Module 2: Splines and Kernel Methods

B-Splines Recap

STAT/BIOSTAT 527, University of Washington
Emily Fox
April 15th, 2014

Cubic Spline Basis and Fit

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

- Basis on $(0,1)$

- Using truncated power basis

- Cubic spline with $K$ knots has $3K - 3$ parameters

- $K-1$ degrees of freedom

- M = 4

- $M-1$ degrees of freedom

- Discontinuous

- Continuous

- Continuous first derivative

- Continuous second derivative

©Emily Fox 2014
B-Splines

- Alternative basis for representing polynomial splines
- Computationally attractive…Non-zero over limited range
- As before:
  - Knots
  - Domain \((a, b)\)
  - Number of basis functions = \(\text{deg. of poly} + 1\)
- Step 1: Add knots \(f_0 = a\) \(f_{K+1} = b\)
- Step 2: Define auxiliary knots \(\tau_j\)

\[
\begin{align*}
\tau_1 & \leq \tau_2 \leq \cdots \leq \tau_M \leq \xi_0 \\
\tau_j + M &= \xi_j \\
\xi_{K+1} & \leq \tau_{K+M+1} \leq \cdots \leq \tau_{K+2M}
\end{align*}
\]

Choice is arbitrary.

For \(m\)th order B-spline, \(m=1, \ldots, M\)

Modify \((m-1)\)th order basis:

\[
B_j^m(x) = \frac{x - \tau_j}{\tau_j + m - 1 - \tau_j} B_j^{m-1} + \frac{\tau_j + m - x}{\tau_j + m - \tau_j + 1} B_{j+1}^{m-1}
\]

- B-spline bases are non-zero over domain spanned by at most \(M+1\) knots
- Only subsets \(\{B_i^m \mid i = M - m + 1, \ldots, M + K\}\) 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
  \[ \hat{f}(x) = \text{E} [Y | X] = c^T \gamma \]

  Estimator:
  \[ \hat{c} = (C^T C)^{-1} C^T Y \]

  Linear smoother:
  \[ \hat{f} = C (C^T C)^{-1} C^T Y \]

Cubic B-Splines

- Cubic B-spline with $K$ knots has basis expansion:
  \[ f(x) = \sum_{j=1}^{K+4} B_j^u (x) \beta_j \]

  Simply a linear model
  \[ \hat{Y} = (B^T B)^{-1} B^T Y \]

  Computational gain:
  An $n \times (K+4)$ matrix $B$ with many 0's
  \[ \rightarrow \text{fewer multiplies (sparse inv.)} \]
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 \( \lambda \)

---

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:

- Linear smoother:
  \[ \hat{\beta} = (N^T N + \lambda \Omega_N)^{-1} N^T y \]
  \[ \mathbf{L}_\lambda = \text{tr}(\mathbf{L}_\lambda) \]
Smoothing Splines

- Model is of the form: \( f(x) = \sum_{j=1}^{n} N_j(x) \beta_j \)
- Using B-spline basis instead: \( f(x) = \sum_{j=1}^{n} B_j(x) \beta_j \)
- 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
- Regression Splines
  - \( K < n \) knots chosen
  - \( M^{th} \) order spline = piecewise \( M-1 \) degree polynomial with \( M-2 \) continuous derivatives at knots
- Linear smoothers, for example using natural cubic spline basis:
Penalized Regression Splines

Module 2: Splines and Kernel Methods

STAT/BIOSTAT 527, University of Washington
Emily Fox
April 15th, 2014

©Emily Fox 2014

Penalized Regression Splines

Alternative approach:
- Use $K < n$ knots
- How to choose $K$ and knot locations?

Option #1:
- Place knots at $n$ unique observation locations $x_i$ and do stepwise
- Issue??

Option #2:
- Place many knots for flexibility
- Penalize parameters associated with knots

Note: Smoothing splines penalize complexity in terms of roughness. Penalized reg. splines shrink coefficients of knots.
Penalized Regression Splines

- General spline model
  - Definition: A **penalized regression spline** is $\hat{\beta}^T h(x)$ with

- Form of resulting spline depends on choice of
  - Basis
  - Penalty matrix
  - Penalty strength

- Still need to choose $K$ and associated locations. RoT (Ruppert et al 2003):
  
  $K = \min\left(\frac{1}{4} \times \# \text{ unique } x_i, 35\right)$
  
  $\xi_k$ at $\frac{k + 1}{K + 2}$th points of $x_i$

PRS Example #1

- Cubic B-spline basis + penalty

- For this penalty, the matrix $D$ is given by

- Leads to

$$\sum_{i=1}^{n_l} (y_i - \beta^T h(x_i))^2 + \lambda \beta^T D \beta$$
**PRS Example #2**

\[ \sum_{i=1}^{n} (y_i - \beta^T h(x_i))^2 + \lambda \beta^T D \beta \]

