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

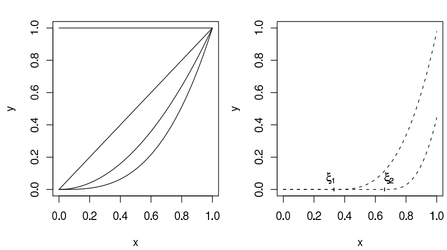
using truncated power basis

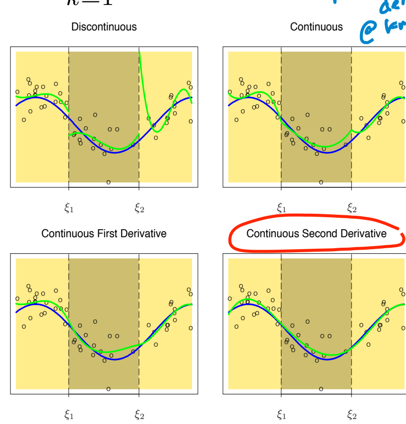
- Cubic spline function with K knots:

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

M=4
M-1 deg poly
M-2 cont. der @ knots

basis on (0,1)





B-Splines

- Alternative basis for representing polynomial splines
- Computationally attractive…Non-zero over limited range
- As before:
  - Knots $\xi_1 < \dots < \xi_K$ ✓
  - Domain (a,b) ✓
  - Number of basis functions = $M + K$

deg. of poly. +1

- Step 1: Add knots $\xi_0 = a$ $\xi_{K+1} = b$
- Step 2: Define auxiliary knots τ_j

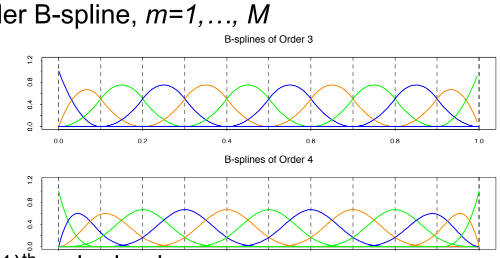
needed to construct basis

choice is arb.

$$\tau_1 \le \tau_2 \le \cdots \le \tau_M \le \xi_0$$
$$\tau_{j+M} = \xi_j$$
$$\xi_{K+1} \le \tau_{K+M+1} \le \cdots \le \tau_{K+2M}$$

B-Splines

- For m^{th} order B-spline, $m=1,\dots, M$



From Hastie, Tibshirani, Friedman book

compact support

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

aux. knots

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

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

for $m=M$ → $M+K$ basis fcns

Cubic Splines as Linear Smoothers

truncated power basis

- 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 $f(x) = E[Y \mid C] = C\gamma$

$$\underset{n \times (4+K)}{C} = \begin{bmatrix} 1 & x_1 & x_1^2 & x_1^3 & (x_1-\xi_1)_+^3 & \cdots & (x_1-\xi_K)_+^3 \\ \vdots & & & & & & \\ 1 & x_n & x_n^2 & x_n^3 & (x_n-\xi_1)_+^3 & \cdots & (x_n-\xi_K)_+^3 \end{bmatrix} \qquad \gamma = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \\ b_1 \\ \vdots \\ b_K \end{bmatrix}$$

not (very) sparse

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

big matrix inversion

- Linear smoother: $\hat{f} = \underbrace{C(C^T C)^{-1} C^T}_{L} y$

Cubic B-Splines as Linear Smoother

- Cubic B-spline with K knots has basis expansion:

$$f(x) = \sum_{j=1}^{K+4} B_j^4(x)\beta_j$$

- Simply a linear model

$$B = \begin{bmatrix} B_1^4(x_1) & \cdots & B_{K+4}^4(x_1) \\ \vdots & & \\ B_1^4(x_n) & \cdots & B_{K+4}^4(x_n) \end{bmatrix} \qquad \gamma = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_{K+4} \end{bmatrix}$$

potentially very sparse bc of compact support of basis function

(lower 4-banded)

$$\hat{\gamma} = (B^T B)^{-1} B^T y$$

- Computational gain:

$n \times (K+M)$ matrix B with many 0's

$\rightarrow$ fewer multiplies (sparse inv.)

