Module 2: Splines and Kernel Methods

Regression Splines, Smoothing Splines

STAT/BIOSTAT 527, University of Washington
Emily Fox
April 10th, 2014

©Emily Fox 2014

Backtrack a bit…

- Instead of just considering input variables $x$ (potentially mult.), augment/replace with transformations = “input features”

- **Linear basis expansions** maintain linear form in terms of these transformations

  $$f(x) = \sum_{m=1}^{M} \beta_m h_m(x)$$

- What transformations should we use?
  - $h_m(x) = x_m \rightarrow$ **linear model**
  - $h_m(x) = x_j^2, \quad h_m(x) = x_j x_k \rightarrow$ **polynomial reg.**
  - $h_m(x) = I(L_m \leq x_k \leq U_m) \rightarrow$ **piecewise constant**
  - ...

©Emily Fox 2014
Piecewise Polynomial Fits

- Again, assume $x$ univariate \textit{(multivariate $x$ later)}

- Polynomial fits are often good locally, but not globally
  - Adjusting coefficients to fit one region can make the function go wild in other regions

- Consider \textit{piecewise polynomial} fits
  - Local behavior can often be well approximated by low-order polynomials

---

Piecewise Polynomial Fits

LIDAR Data Example

From Wakefield book

---
Regression Splines – Linear

- More directly, we can use the **truncated power** basis
  \[ h_1(x) = 1 \]
  \[ h_2(x) = x \]
  \[ h_3(x) = (x - \xi_1)^+ \]
  \[ h_4(x) = (x - \xi_2)^+ \]

- Resulting model:
  \[ f(x) = \beta_0 + \beta_1 x + \beta_2 (x - \xi_1)^+ + \beta_3 (x - \xi_2)^+ \]

- Continuous at the knots because all prior basis functions are contributing to the fit up to any single \( x \)

Regression Splines – Cubic

- Naively, extend as **quadratic**
  \[ f(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 (x - \xi_1)^+ + \beta_4 (x - \xi_1)^2 + \beta_5 (x - \xi_2)^+ + \beta_6 (x - \xi_2)^2 \]
- But, 1\textsuperscript{st} derivative is discontinuous (check this)
- Drop the truncated linear basis:
  \[ f(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + b_1 (x - \xi_1)^2_+ + b_2 (x - \xi_2)^2_+ \]

- Has continuous 1\textsuperscript{st} derivative (check), but not 2\textsuperscript{nd}

- Popular to consider **cubic spline**:
  \[ f(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + b_1 (x - \xi_1)^3_+ + b_2 (x - \xi_2)^3_+ \]

- Has continuous 1\textsuperscript{st} and 2\textsuperscript{nd} derivatives
- Typically people stop here ... *smooth enough*
Cubic Spline Basis and Fit

Cubic Spline Basis and Fit

Cubic Spline Basis and Fit

Cubic Spline as Linear Smoothers

- Cubic spline function with $K$ knots:
  $$f(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \sum_{k=1}^{K} b_k (x - \xi_k)^3$$

- Simply a linear model
  $$\hat{f}(x) = E[Y|X] = \mathbf{C} \cdot \mathbf{\hat{\beta}}$$

- Estimator:
  $$\hat{\mathbf{\beta}} = (\mathbf{C}^T \mathbf{C})^{-1} \mathbf{C}^T \mathbf{y}$$

- Linear smoother:
  $$\hat{f} = \mathbf{C}(\mathbf{C}^T \mathbf{C})^{-1} \mathbf{C}^T \mathbf{y}$$
Natural Cubic Splines

- For polynomial regression, fit near boundaries is erratic.
  - Problem is worse for splines: each is fit locally so no global constraint

- **Natural cubic splines** enforce linearity beyond boundary knots

- Starting from a cubic spline basis, the natural cubic spline basis is:
  \[
  N_1(x) = 1 \quad N_2(x) = x \quad N_{k+2}(x) = d_k(x) - d_{K-1}(x)
  \]

  \[
  d_k(x) = \frac{(x - \xi_k)^3_+ - (x - \xi_K)^3_+}{\xi_K - \xi_k}
  \]

- Derivation

Regression Splines – Summary

- **Definition:**
  
  An order-M spline with knots $\xi_1 < \xi_2 < \cdots < \xi_K$ is a piecewise $M-1$ degree polynomial with $M-2$ continuous derivatives as the knots.

