Nonlinear Regression (Part 2)

Christof Seiler

Stanford University, Spring 2016, STATS 205

Overview

Last time:

- The bias-variance tradeoff
  - Bias: Estimator cannot explain true function
  - Variance: Estimator changes every time we see a new sample
- The curse of dimensionality

Today:

- Linear Smoothers
  - Local Averages
  - Local Regression
  - Penalized Regression

Nonlinear Regression

- We are given n pairs of observations $(x_1, Y_1), \ldots, (x_n, Y_n)$
- The **response variable** is related to the **covariate**

$$Y_i = r(x_i) + \epsilon_i \qquad \mathrm{E}(\epsilon_i) = 0, i = 1, \ldots, n$$

  with r being the **regression function**
- For now, assume that variance $\mathrm{Var}(\epsilon_i) = \sigma^2$ is independent of x
- The covariates x_i are fixed

Linear Smoothers

- All the nonparametric estimators that we will treat in this class are linear smoothers
- An estimator $\widehat{r}_n$ is a linear smoother if, for each x, there exists a vector $\boldsymbol{l}(x) = (l_1(x), \ldots, l_n(x))^T$ such that

$$\widehat{r}_n(x) = \sum_{i=1}^{n} l_i(x) Y_i$$

- Define the vector of fitted values

$$\boldsymbol{r} = (\widehat{r}_n(x_1), \ldots, \widehat{r}_n(x_n))^T$$

- Then we can write in matrix form ($L_{ij} = l_j(x_i)$)

$$\boldsymbol{r} = LY$$

- The ith row shows weights given to each Y_i in forming the estimate $\widehat{r}_n(x_i)$

Regressogram Estimator

- Suppose that $a \leq x_i \leq b, i = 1, \ldots, n$
- Divide (a, b) into m equally spaced bins denoted by $B_1, B_2, \ldots, B_m$
- Define estimator as

$$\widehat{r_n}(x) = \frac{1}{k_j} \sum_{i: x_i \in B_j} Y_i \quad \text{for} \quad x \in B_j$$

  where k_j is the number of points in B_j
- In this case,

$$l(x)^T = \left(0, 0, \ldots, \frac{1}{k_j}, \ldots, \frac{1}{k_j}, 0, \ldots, 0\right)$$

- This estimator is step function

Regressogram Estimator

- For example, $n = 9, m = 3$

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

Local Averages Estimator

- Fix bandwidth $h > 0$ and let $B_x = \{i : |x_i - x| \leq h\}$
- Let n_x be the number of points in B_x
- Estimator is

$$\widehat{r}_n(x) = \frac{1}{n_x} \sum_{i \in B_x} Y_i$$

- This is a special case of the kernel estimator that we will discuss next
- In this case, $l_i(x) = 1/n_x$ if $|x_i - x| \leq h$ and $l(x) = 0$ otherwise

Local Averages Estimator

- For example, $n = 9, x_i = i/9, h = 1/9$

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

Linear Smoothers

- Rowise weighted average with constrain $\sum_{i=1}^{n} l_i(x) = 1$
- Define the effective degrees of freedom by

$$\nu = \text{tr}(L)$$

- The effective degrees of freedom behave very much like the number of parameters in a linear regression model
- For regressogram example: $L_{ii} = 1$, we have $\nu = n$
- For local averages: $L_{ii} \approx 1/\#\text{neighbors}$, we have $\nu \approx n/\#\text{neighbors}$

Local Regression

- We still use the regression model

$$Y_i = r(x_i) + \epsilon_i, \mathrm{E}(\epsilon_i) = 0, i = 1, \dots, n$$

- But now, we consider weighted averages of Y_i's giving higher weights to points close to x
- One option is **kernel regression estimator** called the Nadaraya–Watson kernel estimator

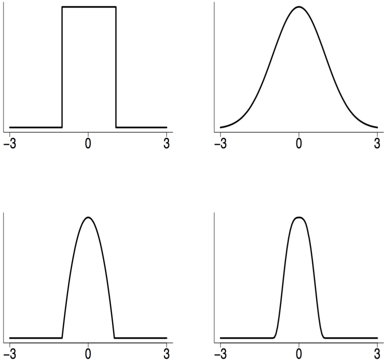
$$\widehat{r}_n(x) = \sum_{i=1}^{n} l_i(x) Y_i$$

with kernel K and weights $l_i(x)$ given by

$$l_i(x) = \frac{K\left(\frac{x - x_i}{h}\right)}{\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right)}$$

