STA 35C: Statistical Data Science III

Lecture 21: Regression Splines (cont'd) & Smoothing Splines

Dogyoon Song

Spring 2025, UC Davis

Today's topics

- **Quick review**
  - Basis function
  - Regression spline
- **Regression splines** (cont'd)
  - Truncated power basis representation
  - "Natural" splines
  - How to place knots
- **Smoothing splines**
  - Overview: interpolation + smoothness penalty
  - Choosing the smoothness parameter

Quick review: Basis functions & regression splines

Basis function: Fit a linear model in transformed features $b_1(X), \ldots, b_K(X)$

$$Y \approx \beta_0 + \beta_1 \cdot b_1(X) + \cdots + \beta_K \cdot b_K(X)$$

- Retains the simple linear-model form yet can model nonlinearities flexibly
- Examples:
  - *Polynomials:* $b_1(X) = X,\ b_2(X) = X^2, \ldots$
  - *Step functions:* $b_1(X) = I_{(c_1, c_2]}(X),\ b_2(X) = I_{(c_2, c_3]}(X), \ldots$

Regression splines:

- *Piecewise polynomials* of degree d, joined smoothly at knots (cutpoints)
- *Continuity constraints* at the knots for the function and its first $(d-1)$ derivatives
- Degrees of freedom: a degree-d spline with K knots has $(d+1) + K$ parameters

Why regression splines?

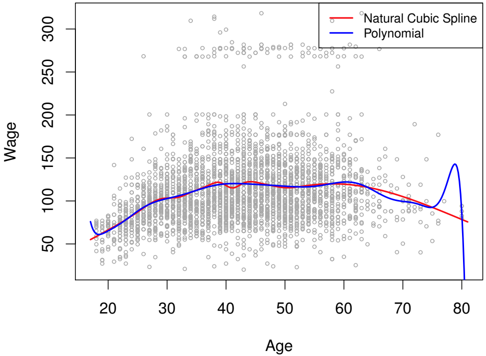

Figure: On the Wage data set, a natural cubic spline with 15 degrees of freedom (**blue**) vs. a degree-15 polynomial (**red**). Polynomials can oscillate excessively near the edges, while splines are more stable [JWHT21, Figure 7.7].

- Higher-degree polynomials can be flexible but often exhibit unwanted oscillations
- Splines restrict the polynomial degree while increasing flexibility via *knots*, yielding more stable fits

Spline basis representation: Truncated power basis

Key question: How to construct a piecewise polynomial that remains $d-1$ times continuously differentiable at each knot?

Truncated power basis for a degree-d spline:

$$\underbrace{1,\ X,\ X^2, \ldots, X^d}_{\text{base polynomials}} \ \cup\ \{ \quad \underbrace{(X-c_k)_+^d}_{\text{truncated power basis}} \quad : k = 1 \ldots K\}$$

where $(x-c)_+^d = \max\{x-c, 0\}^d$

- Then, a regression spline has the form

$$f(x) = \beta_0 + \beta_1 X + \cdots + \beta_d X^d \ + \ \sum_{k=1}^{K} \beta_{d+k} \, (X - c_k)_+^d$$

- This representation automatically ensures continuity up to order $d-1$ at each knot
- Software packages (`splines` in R, etc.) typically handles this basis internally

Spline basis representation: Truncated power basis (cont'd)

A closer look into why/how truncated power basis ensures continuity:

- A function f is continuous at x_0 if

$$\lim_{x \to x_0-} f(x) = f(x_0) = \lim_{x \to x_0+} f(x)$$

- Observe that for $f(x) = (x - c_k)_+^d$,

$$\begin{aligned} \lim_{x \to c_k-} f(x) &= 0, \\ f(c_k) &= 0, \\ \lim_{x \to c_k+} f(x) &= 0. \end{aligned}$$

- You can similarly verify the continuity of derivatives; Homework 5, Problem 3-1)

Example: Truncated power basis for linear spline

Example

Let X range in $[0, 8]$ with knots at $x = 2, 5$. Use piecewise linear polynomials (degree $d = 1$). Hence, from DoF formula $(d + 1) + K = (1 + 1) + 2 = 4$ total parameters.

Basis representation:

$$b_1(x) = 1, \quad b_2(x) = x, \quad b_3(x) = (x - 2)_+, \quad b_4(x) = (x - 5)_+, \qquad (u)_+ = \max(u, 0).$$

Then the resulting linear spline model—which can be fit by least squares—is

$$\widehat{y}(x) = \beta_1\, b_1(x) + \beta_2\, b_2(x) + \beta_3\, b_3(x) + \beta_4\, b_4(x).$$

Interpretation:

- β_1 is the base intercept.
- β_2 is the slope for $0 \leq x \leq 2$.
- β_3 modifies the slope for $2 < x \leq 5$, so the slope in (2,5] is $\beta_2 + \beta_3$.
- β_4 further modifies the slope for $x > 5$, so the slope in $(5, 8]$ is $\beta_2 + \beta_3 + \beta_4$.

