic

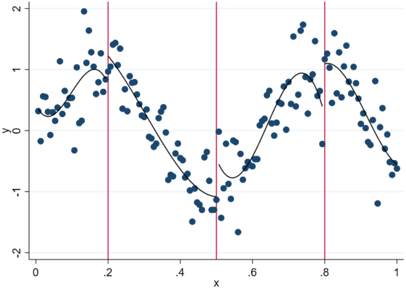


Figure 6: Piecewise cubic

```
Call:
lm(formula = y ~ 0 + int1 + int2 + int3 + int4 + xp1 + xp2 +
    xp3 + xp4 + xpsq1 + xpsq2 + xpsq3 + xpsq4 + xpcub1 +
    xpcub2 + xpcub3 + xpcub4, data = tmp)
Coefficients:
          Estimate Std. Error t value Pr(>|t|)
int1        0.3936     0.3439   1.145  0.25445
int2        2.5333     4.8950   0.518  0.60564
int3       96.1711    32.9519   2.919  0.00413
int4     -292.9223   368.6509  -0.795  0.42826
xp1       -10.2916    14.1744  -0.726  0.46906
xp2         0.0537    44.2323   0.001  0.99903
xp3      -473.5037   153.9935  -3.075  0.00255
xp4       984.3260  1229.2547   0.801  0.42469
xpsq1     189.9589   157.9880   1.202  0.23134
xpsq2     -40.7607   128.7640  -0.317  0.75208
xpsq3     756.3631   237.6620   3.183  0.00182
xpsq4   -1087.6323  1363.4194  -0.798  0.42644
xpcub1   -635.6230   503.0196  -1.264  0.20856
xpcub2     52.1983   121.1653   0.431  0.66730
xpcub3   -392.4121   121.1653  -3.239  0.00151
xpcub4    395.6790   503.0196   0.787  0.43290
R-squared:  0.7609
```

Basis functions:

$b_0(x) = 1, b_1(x) = x, b_2(x) = x^2, b_3(x) = x^3$ if $(x < m_1)$ and 0 otherwise,

$b_4(x) = 1, b_5(x) = x, b_6(x) = x^2, b_7(x) = x^3$ if $(m_1 \geq x < m_2)$ and 0 otherwise,

$b_8(x) = 1, b_9(x) = x, b_{10}(x) = x^2, b_{11}(x) = x^3$ if $(m_2 \geq x < m_3)$ and 0 otherwise,

$b_{12}(x) = 1, b_{13}(x) = x, b_{14}(x) = x^2, b_{15}(x) = x^3$ if $(x \geq m_3)$ and 0 otherwise.

Alternatively, the same function can be rewritten with different basis functions as:

$$
\begin{aligned}
f(x) &= \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 \\
&+ \beta_4 (x - m_1)_+^0 + \beta_5 (x - m_1)_+^1 + \beta_6 (x - m_1)_+^2 + \beta_7 (x - m_1)_+^3 \\
&+ \beta_8 (x - m_2)_+^0 + \beta_9 (x - m_2)_+^1 + \beta_{10} (x - m_2)_+^2 + \beta_{11} (x - m_2)_+^3 \\
&+ \beta_{12} (x - m_3)_+^0 + \beta_{13} (x - m_3)_+^1 + \beta_{14} (x - m_3)_+^2 + \beta_{15} (x - m_3)_+^3
\end{aligned}
$$

The $+$ function is defined as:

$$u_+ = u \text{ if } u > 0$$

$$u_+ = 0 \text{ if } u \leq 0$$

and

$$u_+^0 = 1 \text{ if } u > 0$$

$$u_+^0 = 0 \text{ if } u \leq 0$$

Splines

- The fitted function in the previous slide is not continuous
- To connect the function at the knots and make the function more smooth, continuity constraints can be enforced by removing specific terms
- This gives the function:

$$f(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \sum_{i=1}^{3} \beta_{i+3}(x - m_i)^3$$

- This function is an example of regression splines
- You will be asked to derive the above equation in an exercise.

