STA 35C: Statistical Data Science III

Lecture 20: Basis Functions & Regression Splines

Dogyoon Song

Spring 2025, UC Davis

Announcement

Midterm 2 solution and scores are posted online

- You can review your graded exam during tomorrow's discussion section

Grade disputes/adjustments

- If you believe your score should be changed for any question, please email the TA **by noon on Wednesday (May 21)** including:
    - The specific problem(s) you request regrading
    - A clear explanation of why you believe your answer merits more credit (e.g., by pointing out the key elements in your answer that match the official solution)

Where we are so far

We have covered foundational topics in supervised learning:

- **Regression/Classification** basics
- **Resampling methods:** Cross-validation & bootstrap
- **Model selection:** Subset selection and regularization (ridge & lasso)
- **Multiple testing:** FWER and FDR

Next topics:

- **Beyond linear models:**
  - Basis functions & regression splines
- **Unsupervised learning:**
  - Principal component analysis (PCA)
  - Clustering

Today's topics

- **Basis functions**
    - Recall: polynomial regression
    - Step functions
    - How basis functions unify these ideas
- **Regression splines**
    - Piecewise polynomials
    - Smoothness constraints at knots
    - Truncated power basis representation
    - "Natural" splines

Motivation for basis functions: Beyond linear models

Linear regression is powerful but can sometimes be restrictive

- Assumes $Y \approx \beta_0 + \sum_{j=1}^{p} \beta_j X_j$, i.e. a purely linear combination of predictors
- Real data often exhibits more complex, nonlinear relationships

Goal: Extend linear regression to capture nonlinearities while retaining interpretability and tractable estimation

Examples:

- *Polynomial regression*: use $(X, X^2, X^3, \dots)$
- *Step functions*: approximate the regression function by piecewise-constant segments

Today's plan:

- Review polynomial regression & define step functions
- Unify these via *basis functions*
- Introduce *splines* for even more flexible piecewise polynomials

Example 1: Polynomial regression

Polynomial regression: Replace the standard linear model

$$Y = \beta_0 + \beta_1 X + \epsilon$$

with a polynomial:

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \cdots + \beta_d X^d + \epsilon$$

Remarks:

- The coefficients $\beta_1, \ldots, \beta_d$ can be estimated by standard least squares
  - Multiple regression with $X, X^2, \cdots$ treated as distinct predictors
- Typically, a moderate degree d (2–4) is used to avoid overfitting
- Assumes a single global polynomial shape, which can be overly rigid for complex data

Example 1: Polynomial regression (illustration)

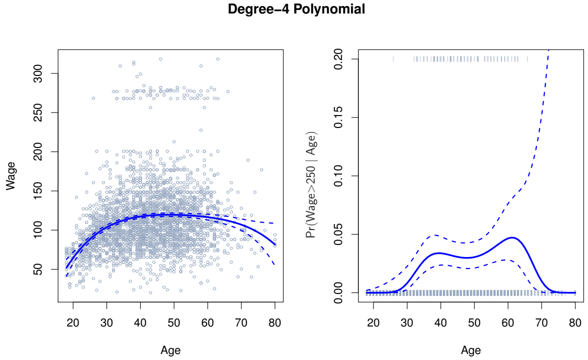

Figure: **Polynomial fit on Wage data.** (*Left*) A degree-4 polynomial of wage on age, with 95% confidence bands. (*Right*) Logistic regression for wage>$250k using a degree-4 polynomial [JWHT21, Figure 7.1].

- **Advantage:** More flexible than a strictly linear model
- **Limitation:** Imposing a global polynomial shape might be too restrictive

Example 2: Step functions

Idea: Partition a continuous variable X into intervals, treating each as a separate "level"

- For cutpoints $c_1 < c_2 < c_3 \ldots$, define indicator functions:

$$C_0(X) := I_{(-\infty, c_1]}(X), \quad C_1(X) := I_{(c_1, c_2]}(X), \quad C_2(X) := I_{(c_2, c_3]}(X) \quad \ldots$$

  - The indicator function

$$I_S(x) = \begin{cases} 1 & \text{if } x \in S, \\ 0 & \text{if } x \notin S. \end{cases}$$

