6. Kernel Smoothing Methods

오영민

Index

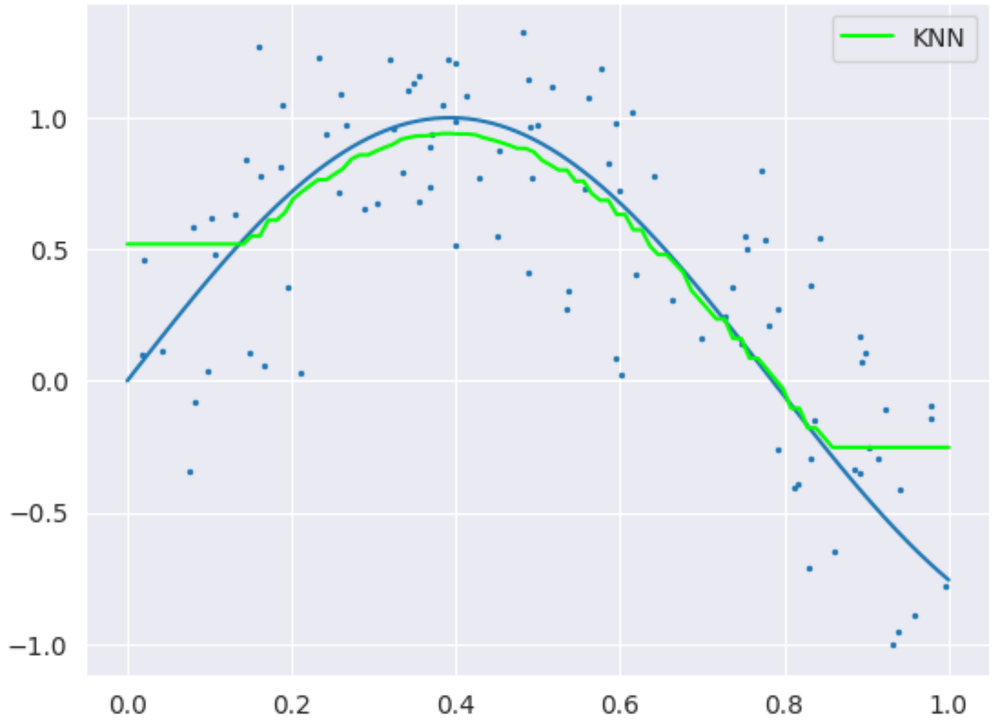
Introduction

- Using only those observations close to the target point x_0 to fit the simple model s.t. resulting estimated $\hat{f}$ is smooth in $\mathbb{R}^p$ with weighting function of kernel $K_\lambda(x_0 \,.\, x_i)$.
- This memory-based methods require in principle little or no training

One-Dimensional Kernel Smoothers

- **k-nearest-neighbor average**: $\hat{f}(x) = \text{Ave}(y_i \,|\, x_i \in N_k(x))$ as an estimate of $\mathbb{E}[Y \,|\, X = x]$

- Ex: In each panel 100 pairs $Y_i = \sin(4X) + \epsilon,\ X \sim U[0,1],\ \epsilon \sim \mathcal{N}(0,1/3)$



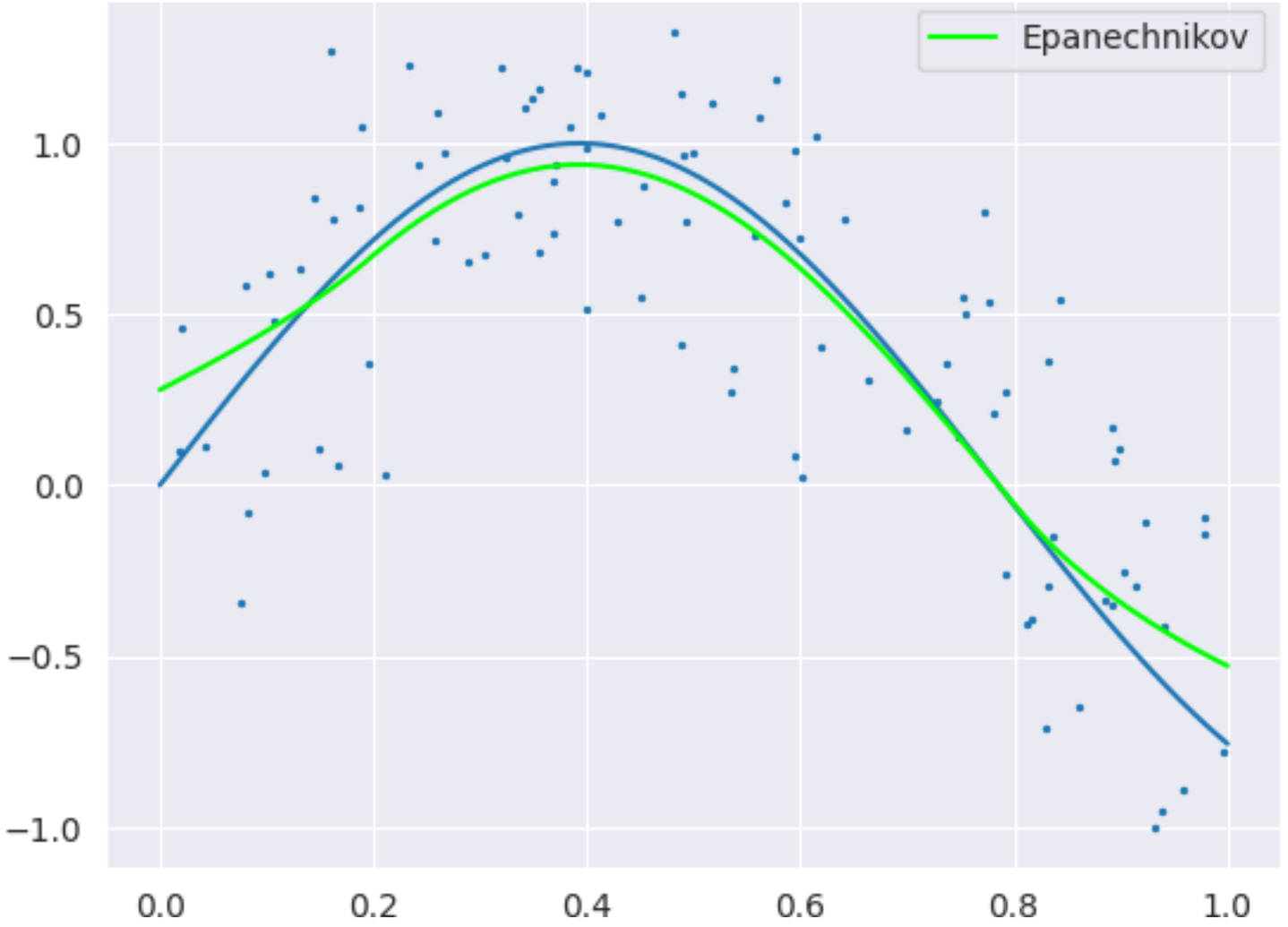
One-Dimensional Kernel Smoothers

- **Nadaraya-Watson kernel-weighted average**: $\hat{f}(x_0) = \frac{\sum_{i=1}^{N} K_\lambda(x_0, x_i) y_i}{\sum_{i=1}^{N} K_\lambda(x_0, x_i)}$

- **Kernel**:
  - Epanechnikov quadratic kernel: $K_\lambda(x_0, x) = I\left[x \in (x_0 - \lambda, x_0 + \lambda)\right] \frac{3}{4}\left[1 - \left(\frac{x - x_0}{\lambda}\right)^2\right]$

    where λ is width or smoothing parameter. More general, use $h_\lambda(x_0)$ instead λ.

