

## Lec 11 : Local Polynomial & Spline Regression

---

Yanjan Han

---

Nov 28, 2023

---



Last lecture: density estimation: KDE, k-NN  
regression: Nadaraya-Watson

This lecture: combine with **linear regression**!

Recap of nonparametric regression: given  $(x_1, y_1), \dots, (x_n, y_n)$ , estimate  
 $f(x) = \mathbb{E}[Y|X=x]$

Last lecture: if  $|f'(x)| \leq L$ , for suitable kernel  $K$  & optimal bandwidth  $h$ ,  
the Nadaraya-Watson estimator  $\hat{f}^{NW}$  achieves  
 $MSE(\hat{f}^{NW}(x_0)) = \mathbb{E}[(\hat{f}^{NW}(x_0) - f(x_0))^2] = O(n^{-\frac{1}{2}})$

Question: what happens if  $|f^{(k)}(x)| \leq L$  for some  $k \geq 2$ ?

Natural estimator:  $\hat{f}(x_0) = \sum_{i=1}^n \underbrace{w(x_0, x_i)}_{\text{weight of data point } x_i \text{ for the new point } x_0} y_i$

We want:

- 1)  $\sum_{i=1}^n w(x_0, x_i) = 1$  (weights sum to 1)
- 2)  $\sum_{i=1}^n (x_0 - x_i)^l w(x_0, x_i) = 0, \quad l = 1, 2, \dots, k-1$   
(analogy to  $\int x^k K(x) dx = 0$  in KDE, with  $k=2$ )
- 3)  $w(x_0, x_i) = 0$  if  $|x_0 - x_i| > C_0 h$  (bandwidth)
- 4)  $\sum_{i=1}^n |w(x_0, x_i)| \leq C_1, \quad \max_{1 \leq i \leq n} |w(x_0, x_i)| \leq \frac{C_2}{nh}$   
(think of  $w(x_0, x_i) = \frac{1}{n} K_h(x_0 - x_i) = \frac{1}{nh} K(\frac{x_0 - x_i}{h})$ )

Implication:  $\sum_{i=1}^n w(x_0, x_i) p(x_i) = p(x_0)$ .  $\forall$  polynomial  $p$  with  $\deg(p) \leq k-1$ .

Pf. Write  $p(x) = a_{k-1}(x-x_0)^{k-1} + \dots + a_1(x-x_0) + a_0$   
 $\Rightarrow \sum_{i=1}^n w(x_0, x_i) p(x_i) = 0 + 0 + \dots + 0 + a_0 = a_0$

Take  $x = x_0 \Rightarrow a_0 = p(x_0)$

□

Estimator analysis (assuming  $|f^{(k)}(x_0)| \leq L$ ,  $\text{Var}(y_i | x_i) \leq \sigma^2$ )

$$\text{Var}(\hat{f}(x_0)) \leq \sigma^2 \sum_{i=1}^n w(x_0, x_i)^2 \leq \sigma^2 \sum_{i=1}^n |w(x_0, x_i)| \cdot \frac{C_2}{nh} \leq \frac{C_1 C_2 \sigma^2}{nh} = O\left(\frac{1}{nh}\right)$$

$$\begin{aligned} |\text{Bias}(\hat{f}(x_0))| &= \left| \sum_{i=1}^n w(x_0, x_i) f(x_i) - f(x_0) \right| \\ &= \min_{p: \deg(p) \leq k-1} \left| \sum_{i=1}^n w(x_0, x_i) (f(x_i) - p(x_i)) - (f(x_0) - p(x_0)) \right| \\ &\leq \min_{p: \deg(p) \leq k-1} \sum_{i=1}^n |w(x_0, x_i)| \cdot \max_{x: |x-x_0| \leq C_0 h} |f(x) - p(x)| \\ &\leq C_1 \cdot \min_{p: \deg(p) \leq k-1} \max_{x: |x-x_0| \leq C_0 h} |f(x) - p(x)| \end{aligned}$$