- B-spline basis + penalty

- For this penalty, the matrix $D$ is given by

- Leads to

**PRS Example #3**

\[ \sum_{i=1}^{n} (y_i - \beta^T h(x_i))^2 + \lambda \beta^T D \beta \]

- Cubic spline using truncated power basis

- + penalty on truncated power coefficients

- For this penalty, the matrix $D$ is given by
A Brief Spline Summary

- **Smoothing spline** – contains $n$ knots
- **Cubic smoothing spline** – piecewise cubic
- **Natural spline** – linear beyond boundary knots
- **Regression spline** – spline with $K < n$ knots chosen
- **Penalized regression spline** – imposes penalty (various choices) on coefficients associated with piecewise polynomial

The # of basis functions depends on
- # of knots
- Degree of polynomial
- A reduced number if a natural spline is considered (add constraints)

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

---

Module 2: Splines and Kernel Methods

Local Polynomial Reg., Kernel Density Estimation

STAT/BIOSTAT 527, University of Washington
Emily Fox
April 15th, 2014
Motivating Kernel Methods

- Recall original goal from Lecture 1:
  - We don't actually know the data-generating mechanism
  - Need an estimator \( \hat{f}_n(x) \) based on a random sample \( Y_1, \ldots, Y_n \), also known as training data

- Proposed a simple model as estimator of \( E[Y|X] \)

Choice #1: k Nearest Neighbors

- Define nbhd of each data point \( x_i \) by the \( k \) nearest neighbors
  - Search for \( k \) closest observations and average these

- Discontinuity is unappealing

From Hastie, Tibshirani, Friedman book
Choice #2: Local Averages

- A simpler choice examines a fixed distance $h$ around each $x_i$
  - Define set: $B_x = \{ i : |x_i - x| \leq h \}$
  - # of $x_i$ in set: $n_x$

- Results in a linear smoother

- For example, with $x_i = \ldots$ and $h = \ldots$
  
  $$L = \ldots$$

More General Forms

- Instead of weighting all points equally, slowly add some in and let others gradually die off

- *Nadaraya-Watson kernel weighted average*

- But what is a *kernel*??
Kernels

- Could spend an entire quarter (or more!) just on kernels
- Will see them again in the Bayesian nonparametrics portion
- For now, the following definition suffices

Example Kernels

- **Gaussian**
  \[ K(x) = \frac{1}{2\pi} e^{-\frac{x^2}{2}} \]

- **Epanechnikov**
  \[ K(x) = \frac{3}{4} (1 - x^2) I(x) \]

- **Tricube**
  \[ K(x) = \frac{70}{81} (1 - |x|^3)^3 I(x) \]

- **Boxcar**
  \[ K(x) = \frac{1}{2} I(x) \]
Nadaraya-Watson Estimator

- Return to Nadaraya-Watson kernel weighted average

\[ \hat{f}(x_0) = \frac{\sum_{i=1}^{n} K(x_0, x_i)y_i}{\sum_{i=1}^{n} K(x_0, x_i)} \]

- Linear smoother:

Example:
- Boxcar kernel →
- Epanechnikov
- Gaussian

- Often, choice of kernel matters much less than choice of \( \lambda \)
Local Linear Regression

- Locally weighted averages can be badly biased at the boundaries because of asymmetries in the kernel

- Reinterpretation:

- Equivalent to the Nadaraya-Watson estimator
- Locally constant estimator obtained from weighted least squares

From Hastie, Tibshirani, Friedman book

Local Linear Regression

- Consider locally weighted linear regression instead
- Local linear model around fixed target \( x_0 \):

- Minimize:

- Return:

- Fit a new local polynomial for every target \( x_0 \)
Local Linear Regression

\[
\min_{\beta x_0} \sum_{i=1}^{n} K_\lambda(x_0, x_i)(y_i - \beta_0 x_0 - \beta_1 x_0 (x_i - x_0))^2
\]

- Equivalently, minimize

- Solution:

Bias calculation:

\[
E[\hat{f}(x_0)] = \sum_i \ell_i(x_0)f(x_i)
\]

- Bias \( E[\hat{f}(x_0)] - f(x_0) \) only depends on quadratic and higher order terms

- Local linear regression corrects bias exactly to 1\textsuperscript{st} order

From Hastie, Tibshirani, Friedman book
Local Polynomial Regression

- Local linear regression is biased in regions of curvature
  - "Trimming the hills" and "filling the valleys"

