Kernel SVM (MSBA 7027)

Zhengli Wang

Faculty of Business and Economics
The University of Hong Kong
2023

Outline

- Kernel
- Local Regression
- SVM

Recap

- Regression

$$\mathbb{E}(Y|X)) = f(X)$$

- Classification

$$log\frac{P(G=1|X)}{P(G=0|X)} = f(X)$$

- We want to learn $f(X)$ from the training set $(x_1, y_1), \ldots, (x_N, y_N)$
- When $f(X) = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p$
  - Multiple linear regression
  - Logistic regression

Recap

- When $f(X) = \beta_0 + \beta_1 h_1(X) + \ldots + \beta_m h_m(X)$ with known h's
  - Cubic spline, natural cubic spline
  - Logistic regression with known basis function
- When f(X) unknown
  - Smoothing spline
  - Nonparametric logistic regression
- When $f(X) = \beta_0 + f_1(X_1) + \ldots + f_p(X_p)$
  - GAM

Recap: K-nearest-neighbor (KNN)

- The k-NN estimate is a direct estimate of the conditional expectation

$$f(x_0) = \mathbb{E}(Y|X = x_0)$$

$$\hat{f}(x_0) = Ave\{y_i : x_i \in \mathcal{N}_k(x_0)\}$$

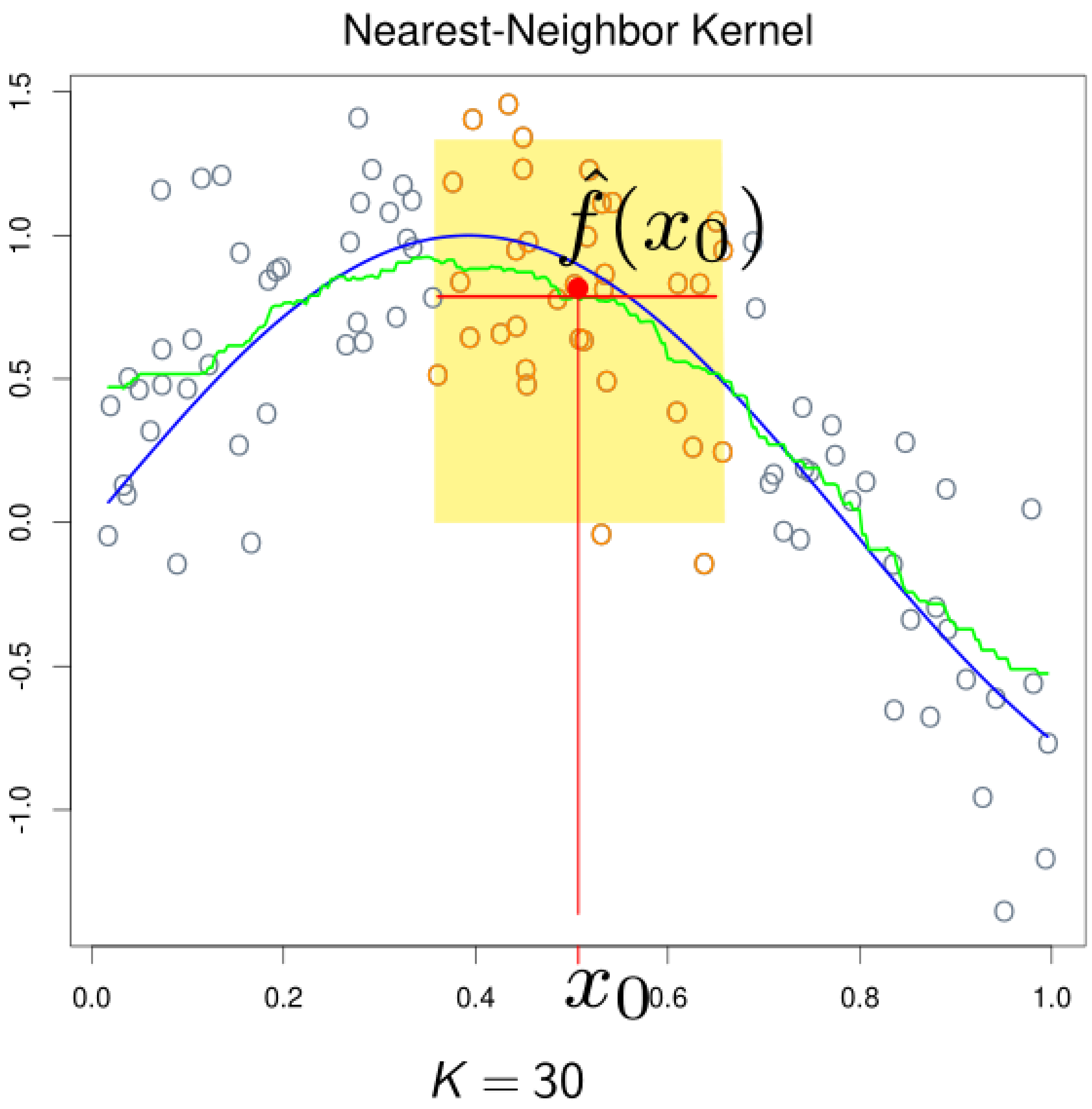
where $\mathcal{N}_k(x_0)$ is the set of k training points nearest to the query point x_0 in squared distance.