(best polynomial approx. err. of  $f$  on  $[x_0 - C_0 h, x_0 + C_0 h]$ )

Choosing  $p(x) = f(x_0) + f'(x_0)(x-x_0) + \dots + \frac{f^{(k-1)}(x_0)}{(k-1)!} (x-x_0)^{k-1}$

(Taylor approx. polynomial)

$$\Rightarrow |\text{Bias}(\hat{f}(x_0))| \leq C_1 \cdot \frac{L(C_0 h)^k}{k!} = O(h^k)$$

$$h = h_n = n^{-\frac{1}{2k+1}}$$

$$\text{MSE}(\hat{f}(x_0)) = \text{Bias}(\hat{f}(x_0))^2 + \text{Var}(\hat{f}(x_0)) = O(h^{2k} + \frac{1}{nh}) = O(n^{-\frac{2k}{2k+1}})$$

Q: How to construct the weights  $w(x_0, x_i)$ ?

A: Local polynomial regression!

Local polynomial regression.

Given kernel  $K(\cdot)$  and bandwidth  $h > 0$ :

$$(\hat{\theta}_0, \dots, \hat{\theta}_{k-1}) = \underset{(\theta_0, \dots, \theta_{k-1})}{\text{argmin}} \frac{1}{2} \sum_{i=1}^n (y_i - \theta_0 - \theta_1 x_i - \dots - \theta_{k-1} x_i^{k-1})^2 K_h(x_0 - x_i)$$

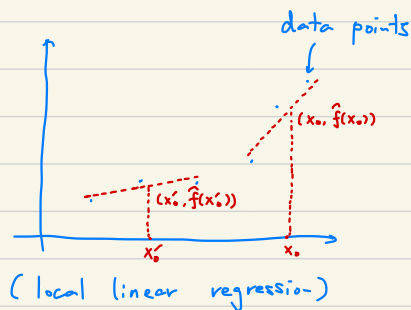
Then estimate  $f(x_0)$  by  $\hat{f}(x_0) = \hat{\theta}_0 + \hat{\theta}_1 x_0 + \dots + \hat{\theta}_{k-1} x_0^{k-1}$ .

Special cases :

$k=1$  : reduce to Nadaraya-Watson:

$$\hat{f}(x_0) = \hat{\theta}_0 = \frac{\sum_{i=1}^n K_h(x_0 - x_i) y_i}{\sum_{i=1}^n K_h(x_0 - x_i)}$$

$k=2$  : local linear regression



Computation of  $\hat{\theta} = (\hat{\theta}_0, \dots, \hat{\theta}_{k-1})$  : weighted linear regression

F.O.C wrt  $\theta_j$  :  $\sum_{i=1}^n x_i^j (y_i - \hat{\theta}_0 - \hat{\theta}_1 x_i - \dots - \hat{\theta}_{k-1} x_i^{k-1}) K_h(x_0 - x_i) = 0$

Define :  $X = \begin{bmatrix} 1 & x_1 & \dots & x_1^{k-1} \\ 1 & x_2 & \dots & x_2^{k-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & \dots & x_n^{k-1} \end{bmatrix} \in \mathbb{R}^{n \times k}$  ,  $D = \begin{bmatrix} K_h(x_0 - x_1) & & \\ & \ddots & \\ & & K_h(x_0 - x_n) \end{bmatrix} \in \mathbb{R}^{n \times n}$

$\Rightarrow$  Matrix form :  $X^T D X \hat{\theta} = X^T D y$  , or

$\hat{\theta} = (X^T D X)^{-1} X^T D y$  ( $D = I_n$  : OLS)

Verification of moment conditions

$\hat{f}(x_0) = [1 \ x_0 \ \dots \ x_0^{k-1}]^T \hat{\theta} = [1 \ x_0 \ \dots \ x_0^{k-1}] (X^T D X)^{-1} X^T D y$