Natural splines

Even a moderate-degree spline can exhibit wild curvature near the boundary

Natural splines[1] impose extra constraints at the boundary so that the spline is linear beyond the outermost knots:

- In practice, "natural spline" typically means a *natural cubic spline*
- For a cubic spline, this is equivalent to forcing $f''(x) = 0$ beyond the outer knots
  - This imposes two extra constraints, reducing DoF by 2
  - A natural cubic spline has $(K + 2)$ parameters, whereas a cubic spline with K knots has $(K + 4)$
- This usually restrains erratic tail behavior and yields narrower confidence intervals

[1] A canonical basis for natural splines exists, but we skip details here. In R, see `splines::ns()`.

Illustration: Cubic spline vs. natural cubic spline

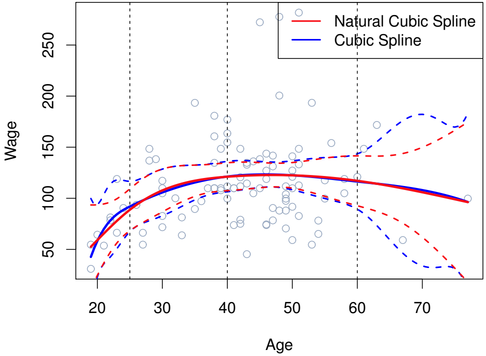

Figure: A cubic spline (**blue**) vs. a natural cubic spline (**red**) on a subset of the `Wage` data; vertical dashed lines show 3 knot locations. Note that natural spline is linear beyond the outer knots [JWHT21, Figure 7.4].

- Confidence intervals are narrower for natural splines
- Less risk of “wild” behavior near edges

Pop-up quiz #1: Splines & natural splines

Which statement is false regarding cubic and natural cubic splines?

A) A cubic spline with K knots has $(K + 4)$ degrees of freedom.

B) A natural cubic spline with K knots forces linear behavior outside the outermost knots.

C) Enforcing the second derivative to be zero at the boundaries *increases* the total degrees of freedom by 2.

D) A natural cubic spline with K knots has $(K + 2)$ degrees of freedom.

Answer: (C) is false.

Enforcing a zero second derivative at the boundaries *reduces* the degrees of freedom by 2, which explains why a natural cubic spline with K knots has $(K + 2)$ rather than $(K + 4)$ parameters.

Where to place knots?

Knots may be placed in various ways:

- *Uniform width:* Evenly spaced across the range of X
- *Uniform mass:* At quantiles, so each segment has roughly equal datapoints
- Additionally, domain knowledge may help identify critical breakpoints
- Cross-validation can be used to pick an optimal set of knots

Typical practice:

- For smaller data sets, use a moderate number of knots (e.g. 3–5)
- For large data or highly nonlinear relationships, more knots might help
- Choose or refine knot placement by cross-validation or certain information criteria

How many knots should we have?

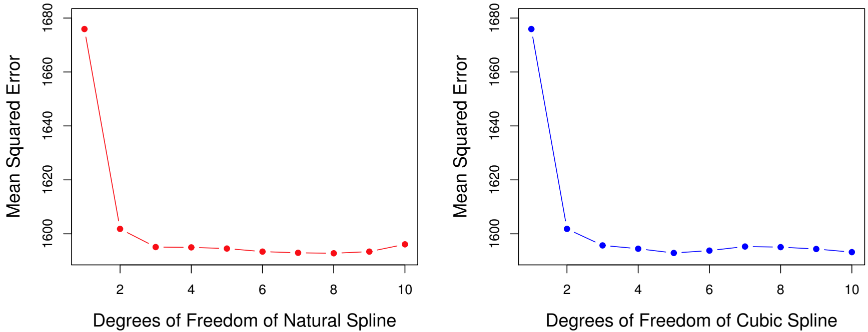

Figure: CV MSEs for different degrees of freedom in splines on the `Wage` data, modeling `wage` vs. `age`. **Left:** natural cubic spline. **Right:** cubic spline [JWHT21, Figure 7.6].

Question: how many knots $\longleftrightarrow$ how many degrees of freedom

$\Longrightarrow$ **Answer:** use cross-validation

Example 2: Placing knots (uniform width vs. uniform mass)

Example

Setup: Suppose we have a hypothetical dataset with $n = 12$ points, $X \in [0, 12]$:

$$X = \{0.2,\ 0.9,\ 1.1,\ 2.2,\ 5.5,\ 6.0,\ 6.2,\ 8.0,\ 8.2,\ 10.8,\ 11.9,\ 12.0\}.$$

Two ways to pick knots (2 interior knots => 3 segments):

1) **Uniform width**: Even spacing in $[0, 12]$.
   - e.g. knots at $x = 4$ and $x = 8$.
   - segments: $[0, 4]$, $(4, 8]$, $(8, 12]$.
2) **Uniform mass**: Each segment equally gets 4 data points
   - After the 4th observation, place a knot near 2.2.
   - After the 8th observation, place a knot near 8.0.
   - segments: $[0, 2.2]$, $(2.2, 8]$, $(8, 12]$.