- Another technique to estimate $f(X)$: This estimator based on local information of x_0, where the value of $f(x_0)$ is of interest.

K-NN example: Discontinuous

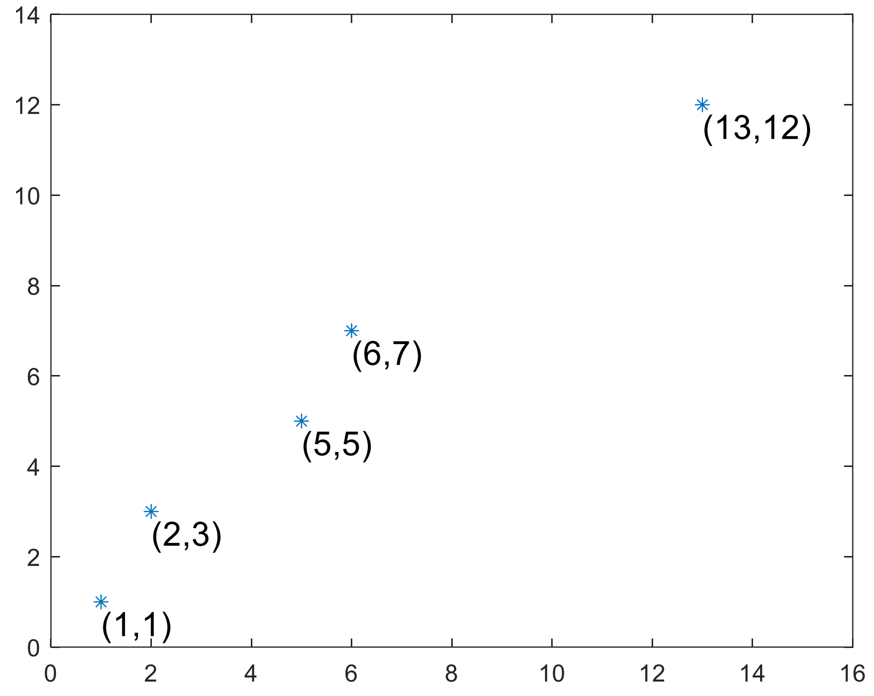
Blue curve: true relationship

Green curve: kNN (k = 30)



Kernel Smoothing: fitting a continuous/smooth curve

- Equal weights $\Rightarrow$ discontinuity
  - $\hat{f}(x_0) = \sum_{i=1}^{N} w_i \, y_i$, where $\sum_{i=1}^{N} w_i = 1$
  - E.g. K = 3, weight w_i is either 1/3 or 0



Kernel Smoothing: fitting a continuous/smooth curve

- Equal weights $\Rightarrow$ discontinuity
  - $\hat{f}(x_0) = \sum_{i=1}^{N} w_i\, y_i$, where $\sum_{i=1}^{N} w_i = 1$
  - E.g. K = 3, weight w_i is either 1/3 or 0
- Tentative Modification: $w_i \propto K(x_0, x_i)$ continuous, measures how far x_0 and x_i is
- Nadaraya-Watson (NW) Estimate: weighted average

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

where the weights are given by the kernel function

$$K_\lambda(x_0, x_i) = D\left(\frac{|x_i - x_0|}{\lambda}\right)$$

- λ is called window size, bandwidth, window width etc
  - Intuitively, think of λ as the standard deviation in the normal dist.