  A spline that is linear beyond the boundary knots is called a natural spline.

- **Choices:**
  - Order of the spline
  - Number of knots
  - Placement of knots

  \[
  \text{requires some thought}
  \]
Return to Smoothing Splines

- Objective:
  \[ \min_{f} \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \int f''(x)^2 \, dx \]

- Solution:
  - **Natural cubic spline**
  - Place knots at every observation location \( x_i \)

- Proof: See Green and Silverman (1994, Chapter 2) or Wakefield textbook

- Notes:
  - Would seem to overfit, but penalty term shrinks spline coefficients toward linear fit
  - Will not typically interpolate data, and smoothness is determined by \( \lambda \)

Smoothing Splines

- Model is of the form:
  \[ f(x) = \sum_{j=1}^{n} N_j(x) \beta_j \]

- Rewrite objective:
  \[ (y - N\beta)^T (y - N\beta) + \lambda \beta^T \Omega N \beta \]

- Solution:
  \[ \hat{\beta} = (N^T N + \lambda \Omega)^{-1} N^T y \text{ as in ridge} \]

- Linear smoother:
  \[ \hat{f} = \frac{N (N^T N + \lambda \Omega) N^T y}{\lambda} \text{ smoothing matrix} \]
  \[ V_\lambda = \text{tr}(L_\lambda) \]
Splines Intro – Summary

- **Regression splines:**
  Fewer number of knots and no regularization

- **Smoothing splines:**
  Knots at every observation and regularization (smoothness penalty) to avoid interpolators

Module 2: Splines and Kernel Methods

B-Splines

STAT/BIOSTAT 527, University of Washington
Emily Fox
April 10th, 2014
Cubic Spline Basis and Fit

- Cubic spline function with $K$ knots:
  \[ f(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \sum_{k=1}^{K} b_k (x - \xi_k)^3 \]

- Using truncated power basis:
  \[ \text{basis on } (0,1) \]

- Number of basis functions = \( M + K \) knots:
  \[ N = \deg \text{ of poly} + 1 \]

- Step 1: Add knots
  \[ f_0 = a \quad f_K+1 = b \]

- Step 2: Define auxiliary knots $\tau_j$
  \[ \tau_1 \leq \tau_2 \leq \cdots \leq \tau_M \leq \xi_0 \]
  \[ \tau_j + M = \xi_j \]
  \[ \xi_{K+1} \leq \tau_{K+M+1} \leq \cdots \leq \tau_{K+2M} \]

B-Splines

- Alternative basis for representing polynomial splines
- Computationally attractive…Non-zero over limited range
- As before:
  \[ \xi_1 < \cdots < \xi_K \]
  \[ (a, b) \]
  \[ \text{Number of basis functions} = (M-1) \text{ knots} \]
  \[ \text{deg of poly} + 1 \]

- Choice is arbitrary.

©Emily Fox 2014
B-Splines

For 1\textsuperscript{st} order B-spline

\[ B_j^1(x) = \begin{cases} 1 & T_j \leq x \leq T_{j+1} \\ 0 & \text{otherwise} \end{cases} \]

Heur basis function

can form any piecewise constantfcn

B-Splines

For 2\textsuperscript{nd} order B-spline

\[ B_j^2(x) = \frac{x-T_j}{T_{j+2}-T_j} B_j^1(x) + \frac{T_{j+2}-x}{T_{j+2}-T_{j+1}} B_{j+1}^1(x) \]

piecewise linear fcn + cont. @ knots

Modify 1\textsuperscript{st} order basis:

\[ B_j^2(x) = \frac{x-T_j}{T_{j+2}-T_j} B_j^1(x) + \frac{T_{j+2}-x}{T_{j+2}-T_{j+1}} B_{j+1}^1(x) \]

Convention: If divide by 0, set basis element to 0
For $m^{th}$ order B-spline, $m=1, \ldots, M$

