STA 35C: Statistical Data Science III

Lecture 21: Natural Splines & Smoothing Splines

Dogyoon Song

Spring 2026, UC Davis

Announcements

Homework 6 is posted

- Please start early and post questions on Piazza as needed
- Submit your solutions in a **single PDF**; if you submit multiple times, only your latest submission will be graded

Office hours today

- 3:30–4:30 PM at MSB 4220

Agenda

Last time: Basis functions and regression splines

- Basis functions let us fit nonlinear functions using least squares.
- Regression splines are piecewise polynomials joined smoothly at knots.

Today: Controlling spline flexibility

- Quick visual recap: truncated power basis
- Natural splines: improving boundary behavior
- Choosing knots and degrees of freedom
- Smoothing splines: penalized RSS and smoothness

Key idea: Splines are flexible nonlinear regression tools, but we must control their complexity through knots, degrees of freedom, or smoothness penalties.

Recap: Basis functions & regression splines

Basis-function model: Create transformed features $b_1(X), \ldots, b_K(X)$, then fit

$$Y \approx \beta_0 + \sum_{k=1}^{K} \beta_k b_k(X).$$

- The fitted function can be nonlinear in X, but the model is linear in the coefficients β_k, so least squares still applies
- 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)
- *Smoothness constraints*: for a degree-d spline, f and its first $d-1$ derivatives are continuous at each knot
- Degrees of freedom: a degree-d spline with K knots has $(d+1)+K$ parameters

Recap: Why splines instead of high-degree polynomials?

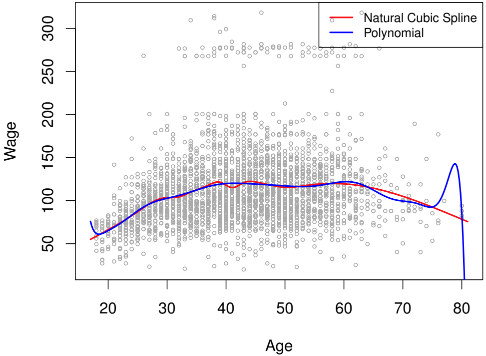


Figure: On the `Wage` data set, a natural cubic spline with 15 degrees of freedom (**blue**) vs. a degree-15 polynomial (**red**). Polynomials can oscillate excessively near the edges, while splines are more stable [JWHT21, Figure 7.7].

- Higher-degree polynomials can be flexible but often exhibit unwanted oscillations
- Splines keep the polynomial degree modest and add flexibility locally through *knots*, often giving more stable fits

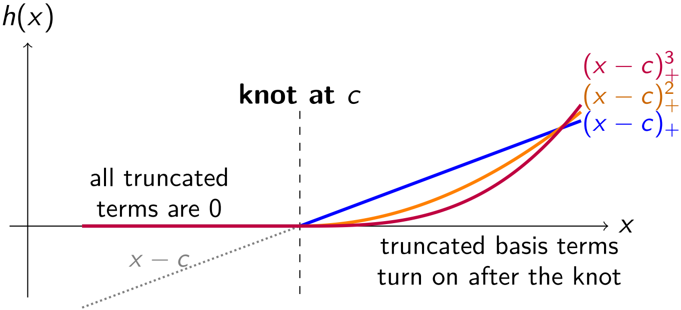
Recap: Truncated power basis

Truncated power basis for a degree-d spline:

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

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

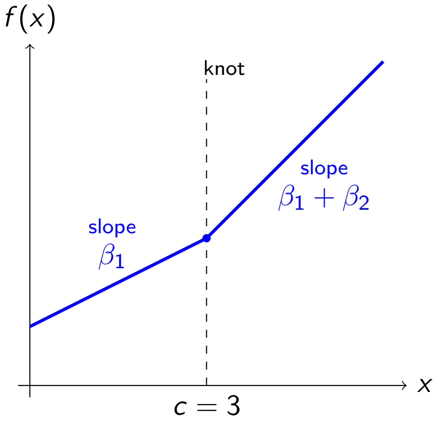
- This representation automatically enforces the desired smoothness at each knot



Visual recap: What does $(x - c)_+$ do?

One-knot linear spline:

$$f(x) = \beta_0 + \beta_1 x + \beta_2 (x - 3)_+, \qquad (u)_+ = \max(u, 0).$$



Equivalently,

$$f(x) = \begin{cases} \beta_0 + \beta_1 x, & x \leq 3, \\ (\beta_0 - 3\beta_2) + (\beta_1 + \beta_2)x, & x > 3. \end{cases}$$

- Before the knot: $(x - 3)_+ = 0$, so the term is inactive.
- After the knot: $(x - 3)_+$ turns on and changes the slope.
- The function remains continuous at $x = 3$.