Kernel Smoothing: Example

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

$$K_\lambda(x_0, x_i) = D(|x_i - x_0|), \quad D(t) = \begin{cases} 1 - |t| & \text{if } |t| \le 1 \\ 0 & \text{Otherwise} \end{cases}$$

- Training observations (1, 2), (1.5, 4), (2.5, 6)

- **Critical points: (0.5, 2), (1, 8/3), (1.5, 10/3), (2, 5), (2.5, 6)**

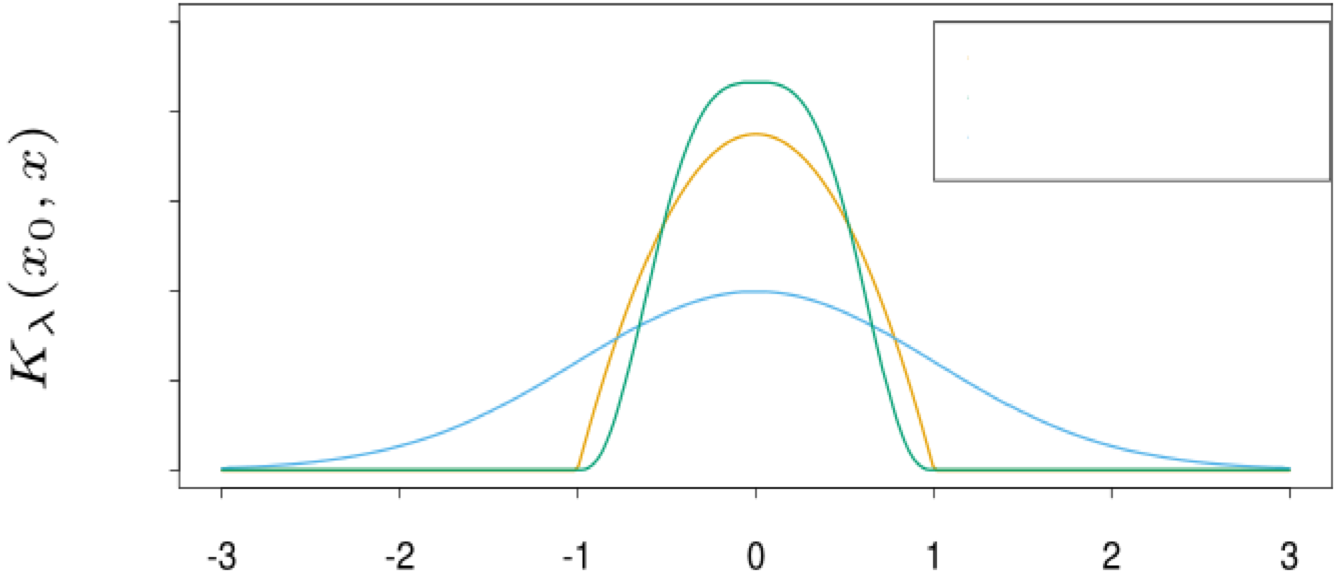
What is a kernel?