Splines

- Splines are defined by piecewise polynomials of a certain degree (k) connected at specific points called knots (m).
- Smoothness is enforced by continuity in derivatives up to degree $k-1$ at each knot.

Truncated power basis functions:

$$
\begin{aligned}
b_0(x) &= 1, & \\
b_i(x) &= x^i, & \text{for } i = 1, \ldots k \\
b_{k+j}(x) &= (x - m_j)_+^k, & \text{for } j = 1, \ldots, M
\end{aligned}
$$

- Also possible to restrict the function to be linear before the first and after the last knot (more on that later).

Splines

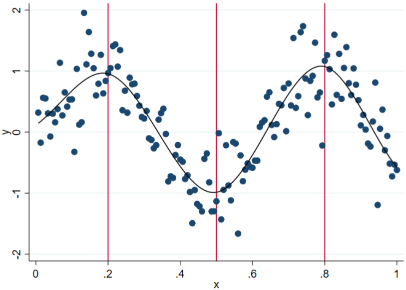


Figure 7: Cubic spline

```
Call:
lm(formula = y ~ x + I(x^2) + I(x^3) + xptr1 + xptr2
    + xptr3, data = tmp)

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)    0.1152     0.2587   0.445  0.65679
x              4.1446     6.4288   0.645  0.52016
I(x^2)        29.3238    42.6074   0.688  0.49242
I(x^3)      -145.2250    81.9182  -1.773  0.07839
xptr1        283.2815    93.0715   3.044  0.00278
xptr2       -294.7164    25.8616 -11.396  < 2e-16
xptr3        358.0500    85.8638   4.170 5.26e-05

R-squared:  0.7224
```

- Typically, *cubic* polynomials ($k = 3$) are used due to their ability to maintain smoothness while minimising complexity [1].

B-splines

- Firstly proposed by de Boor (1978)
- They are based on the idea of Cubic Beizier curve

$$C_0(t) = (1-t)^3 m_0 + 3(1-t)^2 \cdot t \cdot m_1 + 3(1-t) \cdot t^2 \cdot m_2 + t^3 \cdot m_3,$$

where $m_0, \dots m_3$ are knots

- Basis function B_i of order k is defined recursively
- The basis function of order 0 $B_{i,0}$ depends on 2 knots, m_i, m_{i+1}, while $B_{i,1}$ depends on 3 knots $m_i, m_i + 1, m_{i+2}$, and $B_{i,k}$ depends on $k+2$ knots

B-splines, features

- The number of needed B basis functions $i = n + k + 1$, where n - the number of internal knots
- The B-spline basis functions are nonzero over an interval spanning at most $k + 2$ knots, i.e. $B_{i,k}$ is positive for $x \in (t_i, t_{i+k+1})$ and zero for outside the interval
- Non zero basis functions sums to 1

B-basis functions, example

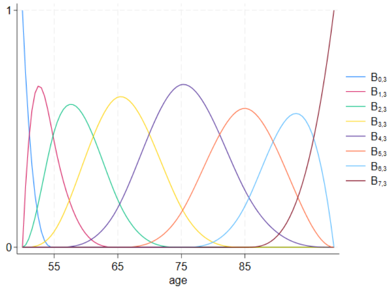


Figure 8

- The number of basis functions $= 4 + 3 + 1 = 8$
- Each of them must be non-zero at most 5 knots (5 consecutive intervals), here max is 3
- All of them sum to 1
- Each basis function can be split into an increasing section and a decreasing one, except
- The first and last curve are the only monotonous functions in the entire set (strictly decreasing / strictly increasing)

Natural splines

Cubic polynomial functions can be unstable near the boundaries, thus, extrapolation can be challenging

Additional constraints: f(x) is linear beyond the boundaries, i.e. the first & second derivatives are zero outside the boundaries