Local Regression

- For example, the Gaussian $K(x) = \frac{1}{2}e^{-x^2/2}$
- Think of them as basis functions anchored at observation locations x_i



Source: Wassermann (2006)

Local Regression

- For points $x_1, \ldots, x_n$ drawn from some density f
- Let $h \to 0, nh \to \infty$
- The bias-variance tradeoff for the Nadaraya–Watson kernel estimator

$$R(\widehat{r}_n, r) \approx \frac{h^4}{4} \text{ Bias}^2 + \frac{1}{nh} \text{ Variance}$$

- Depends on first and second derivatives of the density f

$$\text{Bias}^2 = \left(\int x^2 K(x) dx \right)^2 \int \left(r''(x) + 2r'(x) \frac{f'(x)}{f(x)} \right)^2 dx$$

- The term $2r'(x)\frac{f'(x)}{f(x)}$ is called **design bias**
- It depends on the distribution of the points $x_1, \ldots, x_n$

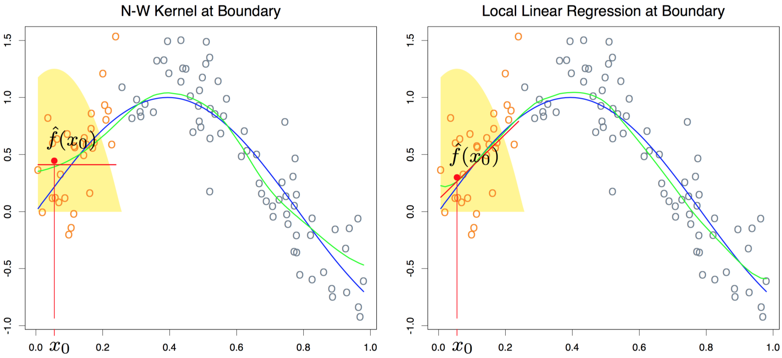
Local Regression

- The bias term

$$2r'(x)\frac{f'(x)}{f(x)}$$

  has two properties:
  - it is large if $f'(x)$ is non-zero and
  - it is large if $f(x)$ is small
- The dependence of the bias on the density $f(x)$ is called design bias
- It can also be shown that the bias is large if x is close to the boundary of the support of p
- These biases can be reduced by by using a refinement called **local polynomial regression**

Local Polynomials



Source: Hastie, Tibshirani, Friedman (2009)

Local Polynomials

- Define function $w_i(x) = K((x_i - x)/h)$ and choose $a \equiv \widehat{r}_n(x)$

$$\sum_{i=1}^{n} w_i(x)(Y_i - a)^2$$

- Take derivative with respect to $\boldsymbol{a}$ and set to zero

$$a = \frac{\sum_{i=1}^{n} w_i(x) y_i}{\sum_{i=1}^{n} w_i(x)}$$

- Thus the kernel estimator is a locally constant estimator, obtained from locally weighted least squares

Local Polynomials

- What if we local polynomial of degree p instead of a local constant?
- Let x be some fixed value at which we want to estimate $r(x)$
- For values u in a neighborhood of x, define the polynomial

$$P_x(u; \boldsymbol{a}) = a_0 + a_1(u - x) + \frac{a_2}{2!}(u - x)^2 + \cdots + \frac{a_p}{p!}(u - x)^p$$

- We approximate the regression function $r(u)$ in neighborhood u

$$r(u) \approx P_x(u; \boldsymbol{a})$$

- Estimate $\boldsymbol{a} = (a_0, \ldots, a_p)^T$ by taking the gradient with respect to $\boldsymbol{a}$ and setting to zero

$$\sum_{i=1}^{n} w_i(x_i)(Y_i - P_x(x_i; \boldsymbol{a}))^2$$

Local Polynomials

- First construct matrices

$$X_x = \begin{bmatrix} 1 & x_1 - x \\ 1 & x_2 - x \\ \vdots & \vdots \\ 1 & x_n - x \end{bmatrix}, W_x = \begin{bmatrix} w(x_1) & 0 & \cdots & 0 \\ 0 & w(x_1) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w(x_n) \end{bmatrix}$$

- Then rewrite in matrix form

$$\sum_{i=1}^{n} w_i(x_i)(Y_i - P_x(x_i; \boldsymbol{a}))^2 = (Y - X_x\boldsymbol{a})^T W_x (Y - X_x\boldsymbol{a})$$