- For now, think of it as a function K, which depends on two inputs: x & x'
  - $K(x,x')$ Measures how close (similar) x and x' is
  - The farther away x and x' is, the smaller K is

Examples of Kernel Function

- Epanechnikov kernel $$D(t) = \begin{cases} \frac{3}{4}(1 - t^2) & if\ |t| \le 1 \\ 0 & Otherwise \end{cases}$$

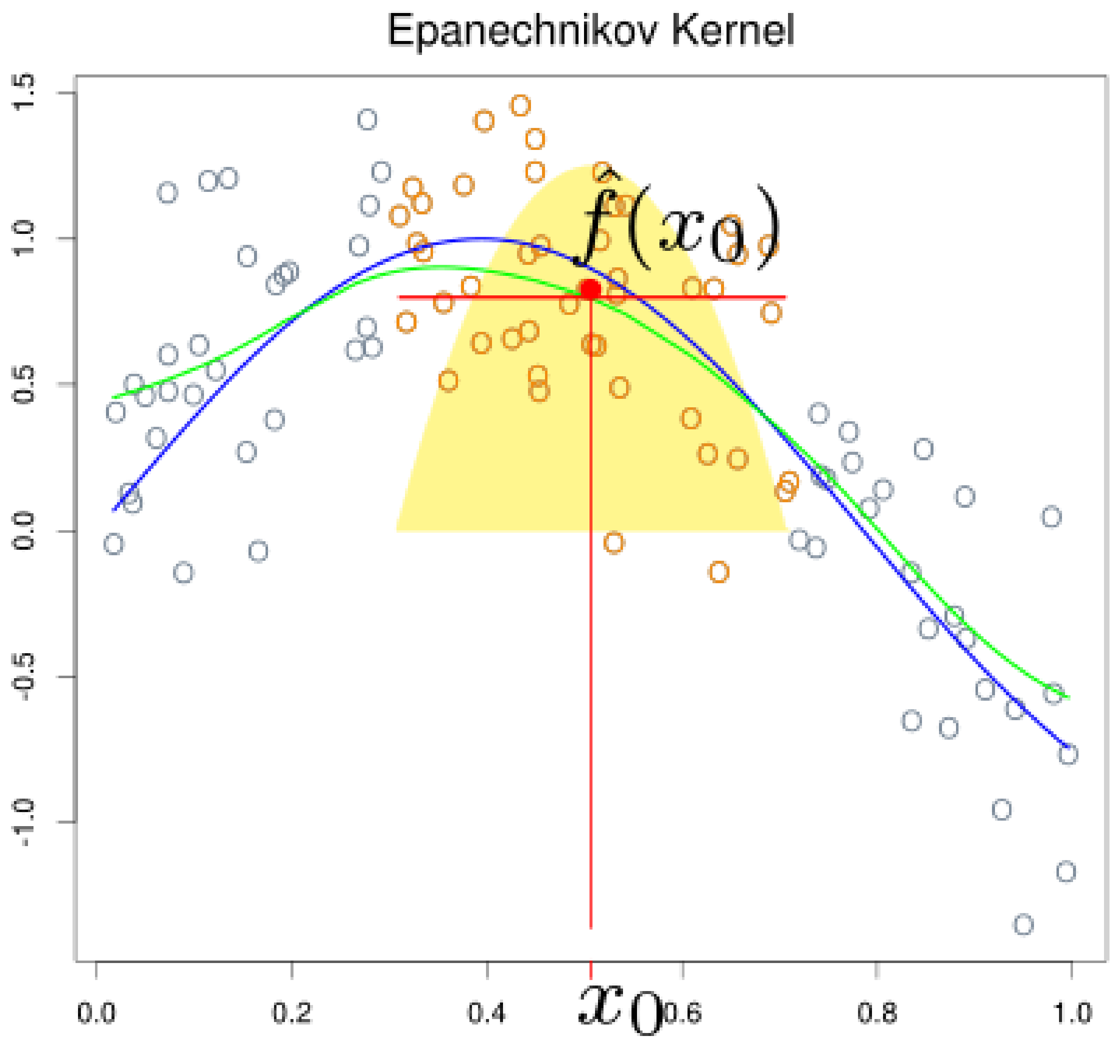
- Tri-cube kernel $$D(t) = \begin{cases} (1 - |t^3|)^3 & if\ |t| \le 1 \\ 0 & Otherwise \end{cases}$$



Epanechnikov kernel

Blue curve: true relationship

Green curve: NW Estimate with Epanechnikov kernel



However, there is a problem near boundary, what is it?