Return to Smoothing Splines

- Objective:

$$\min_f \sum_{i=1}^{n}(y_i - f(x_i))^2 + \lambda \int f''(x)^2 dx$$

roughness penalty

- 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 λ

Smoothing Splines

\# of obs.

What we had before

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

natural cubic spline basis

- Rewrite objective:

$$(y - N\beta)^T(y - N\beta) + \lambda\beta^T\Omega_N\beta$$

$n \times n$ matrix

$[N]_{ij} = N_j(x_i)$

$[\Omega_N]_{jk} = \int N_j''(t)N_k''(t)dt$

- Solution: $\hat{\beta} = (N^TN + \lambda\Omega_N)^{-1}N^Ty$

as in ridge

- Linear smoother:

$$\hat{f} = \underbrace{N(N^TN + \lambda\Omega_N)^{-1}N^T}_{L_\lambda} y$$

"smoothing matrix"

$V_\lambda = tr(L_\lambda)$

Smoothing Splines

previously,

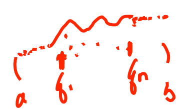
- Model is of the form: $f(x) = \sum_{j=1}^{n} N_j(x)\beta_j$ — K = n knots; n+4 — order M = 4 spline (cubic)

Now,

- Using B-spline basis instead: $f(x) = \sum_{j=1}^{n+4} B_j^4(x)\beta_j$
- Solution: $\hat{\beta} = (B^T B + \lambda \Omega_B)^{-1} B^T y$

B: n x (n+4), lower 4 banded → computational eff.

Ω_B: (nx4) x (nx4)



- Penalty implicitly leads to natural splines
  - Objective gives infinite weight to non-zero derivatives beyond boundary

forces soln to be linear beyond boundary pts

→ natural splines

Spline Overview (so far)

Smoothing Splines

- Knots at data points x_i
- Natural cubic spline
- O(n) parameters
  - Shrunk towards subspace of smoother functions

due to roughness penalty

Regression Splines

- $K < n$ knots chosen
- M^{th} order spline = piecewise M-1 degree polynomial with M-2 continuous derivatives at knots
- no regularization term, but many fewer params

- Linear smoothers, for example using natural cubic spline basis:

$L = N(N^T N + \lambda \Omega_N)^{-1} N^T$ vs. $L = N(N^T N)^{-1} N^T$

L: nxn; $\lambda \Omega_N$: penalty

N: nxK; # params = 4 + K − 4 (4 = M; add'l const.)

Module 2: Splines and Kernel Methods

Penalized Regression Splines

STAT/BIOSTAT 527, University of Washington

Emily Fox

April 15th, 2014

Penalized Regression Splines

- Alternative approach:
  - Use $K < n$ knots few params relative to # of obs.
  - How to choose K and knot locations?
- Option #1:
  - Place knots at n unique observation locations x_i and do stepwise
  - Issue?? 2^n models!
- Option #2:
  - Place many knots for flexibility
  - Penalize parameters associated with knots

  just like ridge/lasso
- Note: Smoothing splines penalize complexity in terms of roughness. Penalized reg. splines shrink coefficients of knots.

Penalized Regression Splines

- General spline model $f(x) = \sum_{j=1}^{J} h_j(x)\beta_j$ ← some spline basis
- Definition: A ***penalized regression spline*** is $\hat{\beta}^T h(x)$ with

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

penalty matrix

- Form of resulting spline depends on choice of
  - Basis $\{h_j(x)\}$
  - Penalty matrix D
  - Penalty strength λ
- Still need to choose *K* and associated locations. RoT (Ruppert et al 2003):

$$K = \min(\frac{1}{4} \times \# \text{ unique } x_i, 35) \qquad \xi_k \text{ at } \frac{k+1}{K+2} th \text{ points of } x_i$$