One-Dimensional Kernel Smoothers



One-Dimensional Kernel Smoothers

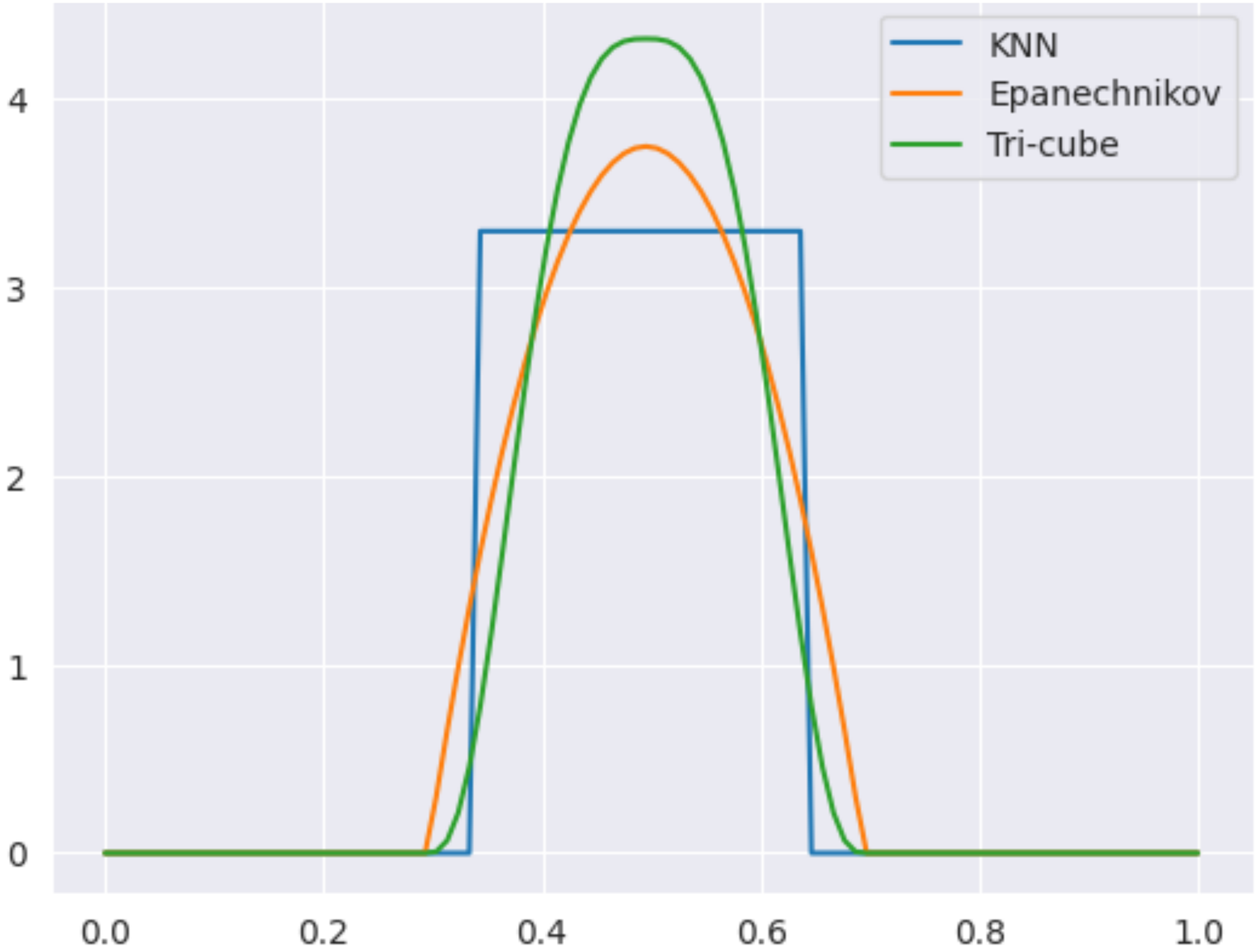
- **Tri-cube kernel**: $K_\lambda(x_0, x) = I\left[x \in (x_0 - \lambda, x_0 + \lambda)\right]\left[1 - \left|\frac{x - x_0}{\lambda}\right|^3\right]^3$

- **k-NN kernel**:

  $K_k(x_0, x) = I\left[x \in N_k(x)\right]\frac{1}{2|x_0 - x_{[k]}|} = I\left[x \in (x_0 - |x_0 - x_{[k]}|, x_0 + |x_0 - x_{[k]}|)\right]\frac{1}{2|x_0 - x_{[k]}|}$

  where $x_{[k]}$ is the k-th closest x_i to x_0.

One-Dimensional Kernel Smoothers

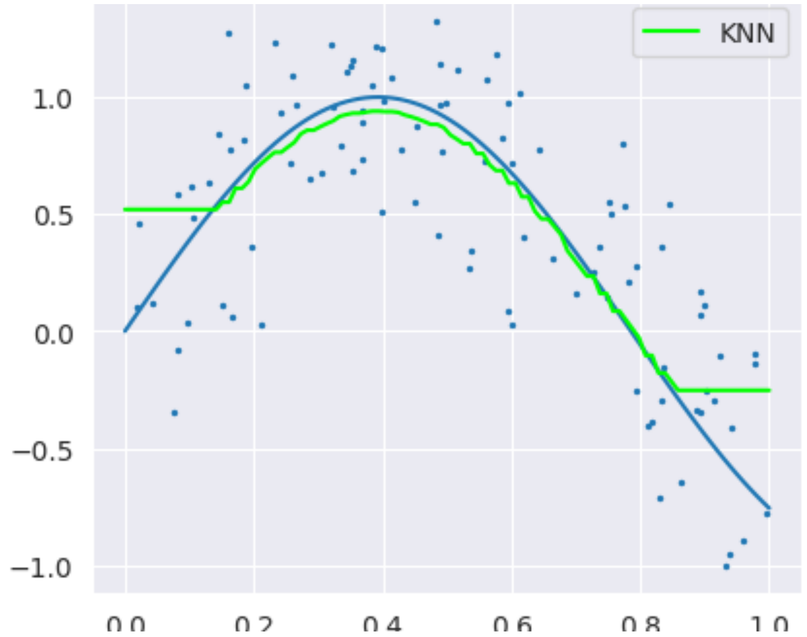


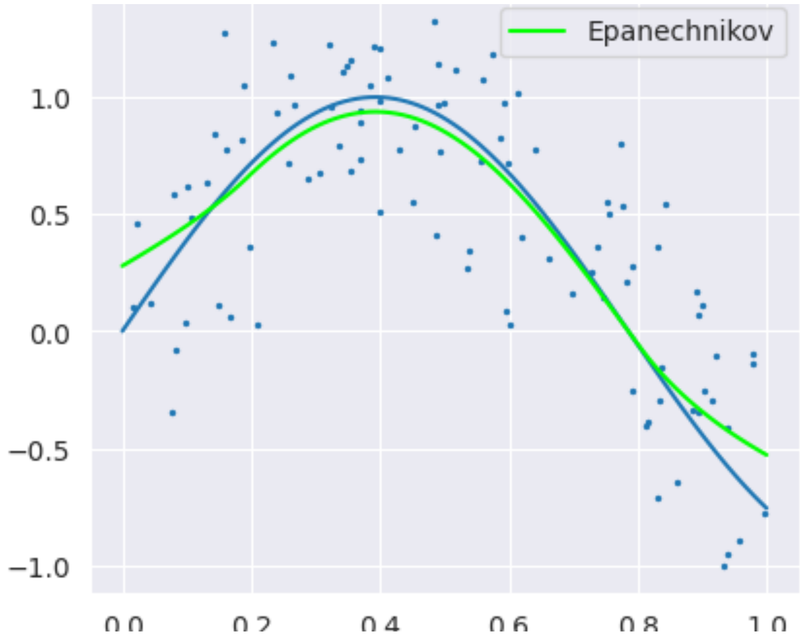
One-Dimensional Kernel Smoothers

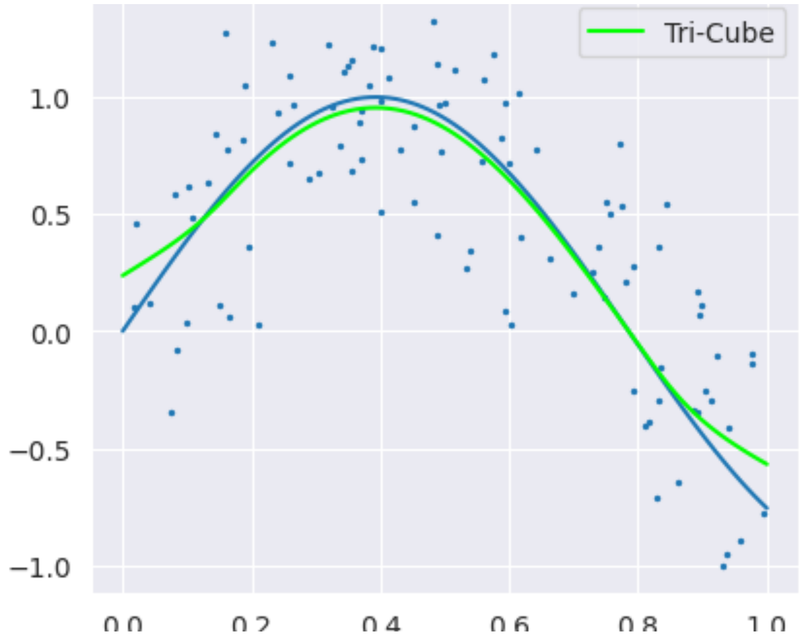
- The smoothing parameter λ, has to be determined. Large λ implies lower variance, but higher bias.
- NW kernel can be badly based on the **boundaries** of the domain, because of asymmetry of the kernel in that region.