- Take the gradient with respect to $\boldsymbol{a}$ and set to zero

$$\boldsymbol{a} = (X_x^T W_x X_x)^{-1} X_x^T W_x Y$$

Local Polynomials

- $p = 1$ is most popular case, this is called **local linear regression**
- $p = 0$ gives back kernel estimator
- This linear smoother

$$\boldsymbol{a} = (X_x^T W_x X_x)^{-1} X_x^T W_x Y = LY$$

and we can choose the bandwith with cross-validation

Local Polynomials

- Comparing the bias-variance tradeoff

$$R(\widehat{r}_n, r) \approx \frac{h^4}{4}\,\mathrm{Bias}^2 + \frac{1}{nh}\,\mathrm{Variance}$$

- for Nadaraya–Watson kernel estimator (depends on first and second derivatives of the density f)

$$\mathrm{Bias}^2 = \left(\int x^2 K(x)dx\right)^2 \int \left(r''(x) + 2r'(x)\frac{f'(x)}{f(x)}\right)^2 dx$$

- and local linear estimator (no dependence on the density f)

$$\mathrm{Bias}^2 = \left(\int x^2 K(x)dx\right)^2 \int r''(x)^2 dx$$

Penalized Regression

- An alternative to local averaging
- Minimize the penalized sums of squares

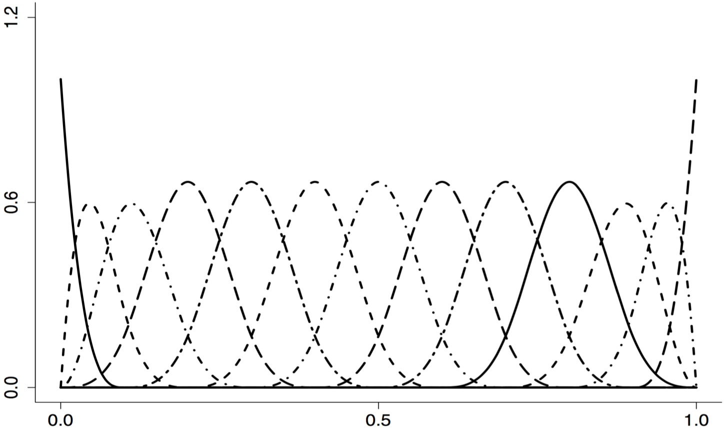
$$M(\lambda) = \sum_{i=1}^{n} (Y_i - \widehat{r}_n(x_i))^2 + \lambda J(r)$$

- Roughness penalty, for example

$$J(r) = \int (r''(x))^2 dx$$

- The parameter λ controls the trade-off between fit
- For $\lambda = 0$ is interpolating function
- As $\lambda \to \infty$ is least squares line
- Between $0 < \lambda < \infty$ $\widehat{r}_n$ are splines (piecewise polynomials)

Penalized Regression



Source: Wasserman (2006)

Penalized Regression

- **Theorem:** The function $\widehat{r}_n(x)$ that minimizes $M(\lambda)$ with penalty from previous slide is a natural cubic spline with knots at the data points. The estimator $\widehat{r}_n$ is called a smoothing spline.
- This theorem doesn't give an explict form of $\widehat{r}_n$
- However we can build an explicit basis using **B-splines**

$$\widehat{r}_n(x) = \sum_{j=1}^{N} \widehat{\beta}_j B_j(x)$$

- where $B_1, \ldots, B_N$ are a basis for B-splines with $N = n + 4$
- Thus, we only need to find the coefficients $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_N)^T$

Penalized Regression

- So we take the derivative and set to zero

$$(Y - B\beta)^T(Y - B\beta) + \lambda\beta^T\Omega\beta$$

  with $B_{ij} = B_j(X_i)$ and $\Omega_{jk} = \int B_j''(x)B_k''(x)$
- and find

$$\widehat{\beta} = (B^TB + \lambda\Omega)^{-1}B^TY$$

- This is another linear smoother

$$\boldsymbol{r} = B(B^TB + \lambda\Omega)^{-1}B^TY = LY$$

  with fittes values $\boldsymbol{r}$ a smooth version of original observations Y

References

- Wassermann (2006). All of Nonparametric Statistics
- Hastie, Tibshirani, Friedman (2009). The Elements of Statistical Learning