Smoothing splines: Formulation

Goal: Estimate a function $g(x)$ that fits observed data (x_i, y_i) well, avoiding overfitting

We minimize a combination of (1) data fidelity and (2) a smoothness penalty:

$$\min_{g \in \mathcal{G}} \left\{ \underbrace{\sum_{i=1}^{n} (y_i - g(x_i))^2}_{\text{RSS}} + \lambda \underbrace{\int (g''(t))^2 \, dt}_{\text{smoothness penalty}} \right\}$$

- The parameter $\lambda \geq 0$ balances data fit vs. smoothness
  - $\lambda = 0$: interpolates all points (leading to a potentially wiggly function)
  - $\lambda \to \infty$: slope is constant, i.e., a straight least squares line
- The solution turns out a *natural cubic spline* with knots at each x_i, but *shrunken* relative to a standard regression spline

In R: The function `smooth.spline()` (in base R) performs smoothing spline fitting

Smoothing splines: Role of the curvature penalty

Why penalize $\int (g''(t))^2 \, dt$?

- The second derivative measures how sharply g bends
- A large $(g'')^2$ indicates "wavy" or erratic behavior
- Minimizing $\int (g'')^2$ forces smaller curvature, yielding a smoother shape

Interpretation of smoothness:

- The penalty $\int (g'')^2$ aims to control "roughness" or high-frequency wiggles
- A more wiggly g has bigger $(g'')^2$, thus a larger penalty

Regression splines vs. smoothing splines:

- **Regression splines**: fix knots/degree and enforce derivative continuity
- **Smoothing splines**: solve a *penalized* least squares problem; knots effectively spread out adaptively to balance data fit vs. smoothness.

Choosing the smoothing parameter λ

Effective degrees of freedom:

- As λ increases from 0 to ∞, the solution transitions from an *interpolation* spline (exactly fitting all data) to a simple *straight line*
- The *effective* degrees of freedom, df_λ, correspondingly decreases from n to 2
  - Degrees of freedom $=$ the number of free parameters
  - The n parameters of smoothing spline are often "shrunk" due to penalty
- df_λ quantifies the spline's complexity, even though we do not explicitly choose knots

Selecting λ (or df_λ):

- Typically use cross-validation
  - Smoothing splines have a handy formula to compute LOOCV errors without re-fitting
- In practice:
  - Pick a small grid of λ-values (or df_λ-values)
  - Compute CV error and select the minimizer

If interested, see [JWHT21, Sec. 7.5.2] for technical details

Pop-up quiz #2: Smoothing splines

Which statement is false about smoothing splines?

A) They solve a penalized least squares problem with a curvature penalty, $\int (g''(t))^2 dt$.

B) As the smoothing parameter $\lambda \to 0$, the spline interpolates all data points, possibly becoming wiggly.

C) As $\lambda \to \infty$, the spline degenerates to a simple linear fit.

D) The effective degrees of freedom always remains fixed at 4 for any smoothing spline fit.

Answer: (D) is false.

The effective degrees of freedom for a smoothing spline varies between n (very wiggly, when $\lambda = 0$) and 2 (almost a straight line, as $\lambda \to \infty$), not a fixed value of 4.

R example: Fitting splines on a toy dataset

Data Setup:

```
set.seed(111)
x <- seq(0, 10, length.out=30)
y <- 2 + 3*sin(x) + rnorm(30, sd=0.3)
df <- data.frame(x, y)
```

Regression spline (`cubic`):

```
# Use 'bs()' from 'splines' package for B-spline basis
library(splines)

# Fit a cubic regression spline with, say, 2 internal knots
fit_spline <- lm(y ~ bs(x, degree=3, knots=c(3,7)), data=df)

# Predictions
x_new <- seq(0,10,length=200)
pred_spline <- predict(fit_spline, newdata=data.frame(x=x_new))
```

R example: Fitting splines on a toy dataset (cont'd)

Natural spline (`cubic by default`):

```
# 'ns()' builds a natural spline basis
fit_ns <- lm(y ~ ns(x, df=5), data=df)
pred_ns <- predict(fit_ns, newdata=data.frame(x=x_new))
```

Smoothing spline:

```
# 'smooth.spline()' solves the penalized objective
fit_smooth <- smooth.spline(x, y, df=6)
pred_smooth <- predict(fit_smooth, x=x_new)$y
```

Note: You can then plot $\hat{y}(x)$ for each model to compare

For more example codes, see [JWHT21, Sec. 7.8.1 & 7.8.2]

Wrap-up: Takeaways

- **Regression splines:**
  - Piecewise polynomials + continuity constraints
  - Truncated power basis for an easy linear-model fit
  - "Natural" splines impose linear boundary conditions to avoid erratic tails
  - Choose knot placement / number of knots via cross-validation
- **Smoothing splines:**
  - A penalized approach balancing data fit vs. curvature
  - The solution is a natural cubic spline with knots at each data point
  - $\lambda \to 0$ yields interpolation; $\lambda \to \infty$ yields a line
  - Effective degrees of freedom: from n to 2
  - Typically choose λ by cross-validation

References

Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani.
An Introduction to Statistical Learning: with Applications in R, volume 112 of *Springer Texts in Statistics*.
Springer, New York, NY, 2nd edition, 2021.