One-Dimensional Kernel Smoothers

- Most observations in neighborhood have higher mean than x_0 s.t. near the boundaries
- By fitting straight lines rather then constant locally, we can remove this bias exactly to first order







One-Dimensional Kernel Smoothers

- **Locally weighted regression:** $\underset{\alpha(x_0)\beta(x_0)}{min} \sum_{i=1}^{N} K_\lambda(x_0, x_i)\left[y_i - \alpha(x_0) - \beta(x_0)x_i\right]^2$

- The estimate is then $\hat{f}(x_0) = \hat{\alpha}(x_0) + \hat{\beta}(x_0)x_0$

- Notice that although we fit an entire linear model to the data in the region, we only use it to evaluate the fit at the single point x_0.

One-Dimensional Kernel Smoothers

- Define the vector-valued function

$$b(x)^T = (1, x), \text{ and } B = \begin{bmatrix} b(x_1)^T \\ \vdots \\ b(x_N)^T \end{bmatrix}, \mathbf{W} = \text{diag}(K_\lambda(x_0, x_1), \ldots, K_\lambda(x_0, x_N))$$

$\Rightarrow$ Let $\beta = \begin{pmatrix} \alpha_0 \\ \beta_0 \end{pmatrix} \in \mathbb{R}^2$. $\Rightarrow \sum_{i=1}^{N} K(x_0, x_i)[y_i - \alpha_0 - \beta_0 x_i]^2 = \| W(x_0)^{\frac{1}{2}}(Y - B\beta) \|_2^2$ ($\because K_\lambda(x_0, x_i) \geq 0$ for $\forall i$).

$\Rightarrow L = \| W(x_0)^{\frac{1}{2}}(Y - B\beta) \|_2^2 = (Y - B\beta)^T W(x_0)(Y - B\beta) = Y^T W(x_0) Y - Y^T W(x_0) B\beta - \beta^T B^T W(x_0) Y + \beta^T B^T W(x_0) B \beta$

$\Rightarrow \frac{\partial L}{\partial \beta} = -Y^T W(x_0) B - Y^T W(x_0) B + \beta^T \cdot 2 B^T W(x_0) B = 0 \Rightarrow \hat{\beta}^T (B^T W(x_0) B) = Y^T W(x_0) B \quad \hat{\beta}^T = Y^T W(x_0) B (B^T W(x_0) B)^{-1}$

$\therefore \hat{\beta} = (B^T W(x_0) B)^{-1} B^T W(x_0) Y.$

One-Dimensional Kernel Smoothers

$$\Rightarrow \hat{f}(x_0) = b(x_0)^T (B^T W(x_0) B)^{-1} B^T W(x_0) y = b(x_0)^T \left(\begin{pmatrix} 1 & \cdots & 1 \\ x_1 & \cdots & x_N \end{pmatrix} \begin{pmatrix} K(x_1, x_0) & K(x_1, x_0)(x_1 - x_0) \\ \vdots & \vdots \\ K(x_N, x_0) & K(x_N, x_0)(x_N - x_0) \end{pmatrix} \right)^{-1} B^T W(x_0) y$$

$$= (1 \quad x_0) \begin{pmatrix} \sum_i K(x_i, x_0) & \sum_i K(x_i, x_0)(x_i - x_0) \\ \sum_i K(x_i, x_0)(x_i - x_0) & \sum_i K(x_i, x_0)(x_i - x_0)^2 \end{pmatrix}^{-1} \begin{pmatrix} K(x_1, x_0) & \cdots & K(x_N, x_0) \\ K(x_1, x_0)(x_1 - x_0) & & K(x_N, x_0)(x_N - x_0) \end{pmatrix} \begin{pmatrix} y_1 \\ \vdots \\ y_N \end{pmatrix}$$

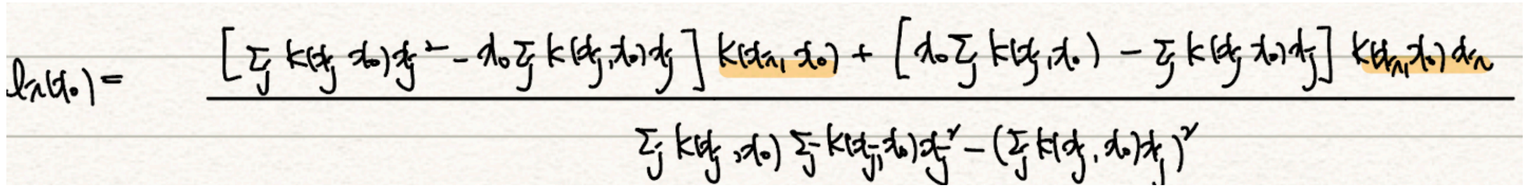
$$= (1 \quad x_0) \frac{1}{\sum_i K(x_i, x_0) \sum_i K(x_i, x_0)(x_i - x_0)^2 - \left(\sum_i K(x_i, x_0)(x_i - x_0)\right)^2} \begin{pmatrix} \sum_i K(x_i, x_0)(x_i - x_0)^2 & -\sum_i K(x_i, x_0)(x_i - x_0) \\ -\sum_i K(x_i, x_0)(x_i - x_0) & \sum_i K(x_i, x_0) \end{pmatrix} \begin{pmatrix} \sum_i K(x_i, x_0) y_i \\ \sum_i K(x_i, x_0)(x_i - x_0) y_i \end{pmatrix}$$

$$= \frac{1}{\sum_i K(x_i, x_0) \sum_i K(x_i, x_0)(x_i - x_0)^2 - \left(\sum_i K(x_i, x_0)(x_i - x_0)\right)^2} \left(\sum_i K(x_i, x_0)(x_i - x_0)^2 - x_0 \sum_i K(x_i, x_0)(x_i - x_0) \quad -\sum_i K(x_i, x_0)(x_i - x_0) + x_0 \sum_i K(x_i, x_0) \right) \begin{pmatrix} \sum_i K(x_i, x_0) y_i \\ \sum_i K(x_i, x_0)(x_i - x_0) y_i \end{pmatrix}$$

$$= \frac{1}{\sum_i K(x_i, x_0) \sum_i K(x_i, x_0)(x_i - x_0)^2 - \left(\sum_i K(x_i, x_0)(x_i - x_0)\right)^2} \left(\sum_i K(x_i, x_0)(x_i - x_0)^2 - x_0 \sum_i K(x_i, x_0)(x_i - x_0) \right) \sum_i K(x_i, x_0) y_i + \left(x_0 \sum_i K(x_i, x_0) - \sum_i K(x_i, x_0)(x_i - x_0) \right) \sum_i K(x_i, x_0)(x_i - x_0) y_i$$

$$= \sum_{i=1}^N l_i(x_0) y_i. \text{ where } l_i(x_0) \text{ does not involve } y_i \Rightarrow \hat{f}(x_0) \text{ is linear in } y_0.$$

One-Dimensional Kernel Smoothers

- Thus, $\hat{f}(x_0) = \sum_{i=1}^{N} l_i(x_0) y_i$ where 

- The estimate is linear in the y_i. These weights $l_i(x_0)$ are sometimes referred to as the equivalent kernel.