Local regression

- Combines linear regression and kernel
- At point x_0 (Weighted LS)

$$\min_{\beta_0 \beta_1} \sum_{i=1}^{N} K_\lambda(x_0, x_i)(y_i - \beta_0 - \beta_1 x_i)^2$$

- Fitted value

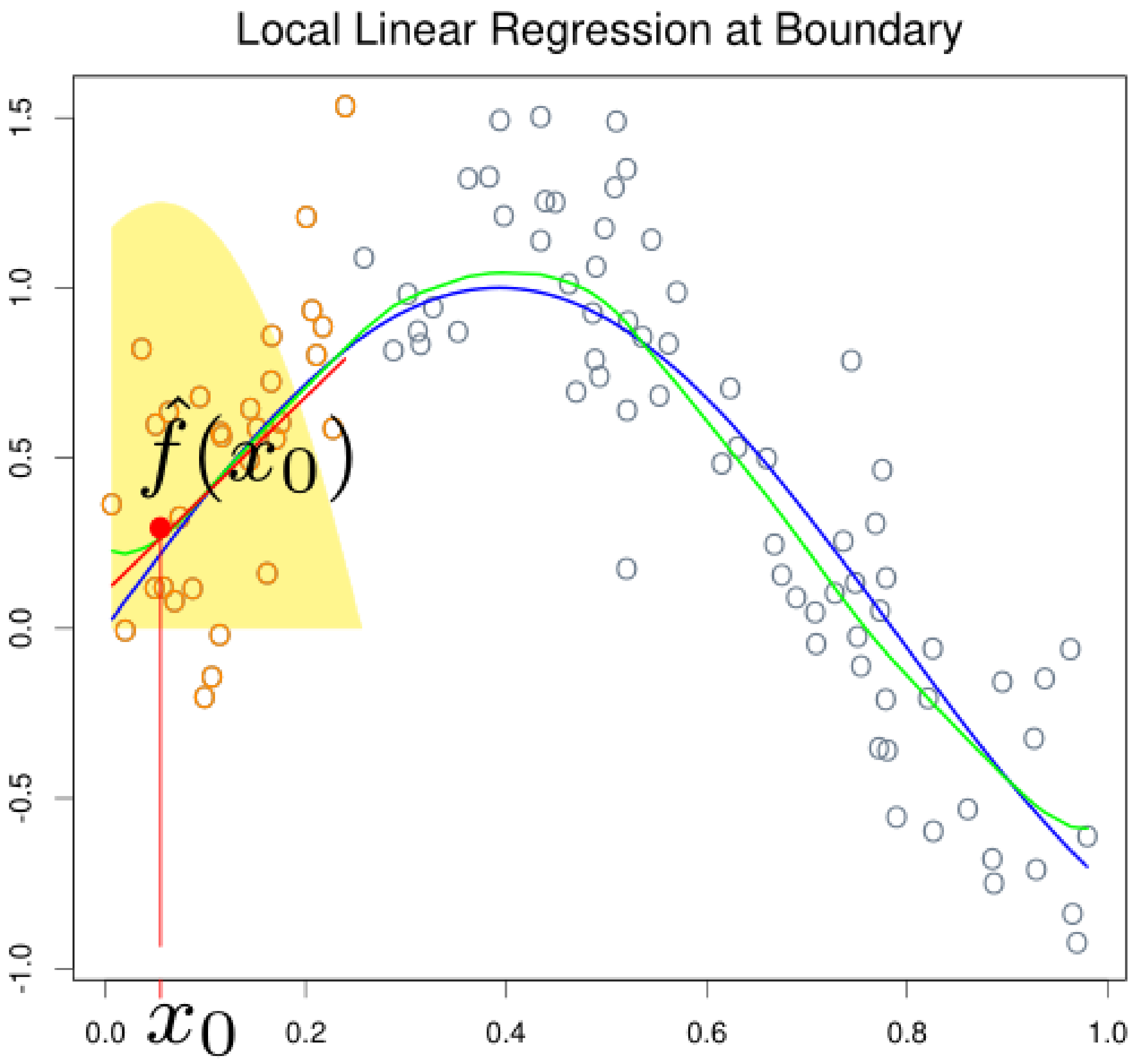
$$\hat{f}(x_0) = \hat{\beta}_0 + \hat{\beta}_1 x_0$$

- In fact, NW Est. is a special case of local regression, why?