- Local quadratics tend to eliminate this bias, but at the cost of increased variance

Consider local polynomial of degree $d$ centered about $x_0$

$$P_{x_0}(x; \beta_{x_0}) =$$

Minimize:

$$\min_{\beta_{x_0}} \sum_{i=1}^{n} K_{\lambda}(x_0, x_i)(y_i - P_{x_0}(x; \beta_{x_0}))^2$$

Equivalently:

- Return:
  - Bias only has components of degree $d + 1$ and higher
Local Polynomial Regression

- Rules of thumb:
  - Local linear fit helps at boundaries with minimum increase in variance
  - Local quadratic fit doesn’t help at boundaries and increases variance
  - Local quadratic fit helps most for capturing curvature in the interior
  - Asymptotic analysis → local polynomials of odd degree dominate those of even degree (MSE dominated by boundary effects)

- Recommended default choice: local linear regression

Kernel Density Estimation

- Kernel methods are often used for density estimation (actually, classical origin)

- Assume random sample

- Choice #1: empirical estimate?

- Choice #2: as before, maybe we should use an estimator

- Choice #3: again, consider kernel weightings instead
Kernel Density Estimation

- Popular choice = Gaussian kernel $\Rightarrow$ **Gaussian KDE**

![Graph of Systolic Blood Pressure for CHD group]

**FIGURE 6.13.** A kernel density estimate for systolic blood pressure (for the CHD group). The density estimate at each point is the average contribution from each of the kernels at that point. We have scaled the kernels down by a factor of 10 to make the graph readable.

**6.6 Kernel Density Estimation and Classification**

Kernel density estimation is an unsupervised learning procedure, which historically precedes kernel regression. It also leads naturally to a simple family of procedures for nonparametric classification.

**6.6.1 Kernel Density Estimation**

Suppose we have a random sample $x_1, \ldots, x_N$ drawn from a probability density $f_X(x)$, and we wish to estimate $f_X$ at a point $x_0$. For simplicity we assume for now that $X \in \mathbb{R}$. Arguing as before, a natural local estimate has the form

$$\hat{f}_X(x_0) = \frac{\#x_i \in N(x_0)}{N},$$

(6.21)

where $N(x_0)$ is a small metric neighborhood around $x_0$ of width $\lambda$. This estimate is bumpy, and the smooth Parzen estimate is preferred

$$\hat{f}_X(x_0) = \frac{1}{n\lambda} \sum_{i=1}^{n} K \left( \frac{x - x_i}{\lambda} \right),$$

(6.22)

From Hastie, Tibshirani, Friedman book

**KDE Properties**

$$\hat{p}^\lambda(x) = \frac{1}{n\lambda} \sum_{i=1}^{n} K \left( \frac{x - x_i}{\lambda} \right)$$

- Let’s examine the bias of the KDE

$$E[\hat{p}^\lambda(x)] =$$

- Smoothing leads to biased estimator with mean a smoother version of the true density
- For kernel estimate to concentrate about $x$ and bias $\to 0$, want
KDE Properties

\[ \hat{p}^\lambda(x) = \frac{1}{n\lambda} \sum_{i=1}^n K \left( \frac{x - x_i}{\lambda} \right) \]

- Assuming smoothness properties of the target distribution, it's straightforward to show that
  \[ E[\hat{p}^\lambda(x)] = \]

- In peaks, negative bias and KDE underestimates \( p \)
- In troughs, positive bias and KDE over estimates \( p \)
- Again, “trimming the hills” and “filling the valleys”

- For var \( \to 0 \), require
- More details, including IMSE, in Wakefield book
- Fun fact: There does not exist an estimator that converges faster than KDE assuming only existence of \( p'' \)

Connecting KDE and N-W Est.

- Recall task:
  \[ f(x) = E[Y \mid x] = \int y p(y \mid x) dy \]

- Estimate joint density \( p(x,y) \) with product kernel
  \[ \hat{p}^{\lambda_x, \lambda_y}(x, y) = \]

- Estimate margin \( p(y) \) by
  \[ \hat{p}^{\lambda_x}(x) = \]
Connecting KDE and N-W Est.

Then,
\[
\hat{f}(x) =
\]

- Equivalent to Naradaya-Watson weighted average estimator

Reading

- Hastie, Tibshirani, Friedman: 6.1-6.2, 6.6
- Wakefield: 11.3
What you should know…

- Definition of a kernel and examples
- Nearest neighbors vs. local averages
- Nadarya-Watson estimation
  - Interpretation as local linear regression
- Local polynomial regression
  - Definition
  - Properties/ rules of thumb
- Kernel density estimation
  - Definition
  - Properties
  - Relationship to Nadarya-Watson estimation