- $\sum_{i=1}^{N} l_i(x_0) = 1$ and $\sum_{i=1}^{N} (x_i - x_0) l_i(x_0) = 0$

One-Dimensional Kernel Smoothers

$\Rightarrow$ Let $\sum_n K(x_n, x_0) = \alpha$, $\sum_n K(x_n, x_0) x_n = \beta$, $\sum_n K(x_n, x_0) x_n^2 = \gamma$

$$\Rightarrow \sum_{n=1}^{N} l_n(x_0) = \frac{\alpha\gamma - x_0\alpha\beta + x_0\alpha\beta - \beta^2}{\alpha\gamma - \beta^2} = \frac{\alpha\gamma - \beta^2}{\alpha\gamma - \beta^2} = 1.$$

$$\sum_{n=1}^{N} (x_n - x_0)\, l_n(x_0) = \frac{1}{\alpha\gamma - \beta^2}\Big[(\gamma - x_0\beta) \underbrace{\textstyle\sum_n K(x_n, x_0)(x_n - x_0)}_{= \beta - x_0\alpha} + (x_0\alpha - \beta) \underbrace{\textstyle\sum_n K(x_n, x_0) x_n (x_n - x_0)}_{= \gamma - x_0\beta} \Big]$$

$$= \frac{1}{\alpha\gamma - \beta^2}\Big[\beta\gamma - x_0\alpha\gamma - x_0\beta^2 + x_0^2\alpha\beta + x_0\alpha\gamma - x_0^2\alpha\beta - \beta\gamma + x_0\beta^2 \Big] = 0.$$

One-Dimensional Kernel Smoothers

- (**Automatic kernel carpentry**) Local linear regression automatically modifies the kernel to correct the bias exactly to first order.

- Let f be true function, $Y_i = f(x_i) + \epsilon_i$ where ϵ_i are i.i.d with $\mathbb{E}[\epsilon_i] = 0$,Var$[\epsilon_i] = \sigma^2$.

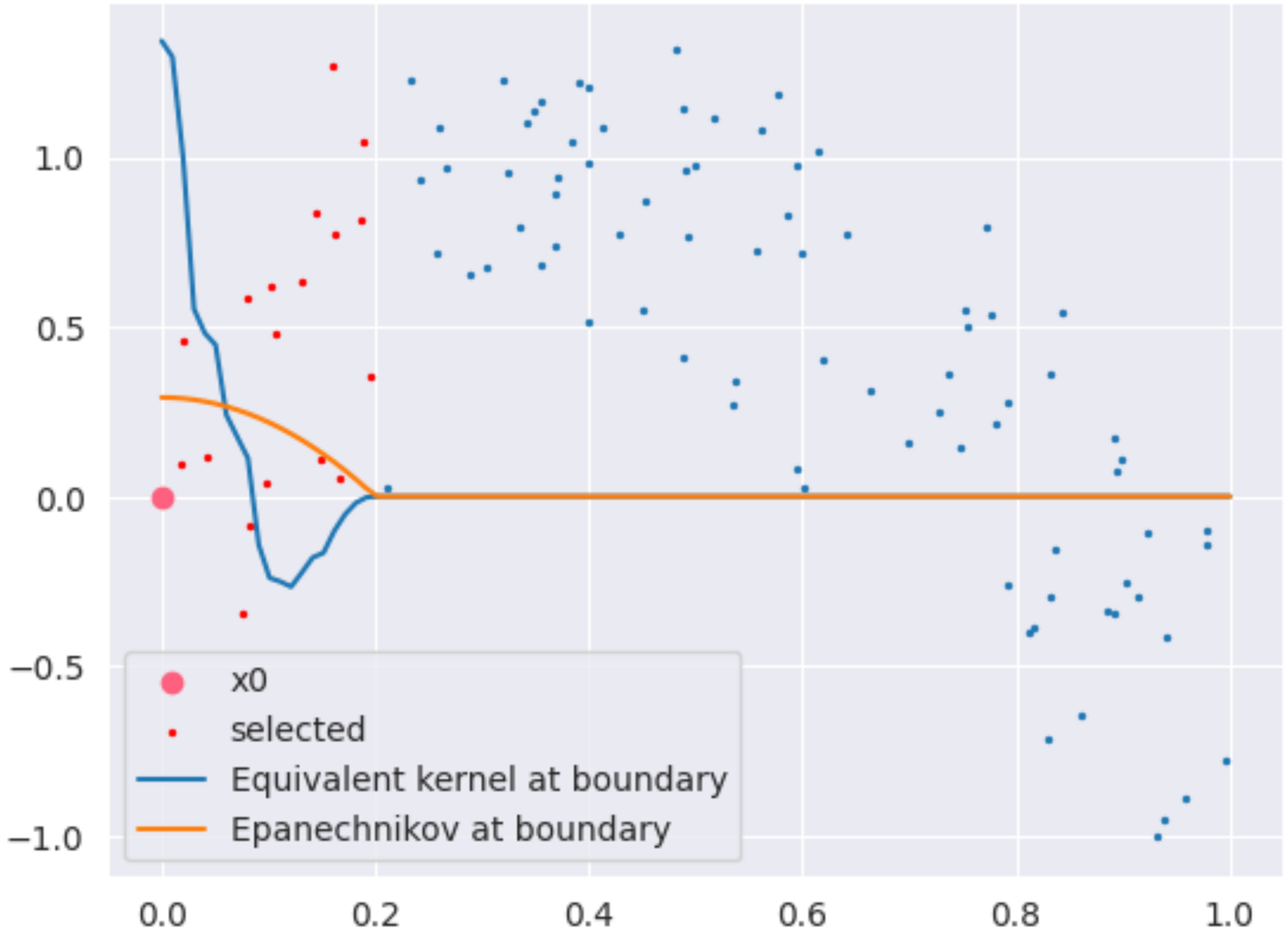
- Then by Taylor series expansion,

$$\begin{aligned}
\mathrm{E}\hat{f}(x_0) &= \sum_{i=1}^{N} l_i(x_0) f(x_i) \\
&= f(x_0) \sum_{i=1}^{N} l_i(x_0) + f'(x_0) \sum_{i=1}^{N} (x_i - x_0) l_i(x_0) \\
&\quad + \frac{f''(x_0)}{2} \sum_{i=1}^{N} (x_i - x_0)^2 l_i(x_0) + R,
\end{aligned}$$