  - Another common convention for the indicator function: $I(x \in S) = I_S(x)$
- The fitted model is piecewise constant:

$$\mathbb{E}[Y \mid X = x] = \beta_0 + \beta_1 \cdot C_1(X) + \beta_2 \cdot C_2(X) + \cdots = \begin{cases} \beta_0 & x \le c_1, \\ \beta_0 + \beta_1 & c_1 < x \le c_2, \\ \beta_0 + \beta_1 + \beta_2 & c_2 < x \le c_3, \\ \vdots & \vdots \end{cases}$$

Example 2: Step functions (illustration)

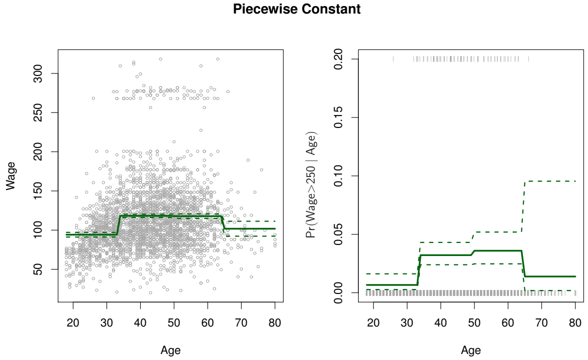

Figure: **Step-function fit on `Wage` data.** (*Left*) Piecewise-constant model for `wage` on `age`, with 95% confidence bands. (*Right*) Logistic regression for `wage`>$250k using a step function [JWHT21, Figure 7.2].

- **Advantage:** Easy to capture abrupt changes ("jumps")
- **Limitations** Not smooth; can be too coarse with few cutpoints

Basis functions: Bridging polynomial, step, and more

Key idea: Transform X to construct new features $\{b_1(X), \ldots, b_K(X)\}$, then fit a linear model in those features:

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

Examples of basis functions:

- *Polynomials*: $b_1(X) = X,\ b_2(X) = X^2, \ldots$
- *Step functions*: $b_1(X) = I(c_1 < X \leq c_2),\ b_2(X) = I(c_2 < X \leq c_3), \ldots$
- *Splines*: piecewise polynomials with continuity constraints
  - Best of both polynomials and step functions

Benefits:

- Still a *linear model* in the transformed features $\{b_k(X)\}$
- Those basis functions can capture nonlinearities more flexibly

Regression splines: Main concept

Want: More flexible than a single polynomial, but smoother than step functions

Idea:

- *Piecewise polynomials* of degree d, joined at *knots* (cutpoints)
- Within each interval between knots, fit a separate polynomial (e.g. cubic)
- Impose *smoothness constraints* at each knot, preventing abrupt jumps or kinks

Example (degree $d = 3$, one knot at c):

$$Y = \begin{cases} \beta_{01} + \beta_{11}X + \beta_{21}X^2 + \ldots \beta_{d1}X^d + \epsilon, & \text{if } x \leq c, \\ \beta_{02} + \beta_{12}X + \beta_{22}X^2 + \ldots \beta_{d2}X^d + \epsilon, & \text{if } x > c. \end{cases}$$

We usually require continuity of the function and its derivatives up to order $d - 1$ at $x = c$

$\rightarrow$ See the next slide for an illustration

Cubic spline illustration: Ensuring smoothness

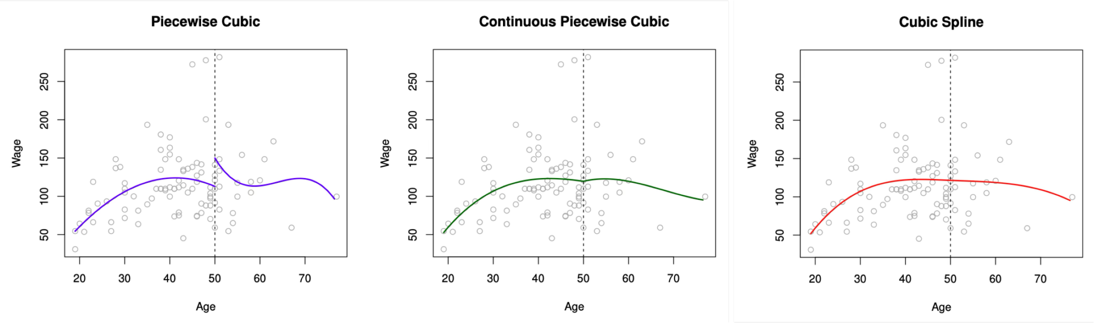

Figure: Various piecewise polynomials fit to a subset of the `Wage` data, with a knot at `age`=50. **Left:** Unconstrained polynomials cause a discontinuity. **Center:** Imposing continuity at the knot eliminates jumps but can form a "kink." **Right:** Additionally constraining continuity of first and second derivatives yields a smooth cubic spline. Excerpted from [JWHT21, Figure 7.3].