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

$$\beta = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_{K+4} \end{bmatrix}$$

- Cubic B-spline basis + penalty

$$h_j = B_j^4 \qquad \lambda \int \left(\sum_{j=1}^{K+4} B_j^4(x)'' \beta_j\right)^2 dx$$

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

$$D_{jk} = \int B_j^4(x)'' B_k^4(x)'' dx$$

- Leads to "O'Sullivan splines"

when $K=n$, exactly equivalent to a smoothing spline

knots @ unique x_i

PRS Example #2

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

- B-spline basis + penalty

$$\lambda \sum_{j=1}^{J-1} (\beta_{j+1} - \beta_j)^2$$

- For this penalty, the matrix D is given by

$$D = \begin{bmatrix} 1 & -1 & 0 & \cdots & & \\ -1 & 2 & -1 & 0 & \cdots & \\ 0 & -1 & 2 & -1 & 0 & \cdots \\ & \cdots & & & & \end{bmatrix}$$

- Leads to

"p-splines"

penalizes large changes in coeff. of adj. basis fcns.

$\rightarrow$ smoothing

$\approx$ integrated squared derivative penalty of O'Sullivan splines

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 h_j

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

+ penalty on truncated power coefficients

$$\lambda \sum_k b_k^2 \quad \Leftrightarrow \quad \lambda \|\underline{b}\|_2^2$$

- For this penalty, the matrix D is given by

$$\beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_3 \\ b_1 \\ \vdots \\ b_K \end{bmatrix} \qquad D = \begin{bmatrix} 0 & & & & & \\ & 0 & & & & \\ & & \ddots & & & \\ & & & 0 & & \\ & & & & 1 & \\ & & & & & \ddots \\ & & & & & & 1 \end{bmatrix}$$

β_j's

b_k's

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 $\hat{f}_n(\cdot)$ based on a random sample $Y_{1,}\ldots, Y_n$, also known as ***training data***

- Proposed a simple model as estimator of $E[\,Y \mid X\,]$

$\hat{f}(x) = \text{Avg}(y_i \mid x_i \in \underline{\underline{\text{Nbhd}}}(x))$

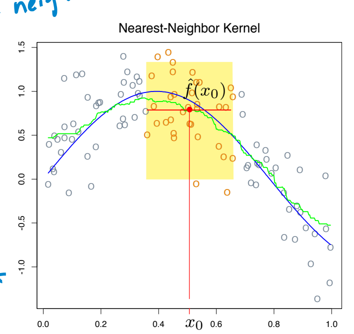
use all obs. y_i in a neighborhood of target x

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

$\hat{f}(x) = \text{Avg}(y_i \mid x_i \in N_k(x))$

k-nearest neighbors



From Hastie, Tibshirani, Friedman book

- Discontinuity is unappealing

neighbors are either in or out
→ disc.

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

$$\hat{f}(x) = \frac{1}{n_x} \sum_{i \in B_x} y_i$$

avg. obs. within distance h

- Results in a linear smoother

$$\hat{f}(x) = \sum_{i=1}^{n} l_i(x) y_i \qquad l_i(x) = \begin{cases} \frac{1}{n_x} & \text{if } |x_i - x| \leq h \\ 0 & \text{ow} \end{cases}$$

- For example, with $x_i = \frac{i}{9}$ and $h = \frac{1}{9}$

$$L = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 & \cdots & \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 & \cdots & \\ 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 & \cdots \end{bmatrix}$$

More General Forms

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

- ***Nadaraya-Watson kernel weighted average***

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

kernel

bandwidth

- 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

$K(\cdot)$ is a kernel if

$K(x) \geq 0 \quad \forall x$

$\int K(u)\,du = 1$

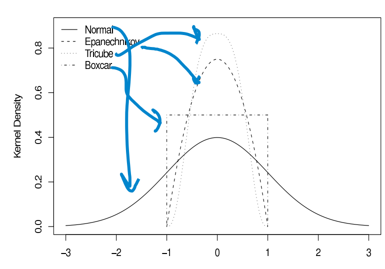
$\int u K(u)\,du = 0 \qquad \sigma_K^2 = \int u^2 K(u)\,du < \infty$