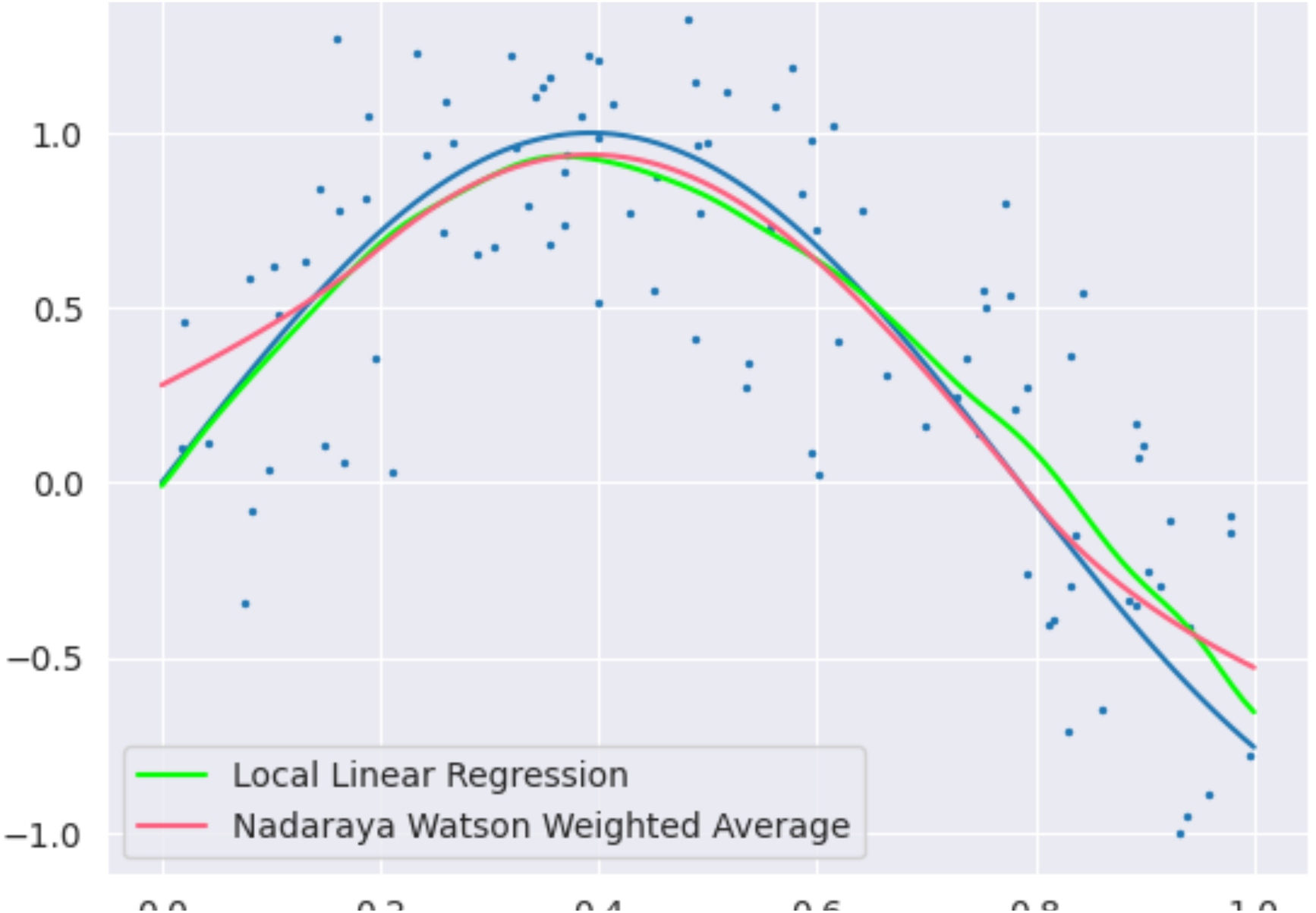
One-Dimensional Kernel Smoothers

- Thus, the bias is $\mathbb{E}\hat{f}(x_0) - f(x_0) = \frac{f''(x_0)}{2}\sum_i (x_i - x_0)^2 l_i(x_0) + R$ which depends only on quadratic and higher-order terms in the expansion of f.
- Under suitable smoothness assumptions, remained term is typically small.
- (**Trimming the hills**) Local linear fits tend to be biased in regions of curvature of the true function. -> local quadratic regression is generally able to correct this bias.

One-Dimensional Kernel Smoothers



One-Dimensional Kernel Smoothers



One-Dimensional Kernel Smoothers

- Local Polynomial Regression:

$$\min_{\alpha(x_0)\beta_j(x_0),j=1,\ldots,d} \sum_{i=1}^{N} K_\lambda(x_0, x_i) \left[y_i - \alpha(x_0) - \sum_{j=1}^{d} \beta_j(x_0) x_i^j \right]^2$$ with solution

$$\hat{f}(x_0) = \hat{\alpha}(x_0) + \sum_{j=1}^{d} \hat{\beta}_j(x_0) x_j^j$$

One-Dimensional Kernel Smoothers

Thm. Let $b_j(x_0) = \sum_{i=1}^N (x_i - x_0)^j l_i(x_0)$ for local polynomial regression of degree k.

Then $b_j(x_0) = \begin{cases} 1 & j = 0 \\ 0 & j = 1, \cdots, k \end{cases}$

Pf) $B = \begin{pmatrix} 1 & x_1 & \cdots & x_1^k \\ \vdots & \vdots & \cdots & \vdots \\ 1 & x_N & & x_N^k \end{pmatrix} = \begin{pmatrix} b(x_1)^T \\ \vdots \\ b(x_N)^T \end{pmatrix} \in \mathbb{R}^{N \times (k+1)}$, $W(x_0) = \text{diag}\left(K_\lambda(x_0, x_1) \cdots K_\lambda(x_0, x_N)\right)$

$\Rightarrow$ let $\beta = (\alpha(x_0)\ \beta_1(x_0) \cdots \beta_k(x_0))^T \in \mathbb{R}^{k+1}$

$\Rightarrow \sum_{i=1}^N K_\lambda(x_0, x_i)\left[y_i - \alpha(x_0) - \beta_1(x_0)x - \cdots - \beta_k(x_0)x^k\right]^2 = \| W(x_0)^{\frac{1}{2}}(Y - B\beta)\|_2^2 \Rightarrow \hat{\beta} = (B^T W(x_0) B)^{-1} B^T W(x_0) Y$

$\Rightarrow l(x_0)^T = b(x_0)^T (B^T W(x_0) B)^{-1} B^T W(x_0) \in \mathbb{R}^{1 \times N}$. Note that $l(x_0)^T B = b(x_0)^T$

$\Rightarrow l(x_0)^T B = (l_1(x_0) \cdots l_N(x_0)) \begin{pmatrix} 1 & x_1 & \cdots & x_1^k \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_N & \cdots & x_N^k \end{pmatrix} = \left(\sum_i l_i(x_0) \quad \sum_i l_i(x_0) x_i \cdots \sum_i l_i(x_0) x_i^k\right) = (1\ x_0 \cdots x_0^k) \Rightarrow b_0(x_0) = \sum_i l_i(x_0) = 1.$

$\Rightarrow \sum_{i=1}^N (x_i - x_0)^j l_i(x_0) = \sum_{i=1}^N \sum_{k=0}^j \binom{j}{k} x_i^k (-x_0)^{j-k} l_i(x_0) = \sum_{k=0}^j \binom{j}{k} (-x_0)^{j-k} \sum_{i=1}^N x_i^k l_i(x_0) = \sum_{k=0}^j \binom{j}{k} (-x_0)^{j-k} x_0^k = (x_0 - x_0)^j = 0$. for $j = 1, \cdots, k$.

One-Dimensional Kernel Smoothers

Corollary. (Relation between degrees & bias). $E[\hat{f}(x_0)] - f(x_0) = R$ where R involves $(k+1)$- and higher-order derivatives of f.

pf) $E[\hat{f}(x_0)] - f(x_0) = \sum_{i=1}^{N} l_i(x_0) f(x_i) - f(x_0) = \left[f(x_0)\sum_{i=1}^{N} l_i(x_0) + f'(x_0)\sum_{i=1}^{N} l_i(x_0)(x_i - x_0) + \cdots + \frac{f^{(k)}(x_0)}{k!}\sum_{i=1}^{N} l_i(x_0)(x_i - x_0)^k + R \right] - f(x_0)$

$= R.$

- (**Bias-Variance Tradeoff**) There is of course a price to be paid for bias reduction, and that is increased variance
- $\text{Var}\left[\hat{f}(x_0)\right] = \sum_{i=1}^{N} l_i(x_0)^2 \sigma^2 = \|l(x_0)\|_2^2$ where $l(x_0) = (l_1(x_0), \ldots, l_N(x_0))^T$

One-Dimensional Kernel Smoothers

Thm. (Relation between degrees & variance). $\|l(x_0)\| = \|(l_1(x_0), \cdots, l_N(x_0))\|$ increases with d.

pf) $\|l(x_0)\|_2^2 = [b(x_0)^T (B^T W(x_0) B)^{-1} B^T W(x_0)][W(x_0) B (B^T W(x_0) B)^{-1} b(x_0)]$

$= b(x_0)^T (B^T W(x_0) B)^{-1} B^T W^2(x_0) B (B^T W(x_0) B)^{-1} b(x_0)$

$= b(x_0)^T B^{-1} (B^T)^{-1} b(x_0) = b(x_0)^T (B^T B)^{-1} b(x_0)$

$\Rightarrow$ let $b'(x_0) = \begin{pmatrix} b(x_0) & x_0^{d+1} \end{pmatrix}$, $B = \begin{pmatrix} B & b \end{pmatrix}$ where $b = (x_1^{d+1} \cdots x_N^{d+1})^T \in \mathbb{R}^{d+1}$

$\Rightarrow \|l'(x_0)\|_2^2 = \begin{pmatrix} b(x_0)^T \\ x_0^{d+1} \end{pmatrix} \left(\begin{pmatrix} B^T \\ b^T \end{pmatrix} \begin{pmatrix} B & b \end{pmatrix} \right)^{-1} \begin{pmatrix} b(x_0) & x_0^{d+1} \end{pmatrix}$ $\Rightarrow$ check that $\|l'(x_0)\|_v^2 \geq \|l(x_0)\|_v^2$

$= \begin{pmatrix} b(x_0)^T \\ x_0^{d+1} \end{pmatrix} \begin{pmatrix} B^T B & B^T b \\ b^T B & b^T b \end{pmatrix}^{-1} \begin{pmatrix} b(x_0) & x_0^{d+1} \end{pmatrix}$