Local regression

- Solution to: $\min_{\beta_0} \sum_{i=1}^{N} (y_i - \beta_0)^2$?
  - Ans: $\frac{\sum_{i=1}^{N} y_i}{N}$
- Solution to: $\min_{\beta_0} \sum_{i=1}^{N} w_i (y_i - \beta_0)^2$?
  - Ans: $\frac{\sum_{i=1}^{N} w_i y_i}{\sum_{i=1}^{N} w_i}$

Local regression: Bias removal

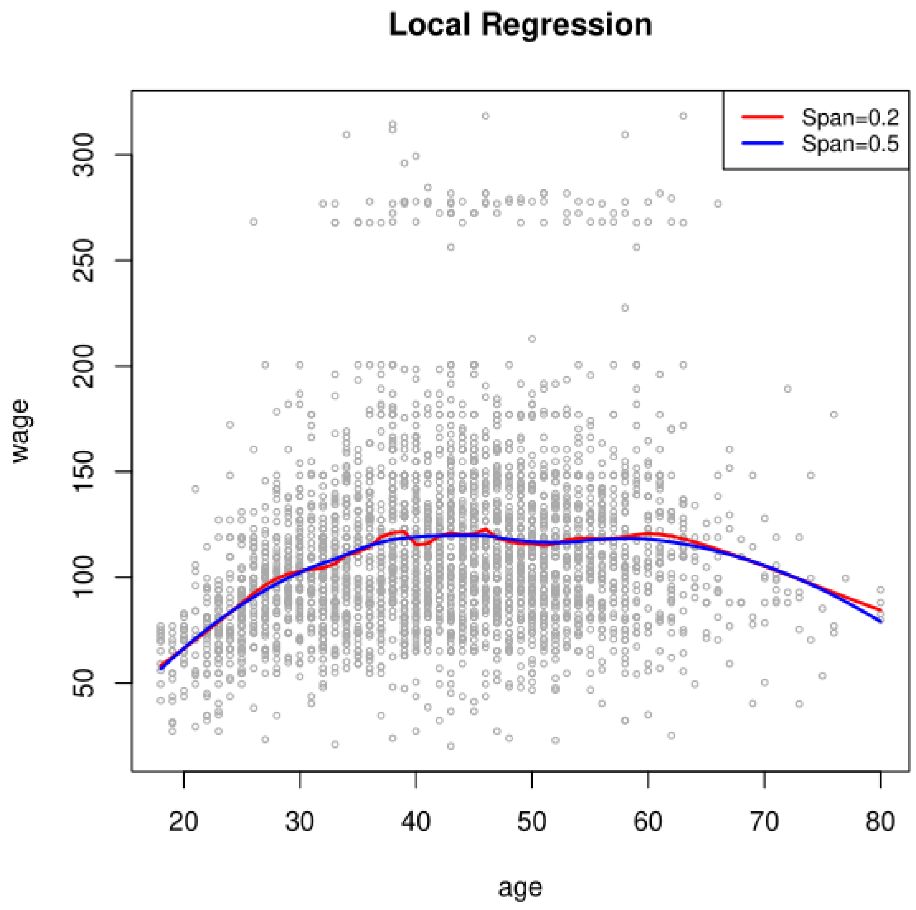


Selecting the Width of the Kernel

- In each of the kernels, K_λ, λ is a parameter that controls its width
  - Epanechnikov or tri-cube kernel, the radius of the support region
  - Gaussian kernel, the standard deviation
  - K-NN, analogous to #[nearest neighbors]
- Bias-variance tradeoff for local averages
  - Large λ vs Small λ