Comparing cubic vs. linear splines

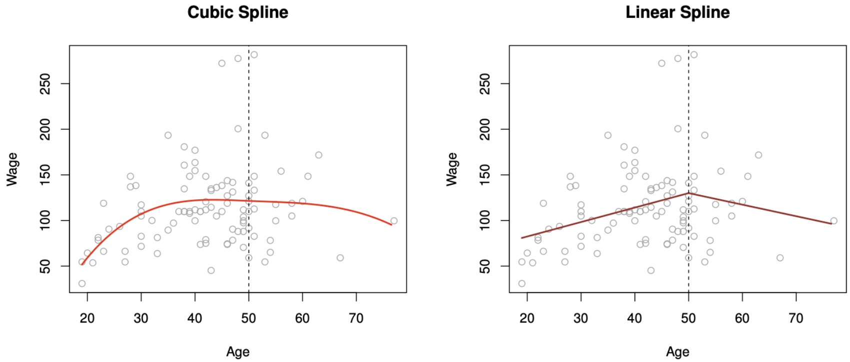

Figure: Again with a knot at age=50 on a subset of Wage. **Left:** A cubic spline, enforcing continuity of function plus its 1st and 2nd derivatives. **Right:** A linear spline, only requiring continuity of function at the knot age=50. Excerpted from [JWHT21, Figure 7.3].

Degrees of freedom for splines

Each additional basis function adds model parameters, increasing flexibility

Piecewise polynomial of degree d with K knots:

- $(d+1)$ polynomial coefficients in each of the $(K+1)$ intervals
  $\implies$ $(K+1) \times (d+1)$ parameters in total *before* smoothness constraints
- Each knot imposes d smoothness constraints (function + $(d-1)$ derivatives)
  - $K \times d$ constraints in total
- The final degrees of freedom $= (K+1)(d+1) - K \cdot d = (d+1) + K$
  - e.g., a cubic spline ($d = 3$) with K knots has $(3+1) + K = K + 4$ parameters

Trade-off:

- Enough degree and knots to capture possible nonlinearities
- But not so many that we overfit or lose interpretability

Spline basis representation: Truncated power basis

Key question: How to systematically fit a piecewise polynomial, enforcing the smoothness constraints at the knots?

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, we can write

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

- This representation automatically encodes smoothness constraints

A toy numerical example: Piecewise 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 $\beta_2 + \beta_3 + \beta_4$.

Cubic spline vs. natural cubic spline

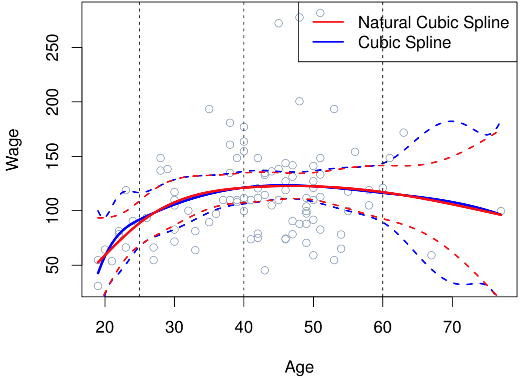

Figure: A cubic spline and a natural cubic spline, with three knots, fit to a subset of the Wage data. The dashed lines denote the knot locations [JWHT21, Figure 7.4].

Natural spline:

- Imposes additional constraints that the function is linear beyond the outermost knots
- Avoids wild oscillations near boundaries
- Often more stable in practice

Wrap-up & next steps

- **Basis functions**: unify polynomial, step, and other expansions for X
  - Allows us to remain in a linear model framework, but with more flexible forms
- **Regression splines**:
  - Piecewise polynomials with continuity at knots
  - Truncated power basis provides a neat representation
  - "Natural" splines add linear constraints in outer intervals
  - Choosing how many knots (and where) to get enough flexibility without overfitting is crucial $\rightarrow$ more on this next time
- **Next lecture**:
  - More on regression splines
  - Smoothing splines

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.