$b^T b \in \mathbb{R}$

One-Dimensional Kernel Smoothers

$$\begin{pmatrix} B^TB & B^Tb \\ b^TB & b^Tb \end{pmatrix} = \begin{pmatrix} A & B \\ B^T & C \end{pmatrix} \text{ Form.}$$

$$\begin{pmatrix} I & 0 \\ -B^TA^{-1} & I \end{pmatrix}\begin{pmatrix} A & B \\ B^T & C \end{pmatrix}\begin{pmatrix} I & -A^{-1}B \\ 0 & I \end{pmatrix} = \begin{pmatrix} A & 0 \\ 0 & C-B^TA^{-1}B \end{pmatrix}$$

$$\begin{pmatrix} A & B \\ B^T & C \end{pmatrix}^{-1} = \begin{pmatrix} I & -A^{-1}B \\ 0 & I \end{pmatrix}\begin{pmatrix} A^{-1} & 0 \\ 0 & (C-B^TA^{-1}B)^{-1} \end{pmatrix}\begin{pmatrix} I & 0 \\ -B^TA^{-1} & I \end{pmatrix} = \begin{pmatrix} A^{-1}+A^{-1}B(C-B^TA^{-1}B)^{-1}B^TA^{-1} & -A^{-1}B(C-B^TA^{-1}B)^{-1} \\ -(C-B^TA^{-1}B)^{-1}B^TA^{-1} & (C-B^TA^{-1}B)^{-1} \end{pmatrix}$$

$$\Rightarrow \begin{pmatrix} B^TB & B^Tb \\ b^TB & b^Tb \end{pmatrix} = \begin{pmatrix} (B^TB)^{-1}+(B^TB)^{-1}B^Tb\,(b^Tb-b^TB(B^TB)^{-1}B^Tb)^{-1}b^TB(B^TB)^{-1} & -(B^TB)^{-1}B^Tb\,(b^Tb-b^TB(B^TB)^{-1}B^Tb)^{-1} \\ -(b^Tb-b^TB(B^TB)^{-1}B^Tb)^{-1}b^TB(B^TB)^{-1} & (b^Tb-b^TB(B^TB)^{-1}B^Tb)^{-1} \end{pmatrix}$$

$= \beta \in \mathbb{R}^N$ (the $-(B^TB)^{-1}B^Tb\,(b^Tb-b^TB(B^TB)^{-1}B^Tb)^{-1}$ block)

$= \alpha \in \mathbb{R}$ (the $(b^Tb-b^TB(B^TB)^{-1}B^Tb)^{-1}$ block)

$$= \begin{pmatrix} (B^TB)^{-1}+\alpha\beta\beta^T & -\alpha\beta \\ -\alpha\beta^T & \alpha \end{pmatrix}$$

One-Dimensional Kernel Smoothers

$$\Rightarrow \|\ell'(x_0)\|_2^2 = b(x_0)^T\left[(B^TB)^{-1} + \alpha\beta\beta^T\right]b(x_0) - b(x_0)^T\alpha\beta x_0^{d+1} - x_0^{d+1}\alpha\beta^T b(x_0)^T + \alpha(x_0^{d+1})^2$$

$$= \|\ell(x_0)\|_2^2 + \alpha\underbrace{\|x_0^{d+1} - b(x_0)^T\beta\|_2^2}_{\geq 0}$$

$$\Rightarrow \text{Enough to show that } b^Tb - b^TB(B^TB)^{-1}B^Tb \geq 0 \Rightarrow b^Tb - b^TB(B^TB)^{-1}B^Tb = b^T\underbrace{(I - B(B^TB)^{-1}B^T)}_{=(I-H)}b \geq 0 \quad \square.$$

Projection matrix.

- Selecting the parameter d, λ -> Cross validation
- Ex. Leave-One-Out Cross-Validation

One-Dimensional Kernel Smoothers

In Local regression, using Smoother matrix,

$$\hat{f} = B(B^T W(x_0) B)^{-1} B^T W(x_0) y = \begin{bmatrix} b(x_1)^T \\ \vdots \\ b(x_N)^T \end{bmatrix} (B^T W(x_0) B)^{-1} B^T W(x_0) y = \begin{bmatrix} l(x_1)^T \\ \vdots \\ l(x_N)^T \end{bmatrix} y = \begin{bmatrix} l_1(x_1) & \cdots & l_N(x_1) \\ \vdots & \ddots & \vdots \\ l_1(x_N) & \cdots & l_N(x_N) \end{bmatrix} y = S_\lambda y.$$

- LOOCV: $\mathrm{CV}(\hat{f}_\lambda) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{f}^{-i}(x_i))^2 = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{y_i - \hat{f}(x_i)}{1 - S_{\lambda_{ii}}}\right)^2$

One-Dimensional Kernel Smoothers

pf) $\{S_\lambda\}_{i,i'} = b(x_i)^T (B^T W(x_0) B)^{-1} b(x_{i'}) K_\lambda(x_0, x_{i'})$

$\Rightarrow$ Let B_{-i}, Y_{-i}, $W(x_0)_{-i}$ be removed $b(x_i)$, y_i, $K_\lambda(x_0, x_i)$ i.e. $B_{-i} = \begin{bmatrix} b(x_1)^T \\ \vdots \\ 0^T \\ b(x_N)^T \end{bmatrix}$, $Y_{-i} = \begin{bmatrix} y_1 \\ \vdots \\ 0 \\ y_N \end{bmatrix}$.

$\Rightarrow \hat{f}^{-i}(x_i) = b(x_i)^T (B_{-i}^T W(x_0)_{-i} B_{-i})^{-1} B_{-i}^T W(x_0)_{-i} Y_{-i}$

$\Rightarrow$
$$B_{-i}^T W(x_0)_{-i} B_{-i} = \sum_{j=1}^N b(x_j) K_\lambda(x_0, x_j) b(x_j)^T - b(x_i) K_\lambda(x_0, x_i) b(x_i)^T$$
$$B_{-i}^T W(x_0)_{-i} Y_{-i} = \sum_{j=1}^N b(x_j) K_\lambda(x_0, x_j) y_j - b(x_i) K_\lambda(x_0, x_i) y_i$$

$\Rightarrow$ By Woodbury matrix identity,

$$(B_{-i}^T W(x_0)_{-i} B_{-i})^{-1} = (B^T W(x_0) B - b(x_i) K_\lambda(x_0, x_i) b(x_i)^T)^{-1}$$

$$= (B^T W(x_0) B)^{-1} + \frac{1}{\frac{1}{K_\lambda(x_0, x_i)} - b(x_i)^T (B^T W(x_0) B)^{-1} b(x_i)} \cdot (B^T W(x_0) B)^{-1} b(x_i) b(x_i)^T (B^T W(x_0) B)^{-1}.$$

$$= (B^T W(x_0) B)^{-1} + \frac{K_\lambda(x_0, x_i)}{1 - \{S_\lambda\}_{i,i}} \cdot (B^T W(x_0) B)^{-1} b(x_i) b(x_i)^T (B^T W(x_0) B)^{-1}.$$

One-Dimensional Kernel Smoothers

$$\hat{f}^{-i}(x_i) = b(x_i)^T \left[(B^T W(x_0) B)^{-1} + \frac{K_\lambda(x_0, x_i)}{1 - \sum_{i'} s_{i'} y_{i'i}} \cdot (B^T W(x_0) B)^{-1} b(x_i) b(x_i)^T (B^T W(x_0) B)^{-1} \right] \left[B^T W(x_0) y - K_\lambda(x_0, x_i) b(x_i) y_i \right]$$

$$= \hat{f}(x_i) - b(x_i)^T (B^T W(x_0) B)^{-1} K_\lambda(x_0, x_i) b(x_i) y_i + \frac{\sum_{i'} s_{i'} y_{i'i} b(x_i)^T (B^T W(x_0) B)^{-1}}{1 - \sum_{i'} s_{i'} y_{i'i}} \cdot B^T W(x_0) y - \frac{\sum_{i'} s_{i'} y_{i'i} b(x_i)^T (B^T W(x_0) B)^{-1}}{1 - \sum_{i'} s_{i'} y_{i'i}} K_\lambda(x_0, x_i) b(x_i) y_i$$

$$= \hat{f}(x_i) - \sum_{i'} s_{i'} y_{i'i} y_i + \frac{\sum_{i'} s_{i'} y_{i'i} \hat{f}(x_i)}{1 - \sum_{i'} s_{i'} y_{i'i}} - \frac{\sum_{i'} s_{i'} y_{i'i}^2 y_i}{1 - \sum_{i'} s_{i'} y_{i'i}}$$

$$= \frac{\hat{f}(x_i) - \sum_{i'} s_{i'} y_{i'i} y_i}{1 - \sum_{i'} s_{i'} y_{i'i}}$$

$$\Rightarrow y_i - \hat{f}^{-i}(x_i) = y_i - \frac{\hat{f}(x_i) - \sum_{i'} s_{i'} y_{i'i} y_i}{1 - \sum_{i'} s_{i'} y_{i'i}} = \frac{y_i - \hat{f}(x_i)}{1 - \sum_{i'} s_{i'} y_{i'i}}.$$