Example Kernels

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

ind. on -1, 1

- *Epanechnikov* $\quad K(x) = \frac{3}{4}(1-x)^2 I(x)$
- *Tricube* $\quad K(x) = \frac{70}{81}(1-|x|^3)^3 I(x)$
- *Boxcar* $\quad K(x) = \frac{1}{2} I(x)$



Nadaraya-Watson Estimator

- Return to Nadaraya-Watson kernel weighted average

$$\hat{f}(x_0) = \frac{\sum_{i=1}^n K_\lambda(x_0, x_i) y_i}{\sum_{i=1}^n K_\lambda(x_0, x_i)}$$

- Linear smoother:

$$\hat{f}(x_0) = \sum_{i=1}^n \underbrace{\frac{K_\lambda(x_0, x_i)}{\sum K_\lambda(x_0, x_i)}}_{l_i(x_0)} y_i = \sum_{i=1}^n l_i(x_0) y_i$$

$$\hat{f} = L_\lambda y$$

$$\nu_\lambda = tr(L_\lambda)$$

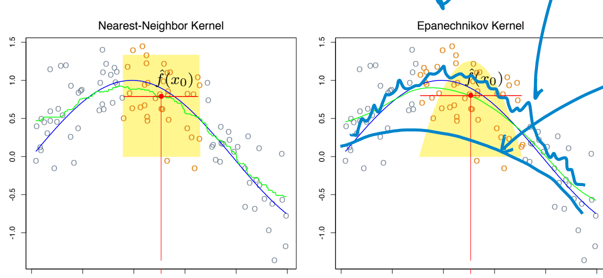
Nadaraya-Watson Estimator

$$\hat{f}(x_0) = \frac{\sum_{i=1}^n K_\lambda(x_0, x_i) y_i}{\sum_{i=1}^n K_\lambda(x_0, x_i)}$$

- Example:
  - Boxcar kernel → local avgs
  - Epanechnikov
  - Gaussian typical

- Often, choice of kernel matters much less than choice of λ

small λ, low bias, high var



large λ high bias, low var

From Hastie, Tibshirani, Friedman book

Local Linear Regression

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

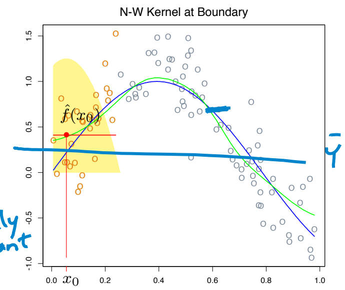
$\hat{f} = \arg\min_a \sum (y_i - a)^2 \quad K\left(\frac{|x_0 - x_i|}{\lambda}\right)$

$\rightarrow \hat{f} = \bar{Y}$

$\hat{f}(x_0) = \arg\min_a \sum w_i(x_0)(y_i - a)^2$ ← restrict to locally constant

local to x_0

$\hat{f}(x_0) = \frac{\sum w_i(x_0) y_i}{\sum w_i(x_0)}$



From Hastie, Tibshirani, Friedman book

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

Local Linear Regression

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

$$\beta_{0x_0} + \beta_{1x_0}(x - x_0)$$

- Minimize:

$$\min_{\beta_{x_0}} \sum_i K_\lambda(x_0, x_i)\left(y_i - (\beta_{0x_0} + \beta_{1x_0}(x_i - x_0))\right)^2$$

- Return: $\hat{f}(x_0) = \hat{\beta}_{0x_0}$ ← fit at x_0

Note: not equivalent to fitting a local constant

- Fit a new local polynomial for *every* target x_0

Local Linear Regression

$$\min_{\beta_{x_0}} \sum_{i=1}^{n} K_\lambda(x_0, x_i)(y_i - \beta_{0x_0} - \beta_{1x_0}(x_i - x_0))^2$$