$\Rightarrow [w(x_0, x_1) \ \dots \ w(x_0, x_n)] = [1 \ x_0 \ \dots \ x_0^{k-1}] (X^T D X)^{-1} X^T D$

Therefore, for  $p(x) = a_0 + a_1 x + \dots + a_{k-1} x^{k-1} = [1 \ x \ \dots \ x^{k-1}] [a_0 \ a_1 \ \dots \ a_{k-1}]^T$ ,

$$\sum_{i=1}^n w(x_0, x_i) p(x_i) = [w(x_0, x_1) \ \dots \ w(x_0, x_n)] \begin{bmatrix} p(x_1) \\ \vdots \\ p(x_n) \end{bmatrix} = [1 \ x_0 \ \dots \ x_0^{k-1}] \underbrace{(X^T D X)^{-1} X^T D}_{= I_k} \cdot X \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{k-1} \end{bmatrix}$$

$= a_0 + a_1 x_0 + \dots + a_{k-1} x_0^{k-1} = p(x_0)$  , as desired.

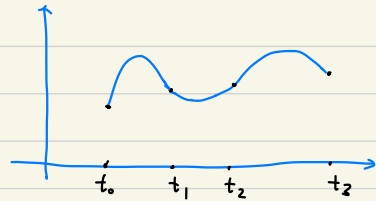
## A different estimation procedure: splines

Defn: a degree- $d$  **spline** with **knots**  $t_0 < t_1 < \dots < t_m$  is a function  $S$   $[t_0, t_m] \rightarrow \mathbb{R}$  such that:

- 1)  $S$  is a deg- $d$  polynomial on  $[t_{i-1}, t_i]$ ,  $\forall i=1, \dots, m$ ;
- 2)  $S \in C^{d-1}$  is  $(d-1)$ -times continuously differentiable  
(i.e. the first  $(d-1)$  derivatives match at the midpoints)

Example:  $d=1$ : piecewise linear

$d=2$  (quadratic spline):



Property: the basis functions for degree- $d$  splines at knots  $t_0 < t_1 < \dots < t_m$  are

$$\begin{aligned} & \{1, x, x^2, \dots, x^d, (x-t_1)_+^d, \dots, (x-t_{m-1})_+^d\} \\ &= \begin{cases} (x-t_i)_+^d & \text{if } x > t_i \\ 0 & \text{o.w.} \end{cases} \end{aligned}$$

(Verification via degree-of-freedom:

$$\begin{aligned} \# \text{ parameters} &= m(d+1) \\ \# \text{ constraints} &= (m-1)d \end{aligned} \Rightarrow \# \text{ basis functions} = m(d+1) - (m-1)d = m + d$$

## Regression splines

Given knots  $(t_0, \dots, t_m)$ , model the regression function  $f$  as a degree- $d$  spline at the given knots:

$$f(x) = a_0 + a_1 x + \dots + a_d x^d + b_1 (x-t_1)_+^d + \dots + b_{m-1} (x-t_{m-1})_+^d.$$

How to compute  $(a_0, \dots, a_d, b_1, \dots, b_{m-1})$ :

$$\min_{\theta = (a_0, \dots, a_d, b_1, \dots, b_{m-1})} \frac{1}{2} \sum_{i=1}^n (y_i - f_\theta(x_i))^2 \Rightarrow \text{An OLS with feature vector } (1, x_i, \dots, x_i^d, (x_i - t_1)_+, \dots, (x_i - t_{m-1})_+)$$

for  $i$ -th observation

Sketch of analysis. Use evenly spaced knots  $t_i = \frac{i}{m}$ ,  $i = 0, 1, \dots, m$

Bias: if  $|f^{(k)}(x)| \leq L$ , for  $d = k$  there is a spline  $f_0$  s.t.

$$|f(x) - f_0(x)| = O(m^{-k}) \text{ for all } x \Rightarrow \text{Bias} = O(m^{-k})$$