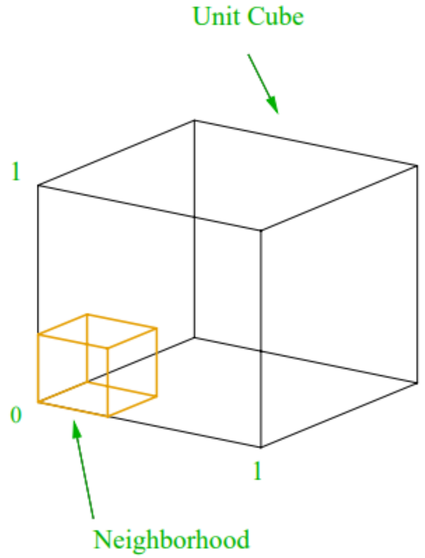
Local Regression in $\mathbb{R}^p$

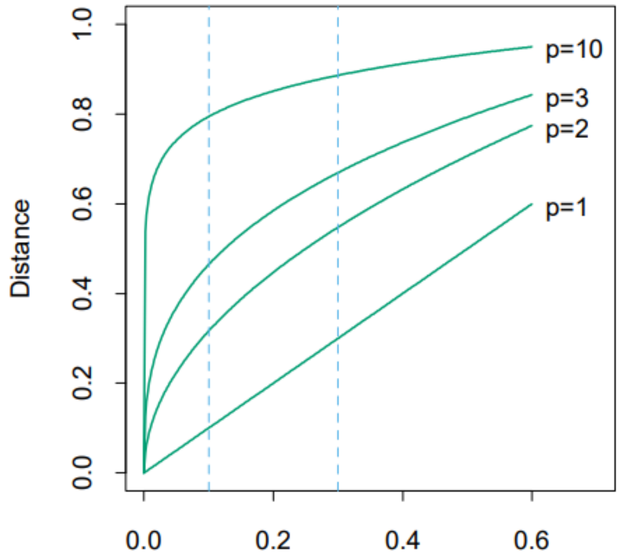
- Define the vector of polynomial terms in X of maximum degree d. For $d = 1$, $p = 2$, $b(X) = (1, X_1, X_2, X_1^2, X_2^2, X_1 X_2)$
- At each $x_0 \in \mathbb{R}^p$, solve $\min_{\beta(x_0)} \sum_{i=1}^{N} K_\lambda(x_0, x_i) \left[y_i - b(x_i)^T \beta(x_0) \right]^2$ to produce the fit $\hat{f}(x_0) = b(x_0)^T \hat{\beta}(x_0)$ with kernel metric $\frac{\|x - x_0\|}{\lambda}$ where $\| \cdot \|$ is the Euclidean norm.
- Since Euclidean norm depends on the units in each coordinate, it makes sense to standardize each predictor.

Local Regression in $\mathbb{R}^p$

- (**Curse of dimensionality**) Boundary effects are a much higher problem in high dimensionality. Local polynomial regression seamlessly performs boundary correction to the desired order in any dimensions.

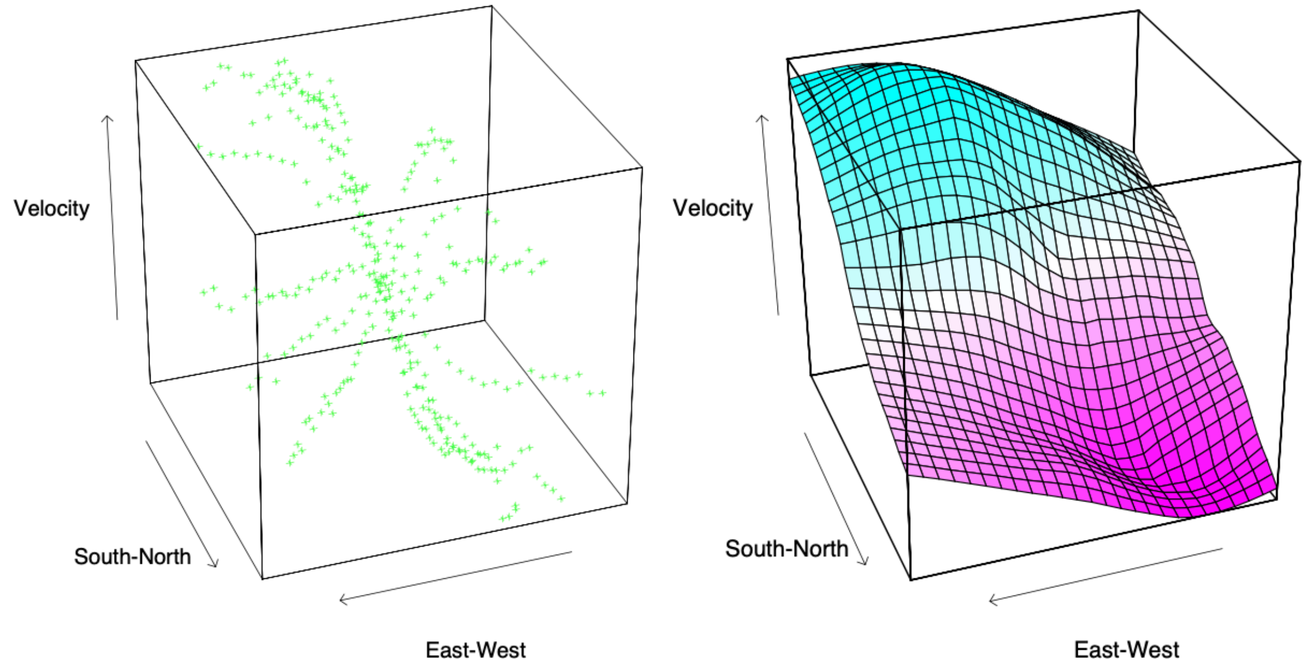
.





Local Regression in $\mathbb{R}^p$

- (**Irregular Boundary case**) For $\mathbb{R}^2$, where the response is the velocity measurements on a galaxy, and the two predictors record positions on the celestial sphere.
- Local linear regression smoothing with NN window with 15% of the data