Modify $(m-1)^{th}$ order basis:

$$B^m_j(x) = \frac{x - \tau_j}{\tau_j + m - \tau_j} B^{m-1}_j(x) + \frac{\tau_j + m - x}{\tau_j + m - \tau_j} B^{m-1}_{j+1}(x)$$

- B-spline bases are non-zero over domain spanned by at most $M+1$ knots
- Only subsets $\{B^m_i \mid i = M - m + 1, \ldots, M + K\}$ are needed for basis of order $m$ with knots $\xi_j$

Cubic Splines as Linear Smoothers

Cubic spline function with $K$ knots:

$$f(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \sum_{k=1}^{K} b_k (x - \xi_k)^3$$

- Simply a linear model

$$f(x) = E(CY) = C \beta$$

Estimator:

$$\hat{\beta} = (C^T C)^{-1} C^T y$$

Linear smoother:

$$\hat{f} = C(C^T C)^{-1} C^T y$$
Cubic B-Splines

- Cubic B-spline with $K$ knots has basis expansion:
  \[ P(x) = \sum_{j=1}^{K+n} B_j^1(x) \beta_j \]

- Simply a linear model
  \[
  \begin{bmatrix}
  B_1^1(x_1) & \ldots & B_{K+n}^1(x_1) \\
  \vdots & \ddots & \vdots \\
  B_1^n(x_n) & \ldots & B_{K+n}^n(x_n)
  \end{bmatrix}
  \begin{bmatrix}
  \beta_1 \\
  \vdots \\
  \beta_{K+n}
  \end{bmatrix}
  = \begin{bmatrix} Y_1 \\
  \vdots \\
  Y_n \end{bmatrix}
  \]

- Computational gain:
  - $n \times (K+n)$ matrix $B$ with many 0’s
  - fewer multiplies (sparse inv.)

Return to Smoothing Splines

- Objective:
  \[
  \min_{f} \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \int f''(x)^2 dx
  \]

- Solution:
  - *Natural cubic spline*
  - Place knots at every observation location $x_i$

- Proof: See Green and Silverman (1994, Chapter 2) or Wakefield textbook

- Notes:
  - Would seem to overfit, but penalty term shrinks spline coefficients toward linear fit
  - Will not typically interpolate data, and smoothness is determined by $\lambda$
Model is of the form: \[ f(x) = \sum_{j=1}^{N} N_j(x) \beta_j \]

Rewrite objective:
\[ (y - N \beta)^T (y - N \beta) + \lambda \beta^T \Omega_N \beta \]

Solution:
\[ \beta = (N^T N + \lambda \Omega_N)^{-1} N^T y \]

Linear smoother:
\[ \hat{\beta} = \left( N^T N + \lambda \Omega_N \right)^{-1} N^T y \]
\[ L_\lambda \quad V_\lambda = \text{tr}(L_\lambda) \]

Using B-spline basis instead:
\[ f(x) = \sum_{j=1}^{n} B_j^y (x) \beta_j \]

Solution:
\[ \hat{\beta} = (B^T B + \lambda \Omega_B)^{-1} B^T y \]

Penalty implicitly leads to natural splines
- Objective gives infinite weight to non-zero derivatives beyond boundary
Spline Overview (so far)

**Smoothing Splines**
- Knots at data points $x_i$
- Natural cubic spline
- $O(n)$ parameters
  - Shrunken towards subspace of smoother functions
- Linear smoothers, for example using natural cubic spline basis:

**Regression Splines**
- $K < n$ knots chosen
- $M^{th}$ order spline = piecewise $M-1$ degree polynomial with $M-2$ continuous derivatives at knots

---

**Reading**
- Hastie, Tibshirani, Friedman: 5.1-5.5 (skipping 5.3), Ch. 5 appendix
- Wakefield: 11.1.1-11.2.6
What you should know…

- **Regression splines**
  - Cubic splines, natural cubic splines, …
  - Interpretation as a linear smoother
  - Degrees of freedom

- **Smoothing splines**
  - Arising from penalized regression setting with smoothness penalty
  - Cubic spline basis with knots at every data point

- **Natural splines**
  - Linear beyond boundary points

- **B-splines**
  - Basis functions with compact support

- **Penalized regression splines**
  - Choose knots as in regression splines, but penalize associated coefficients