Definition: If $\mathbf{B_3(x)}$ represent the cubic B-spline basis vector, then $N(x) = \mathbf{H^T} \cdot \mathbf{B_3(x)}$ is a vector of natural cubic basis if $\mathbf{C^T} \cdot \mathbf{H} = 0$, where $\mathbf{C^T}$ is a matrix of second derivatives of B-basis functions at boundary knots:

$$\mathbf{C} = \left(\frac{d^2\mathbf{B_3(x)}}{dx^2}\Big|_{x=L}, \frac{d^2\mathbf{B_3(x)}}{dx^2}\Big|_{x=U}\right)$$

and $\mathbf{H}$ is a $i \times l$ full column rank matrix ($i = m + 3 + 1$ is the number of B-basis functions, $l = m + 2$ is the number of natural spline basis functions, and m is the number of internal knots)

Natural splines

Matrix **C** can be explicitly written down for cubic B-splines

Then the chosen matrix **H** consists of nonnegative elements. The natural cubic basis are thus nonnegative within the boundary [3]:

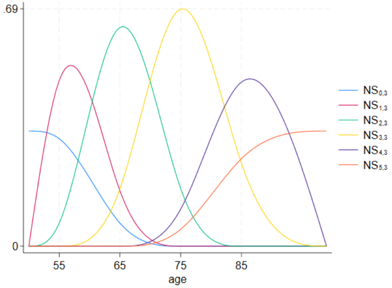


Figure 9

Natural vs B-splines

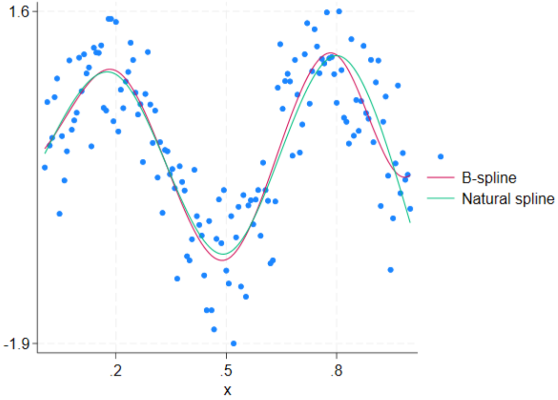


Figure 10

Smoothing splines

All above discussed splines are referred to *regression* splines.

Smoothing splines, also known as *P-splines*, is a commonly used penalized regression splines. It based on the cubic B-spline basis function and a second-order difference penalty defined as:

$$J_\beta^* = \sum (\Delta^2 \beta_k)^2$$

P-spline offers an alternative approach to smoothly model the underlying function by incorporating a penalty functions and a large amount of knots[2].

Do not require knot selection, but may not capture complex relationships in the data.

Determining the parameter on the penalty function can lead to an additional optimization problem.

Bs, Ns, & penalised

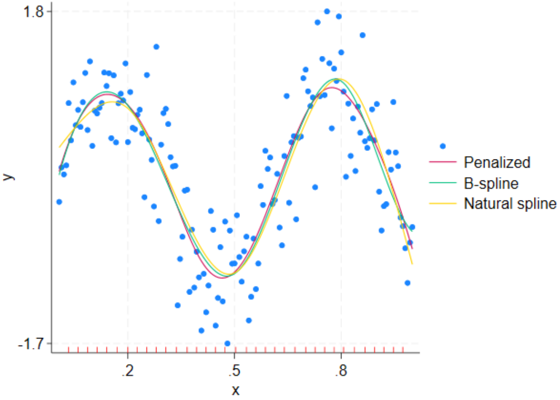


Figure 11: Caption

Choice of the number and location of knots

[1] P.L. Smith. "Splines As a Useful and Convenient Statistical Tool". In: *The American Statistician* 33.2 (1979), pp. 57–62. ISSN: 00031305. URL: http://www.jstor.org/stable/2683222 (visited on 05/23/2023).

[2] A. Perperoglou et al. "A review of spline function procedures in R". In: *BMC Medical Research Methodology* 19.46 (2019). DOI: https://doi.org/10.1186/s12874-019-0666-3. URL: https://doi.org/10.1186/s12874-019-0666-3.

[3] Wenjie Wang and Jun Yan. "Shape-Restricted Regression Splines with R Package splines2". In: *Journal of Data Science* 19.3 (2021), pp. 498–517. ISSN: 1680-743X. DOI: 10.6339/21-JDS1020.