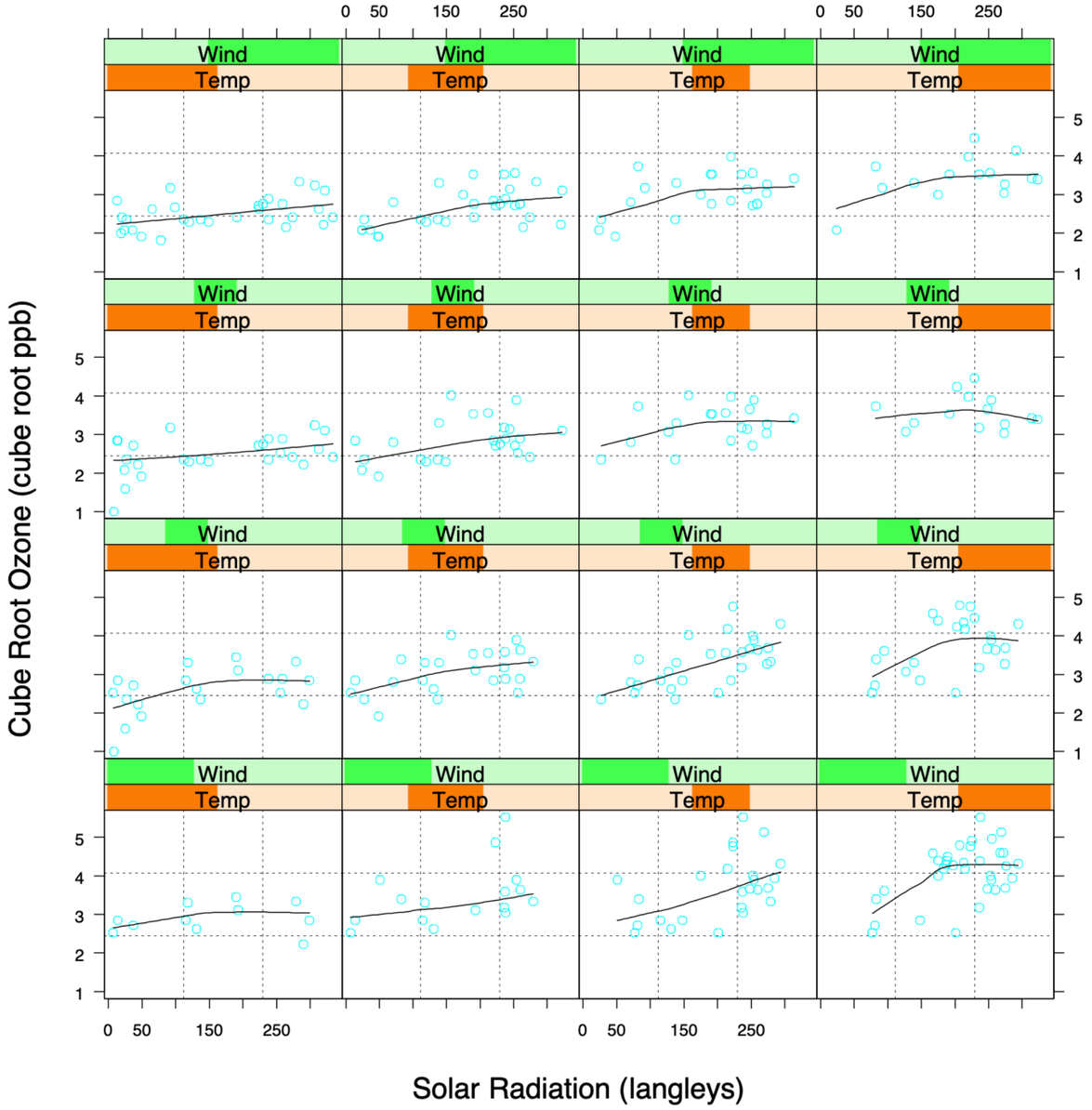
Local Regression in $\mathbb{R}^p$

- **Structured Local Regression:** We need to make some structured assumptions about the model to avoid curse of dimensionality.

- Ex. For $\mathbb{R}^3$ smoothing case. The response is ozone concentration with predictors temperature, wind speed, radiation. The trellis display shows ozone as a function of radiation, conditioned on intervals of temperature and wind speed.

.

Local Regression in $\mathbb{R}^p$



Local Regression in $\mathbb{R}^p$

- **ANOVA decompositions of the form:** $\mathbb{E}(Y|X) = f(X_1, \ldots, X_p) = \alpha + \sum_j g_j(X_j) + \sum_{k<l} g_{kl}(X_k, X_l) + \ldots$
  - **Additive models** assume only main effect terms: $f(X) = \alpha + \sum_{j=1}^{p} g_j(X_j)$
  - **Second order models:** $f(X) = \alpha + \sum_j g_j(X_j) + \sum_{k<l} g_{kl}(X_k, X_l)$
  - **Varying coefficients models:** $f(X) = \alpha(Z) + \sum_{j=1}^{q} \beta_j(Z)X_j$ where $Z = (X_{q+1}, \ldots, X_p)$ with $q < p$ and fitting by locally weighted least squares.

Local Regression in $\mathbb{R}^p$

- **(Iterative back-fitting algorithms)** In the additive model, for example, if all but the k-th term is assumed known, then we can estimate $\hat{g}_k$ based on $\{(x_i, y_i - \hat{\alpha} - \sum_{k \neq j} \hat{g}_j(X_j)) : i = 1,..,N\}$. (Initialize $\hat{\alpha} = \bar{y}, \hat{g}_j(X_j) = 0$) until convergence for $j = 1,...,p$. Notice that at any stage, one-dimensional local regression is all that is needed.

Local Likelihood and Other Models

- Any parametric model can be made local if the fitting method accommodates observation weights.
- **Local Log Likelihood:** $l(\beta(x_0)) = \sum_{i=1}^{N} K_\lambda(x_0, x_i) l(y_i, x_i^T \beta(x_0))$
  - **Ex. Locally weighted Logistic Regression:** $l(\beta(x_0)) = \sum_{i=1}^{N} K_\lambda(x_0, x_i) \log p_{g_i}(x_i - x_0; \beta(x_0))$

    (Centered at x_0**)** where $p_{g_i}(x_i; \beta(x_0)) = Pr(G = g_i \mid X = x)$, $\log \frac{Pr(G = j \mid X = x)}{Pr(G = J \mid X = x)} = \beta_j(x_0)^T x$ for $j = 1, \ldots, J-1$

Local Likelihood and Other Models

Note that $Pr(G = J \mid X = x) = \dfrac{1}{1 + \sum_{k=1}^{J-1} \exp(\beta_{k0}(x_0) + \beta_k(x_0)^T x)}$,

$$Pr(G = j \mid X = x) = \frac{\exp(\beta_{j0}(x_0) + \beta_j^T(x_0)x)}{1 + \sum_{k=1}^{J-1} \exp(\beta_{k0}(x_0) + \beta_k(x_0)^T x)} = Pr(G = J \mid X = x)\exp(\beta_{j0}(x_0) + \beta_j(x_0)^T x) \text{ for } j = 1,\ldots J-1$$

$$l(\beta(x_0)) = \sum_{i=1}^{N} K_\lambda(x_0, x_i) \log p_{g_i}(x_i - x_0; \beta(x_0))$$

$$= \sum_{i=1}^{N} K_\lambda(x_0, x_i) \left[\sum_{j=1}^{J-1} \left\{ I(y_i = j)(\beta_{j0}(x_0) + \beta_j(x_0)^T(x_i - x_0) + \log p_J(x_i - x_0; \beta_j(x_0))) \right\} + I(y_i = J) \log p_J(x_i - x_0; \beta_j(x_0)) \right]$$

$$= \sum_{i=1}^{N} K_\lambda(x_0, x_i) \left[\beta_{g_i 0} + \beta_{g_i}(x_0)^T(x_i - x_0) - \log \left\{ 1 + \sum_{k=1}^{J-1} \exp(\beta_{k0}(x_0) + \beta_k(x_0)^T(x_i - x_0)) \right\} \right]$$

Local Likelihood and Other Models

- We have centers the local regression at x_0, so that the fitted posterior probabilities at x_0:

$$\hat{Pr}(G = j \,|\, X = x_0) = \frac{\exp(\hat{\beta}_{j0}(x_0))}{1 + \sum_{k=1}^{J-1} \exp(\hat{\beta}_{k0}(x_0))}$$

- **Newton-Rapson Method:**

  - Let X be centered at x_0 , $W(x_0) = \text{diag}(K_\lambda(x_0, x_1), \ldots, K_\lambda(x_0, x_N))$,
    $W' = \text{diag}(K_\lambda(x_0, x_1)p(x_1 - x_0; \beta(x_0)(1 - p(x_1 - x_0; \beta(x_0)), \ldots$
    $, K_\lambda(x_N, x_1)p(x_N - x_0; \beta(x_0)(1 - p(x_N - x_0; \beta(x_0))$.

  - Then $\frac{\partial l(\beta(x_0)}{\partial \beta} = W(x_0)X^T(y - p)$, $\frac{\partial l(\beta(x_0)}{\partial \beta \partial \beta^T} = -X^T W' X$

Local Likelihood and Other Models

- $\beta(x_0)^{new} = \beta(x_0)^{old} + (X^T W' X)^{-1} W(x_0) X^T (y - p)$
- Since $y_i \sim$ Bernoulli(p_i), by CLT, $\hat{\beta}(x_0) \sim \mathcal{N}(\beta(x_0), V)$ where
  $V = (X^T W' X)^{-1} W(x_0) X^T W X W(x_0) (X^T W' X)^{-1}$,
  $W = \text{diag}(p(x_1 - x_0; \beta(x_0))(1 - p(x_1 - x_0; \beta(x_0))), \ldots,$
  $p(x_N - x_0; \beta(x_0)(1 - p(x_N - x_0; \beta(x_0)))$
- Ex. (South Africa Heart Disease) Response variable is chd which is binary indicator, response is sbp and obesity respectively.
- logit $\left[Pr(\text{chd} \,|\, X)\right] = \beta_0(x_0)$, Thus $100(1 - \alpha)\,\%$ CI of $f(x_0)$: $\hat{\beta}_0(x_0) \pm 2\sqrt{V_{11}}$

Local Likelihood and Other Models