Variance: there are  $m+d$  unknowns in the OLS  $\Rightarrow \text{Variance} = O(\frac{m}{n})$

$$\text{MSE} = \text{Bias}^2 + \text{Variance} = O(m^{-2k} + \frac{m}{n}) = O(n^{-\frac{2k}{2k+1}})$$

$m = m_n = n^{\frac{1}{2k+1}}$

Smoothing splines  $(\{t_0, \dots, t_m\} = \{x_1, \dots, x_n\})$

Cubic smoothing spline with regularization  $\lambda > 0$ :

$$\hat{f} = \arg \min_f \sum_{i=1}^n (y_i - f(x_i))^2 + \lambda \int_0^1 f''(x)^2 dx$$

Penalty on the smoothness of  $f$ .

Theorem.  $\hat{f}$  is a cubic spline with knots  $\{x_1, \dots, x_n\}$ .

Pf (optional). Suffice to show that (natural) cubic spline  $f$  minimizes

$$J(f) = \int_0^1 f''(x)^2 dx \quad \text{s.t.} \quad f(x_i) = z_i, \quad \forall i = 1, \dots, n.$$

Let  $g$  be any function w/  $g(x_i) = z_i$ , and  $h = g - f$ . Then

$$J(g) - J(f) = 2 \int_0^1 f''(x) h''(x) dx + J(h) = -2 \int_0^1 f'''(x) h'(x) dx + J(h)$$

$$= -2 \sum_{i=1}^{n-1} f'''(x_i) \int_{x_i}^{x_{i+1}} h'(x) dx + J(h) \quad (\text{Integration by parts})$$

$$= -2 \sum_{i=1}^{n-1} f'''(x_i) (h(x_{i+1}) - h(x_i)) + J(h) = J(h) \geq 0.$$

(f''' piecewise constant)  $= h(x_{i+1}) - h(x_i) = 0$  as  $f(x_i) = g(x_i) = z_i$   
 $f(x_{i+1}) = g(x_{i+1}) = z_{i+1}$

overparametrized as  $n+2 > n$

Computation. Let  $\{f_1(x), \dots, f_{n+2}(x)\}$  be the basis functions of cubic splines with knots  $\{x_1, \dots, x_n\}$ . then

$$\hat{f}(x) = \hat{\theta}_1 f_1(x) + \dots + \hat{\theta}_{n+2} f_{n+2}(x).$$

Matrix notation:

$$X = [f_j(x_i)] \in \mathbb{R}^{n \times (n+2)}, \quad W = [W_{ij} = \int f_i'' f_j'' dx] \in \mathbb{R}^{(n+2) \times (n+2)}$$

Then 
$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \quad \|y - X\theta\|_2^2 + \lambda \theta^T W \theta = (X^T X + \lambda W)^{-1} X^T y$$
  
(Ridge regression)

Performance. If  $|f''(x)| \leq L$ , for suitably chosen  $\lambda$ ,

$$\|\hat{f} - f\|_2^2 = O(n^{-4/5}) \quad (\text{pf omitted})$$

Optional: Multivariate adaptive regression spline (MARS)

(See J.H. Friedman, "Multivariate Adaptive Regression Splines", AoS 1991)

Question: how to choose the knots? can we do it adaptively?

Idea of MARS:  $\hat{f}(x) = \sum_{j=1}^J \theta_j B_j(x)$

1)  $J$  changes over time

2) forward pass: recursively, using a greedy algorithm, find

2.1) a data point  $i \in \{1, \dots, n\}$ ;

2.2) an existing basis function  $j \in \{1, \dots, J\}$ ;

2.3) a dimension  $k \in \{1, \dots, d\}$

update  $J \rightarrow J+1$ ,  $B_i(x) \rightarrow \{B_i(x)(x^{(k)} - x_i^{(k)})_+, B_i(x)(x^{(k)} - x_i^{(k)})_-\}$

$$(x_+ = \max\{x, 0\}, \quad x_- = \max\{-x, 0\})$$

3) backward pass: prunes the model to prevent overfitting