(Optional) Continuity ensured by truncated power basis

A closer look at 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; see Homework 6

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 natural cubic spline, the curve is constrained to be *linear beyond the boundary knots*; equivalently, the second derivative is 0 in the tails
  - Relative to a cubic spline with the same K interior knots, this reduces the degrees of freedom by 2
  - With K interior knots, a natural cubic spline has $K + 2$ degrees of freedom, compared with $K + 4$ for an ordinary cubic spline
- 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

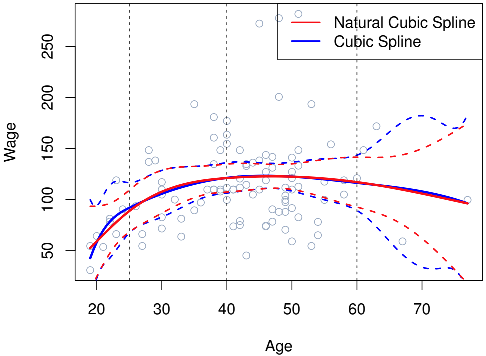


Figure: A cubic spline (**blue**) vs. a natural cubic spline (**red**) on a subset of the `Wage` data; vertical dashed lines show 3 knot locations. Note that natural spline is linear beyond the outer knots [JWHT21, Figure 7.4].

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

Pop-up quiz: Natural splines

Suppose we compare an ordinary cubic spline and a natural cubic spline, both with K interior knots.

Question: Which statement is false?

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

B) A natural cubic spline imposes extra boundary constraints to make the tails linear.

C) The natural boundary constraints increase the degrees of freedom by 2.

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

Answer: C. The natural boundary constraints reduce, rather than increase, flexibility. Relative to a cubic spline with K interior knots, a natural cubic spline has 2 fewer degrees of freedom: $K + 2$ instead of $K + 4$.

Choosing knots and degrees of freedom

Knot placement choices:

- **Uniform width:** knots evenly spaced across the range of X.
- **Uniform mass:** knots placed at quantiles, so each region has roughly the same number of observations.
- **Domain knowledge:** place knots near meaningful breakpoints.

How much flexibility?

- More knots / higher degrees of freedom $\Rightarrow$ more flexible fit.
- Too little flexibility can underfit; too much can overfit.

Common 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 the number of degrees of freedom using cross-validation; knot placement is often chosen by quantiles or domain knowledge

How many knots should we have?

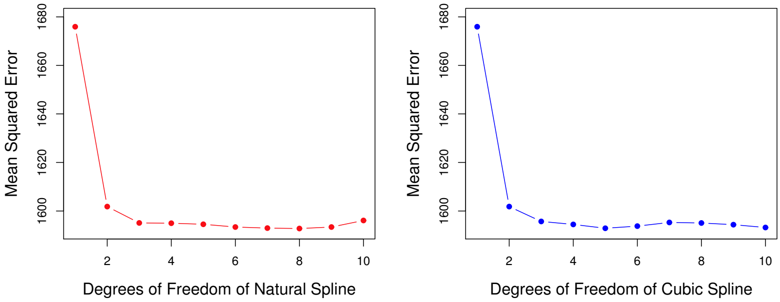


Figure: CV MSEs for different degrees of freedom in splines on the `Wage` data, modeling `wage` vs. `age`. **Left:** natural cubic spline. **Right:** cubic spline [JWHT21, Figure 7.6].

Question: How much flexibility should the spline have?

$$\text{more knots / larger df} \quad \Longleftrightarrow \quad \text{more flexible fit}$$

Common answer: choose degrees of freedom by cross-validation.

Example: 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: $[0, 2.2]$, $(2.2, 8]$, $(8, 12]$.

Pop-up quiz: Placing knots

Suppose X is highly unevenly distributed: many observations are near 0, and only a few are near 10. You want a spline with three intervals.

Question: Which statement is most accurate?

A) Uniform-width knots always give each interval roughly the same amount of data.

B) Quantile-based knots are designed to place roughly the same number of observations in each interval.

C) More knots always improve test performance because the fit becomes more flexible.

D) Knot placement does not matter once least squares is used.

Answer: B. Quantile-based knots adapt to the distribution of X, giving each interval roughly equal sample size. More knots can reduce bias but may increase variance, so flexibility should be controlled, often by cross-validation.

Smoothing splines: Formulation

Goal: Estimate a smooth function $g(x)$ that fits the data well without 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
  - $\lambda = 0$: prioritizes perfect fit to the training data, leading to (wiggly) interpolation
  - $\lambda \to \infty$: heavily penalizes curvature, so the solution approaches a least-squares line
- The solution turns out to be a *natural cubic spline* with knots at each of the observed x_i's, but *shrunken* relative to a standard regression spline (due to penalty)

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 are placed at the observed x_i's, but the penalty controls the effective flexibility.

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
  - Although the solution has *n* knots, the penalty reduces the effective flexibility
- 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: Smoothing splines

Suppose we increase the smoothing parameter λ in a smoothing spline.

Question: Which statement is most accurate?

A) The curve becomes more wiggly, training RSS decreases, and effective degrees of freedom increase.

B) The curve becomes smoother, training RSS generally increases, and effective degrees of freedom decrease.

C) The curve becomes a step function because knots are removed one by one.

D) The irreducible error decreases because the fitted curve is smoother.

Answer: **B.** Larger λ penalizes curvature more heavily, producing a smoother curve with lower effective degrees of freedom. The training RSS generally increases, but test performance may improve if variance is reduced.

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

- **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
  - Effective degrees of freedom: from n to 2
    - $\lambda \to 0$ yields interpolation
    - $\lambda \to \infty$ yields a least-squares line
  - 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.