Local regression

Look at some code examples



Local logistic regression: Similar

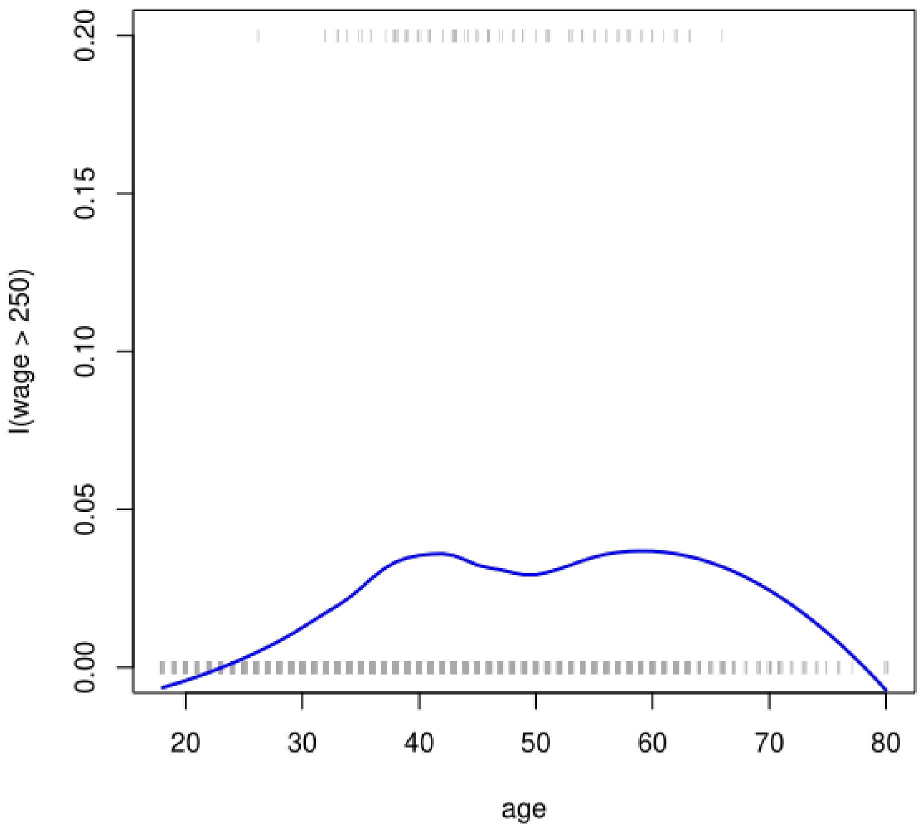
- The log likelihood $l(y_i, x_i; \beta)$ can be localized at the query point x_0. We choose para. β to maximize the kernel-weighted log likelihood

$$max_\beta \sum_{i=1}^{N} K_\lambda(x_0, x_i)\, l(y_i, x_i; \beta)$$

- Recall in MSBA7002
  - $l(y_i, x_i; \beta) = \ln p(x_i)$ if $y_i = 1$; $l(y_i, x_i; \beta) = \ln(1 - p(x_i))$ if $y_i = 0$
  - $log \frac{p(x_i)}{1 - p(x_i)} = \beta_0 + \beta_1 x_i$, and note $\beta = (\beta_0, \beta_1)$

Example: Local logistic regression

Look at some code examples



Local Regression for p-dim

- $X \in \mathbb{R}^P$

- Criterion

$$\min_{\beta_0, \beta_1, \dots \beta_p} \sum_{i=1}^{N} K_\lambda(x_0, x_i) \left(y_i - \beta_0 - \beta_1 x_{i1} - \cdots - \beta_p x_{ip}\right)^2$$

where

$$K_\lambda(x_0, x) = D\left(\frac{||x - x_0||}{\lambda}\right)$$

Local Regression for *p*-dim

- $X \in \mathbb{R}^P$
- Note: difference from GAM
  - GAM: wage = $\beta_0 + f_1(year) + f_2(age)$
  - Local regression: wage = *f(year, age)*, can consider interaction terms
- However, local regression does not perform well when p is large
  - When p large: curse of dimensionality, in most parts of the space → few data points
  - p = 2 is usually OK, and may perform well

Local regression

Code examples