- Equivalently, minimize

$$(y - X_{x_0}\beta_{x_0})^T W_{x_0} (y - X_{x_0}\beta_{x_0}) \qquad W_{x_0} = \begin{bmatrix} K_\lambda(x_0,x_1) & & \\ & \ddots & \\ & & K_\lambda(x_0,x_n) \end{bmatrix}$$

- Solution:

$$X_{x_0} = \begin{bmatrix} 1 & x_1 - x_0 \\ 1 & x_2 - x_0 \\ \vdots & \vdots \\ 1 & x_n - x_0 \end{bmatrix}$$

$$\hat{\beta}_{x_0} = (X_{x_0}^T W_{x_0} X_{x_0})^{-1} X_{x_0}^T W_{x_0} y$$

$$\hat{f}(x_0) = e_1^T \hat{\beta}_{x_0} \qquad e_1 = [1\ 0 \cdots 0] \text{ grabs out 1st element}$$

$$= \sum l_i(x) y_i$$

Local Linear Regression

$y_i = f(x_i) + \epsilon_i$

- Bias calculation: $\hat{f}(x_0) = \sum_i l_i(x_0) y_i$

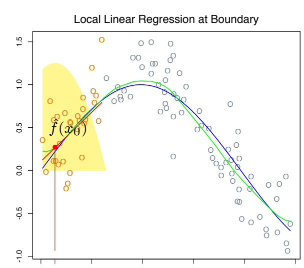
$$E[\hat{f}(x_0)] = \sum_i \ell_i(x_0) f(x_i) \qquad E[y_i]$$

$$= f(x_0) \underbrace{\sum l_i(x_0)}_{=1 \text{ by defn}} + f'(x_0) \underbrace{\sum (x_i - x_0) l_i(x_0)}_{=0 \text{ (can show)}} +$$

$$\frac{f''(x_0)}{2} \sum (x_i - x_0)^2 l_i(x_0) + R \leftarrow \text{higher order terms}$$

$$= f(x_0) + f''(x_0) \cdots$$

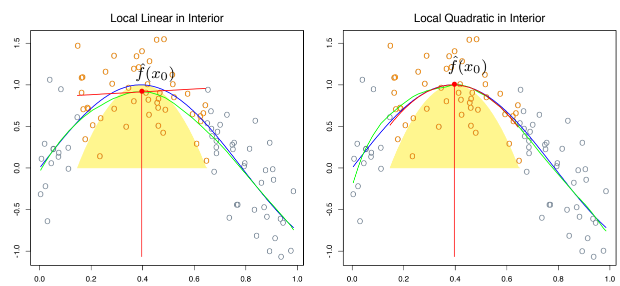
- Bias $E[\hat{f}(x_0)] - f(x_0)$ only depends on quadratic and higher order terms
- Local linear regression corrects bias exactly to 1st order



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



From Hastie, Tibshirani, Friedman book

Local Polynomial Regression

- Consider local polynomial of degree *d* centered about x_0

$$P_{x_0}(x;\beta_{x_0}) = \beta_{0x_0} + \beta_{1x_0}(x-x_0) + \frac{\beta_{2x_0}}{2!}(x-x_0)^2 + \dots + \frac{\beta_{dx_0}}{d!}(x-x_0)^d$$

- 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: $\min (y - X_{x_0}\beta_{x_0})^T W_{x_0} (y - X_{x_0}\beta_{x_0})$

$$X_{x_0} = \begin{bmatrix} 1 & x_1-x_0 & \cdots & \frac{(x_1-x_0)^d}{d!} \\ \vdots & \vdots & & \vdots \\ 1 & x_n-x_0 & \cdots & \frac{(x_n-x_0)^d}{d!} \end{bmatrix}$$

- Return: $\hat{f}(x_0) = \hat{\beta}_{0x_0}$
- 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**

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