Week 4: Nonparametric Regression

MATH-517 Statistical Computation and Visualization

Linda Mhalla

2023-10-13

KDE

One-dimensional KDE (from last week):

$$\hat{f}(x) = \frac{1}{nh_n} \sum_{i=1}^{n} K\left(\frac{X_i - x}{h_n}\right)$$

Multidimensional generalization (separable) when $X_1, \dots, X_n \in \mathbb{R}^d$:

$$\hat{f}(\mathbf{x}) = \frac{1}{nh^d} \sum_{i=1}^{n} K\left(\frac{X_{i,1} - x_1}{h}\right) \cdot \ldots \cdot K\left(\frac{X_{i,d} - x_d}{h}\right)$$

Section 1

Non-parametric Regression

Non-parametric Regression Setup

- we observe i.i.d. copies of a bivariate random vector $(X, Y)^\top$
  - a random sample $(X_1, Y_1)^\top, \dots, (X_n, Y_n)^\top$
- the response variable Y is related to the covariate X through

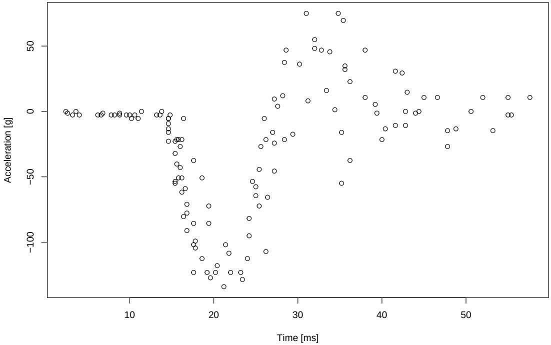
$$Y_i = m(X_i) + \epsilon_i, \quad \mathbb{E}(\epsilon_i) = 0 \quad \text{and } \mathrm{var}(\epsilon_i) = \sigma^2$$

- we are interested in the conditional expectation of Y given X, i.e., the regression function

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

- we want to avoid parametric assumptions

Data Example



- head acceleration Y depending on time X in a simulated motorcycle accident used to test crash helmets

Local average estimator

Goal: estimate $m(x) = \mathbb{E}(Y \big| X = x)$ from $(X_1, Y_1)^\top, \ldots, (X_n, Y_n)^\top$ i.i.d.

Since $m(x) = \mathbb{E}(Y|X = x)$, one can estimate $m(x)$ by averaging the Y_is for which X_i is "close" to x

$\Rightarrow$ different averaging methods and different measures of closeness yield different estimators

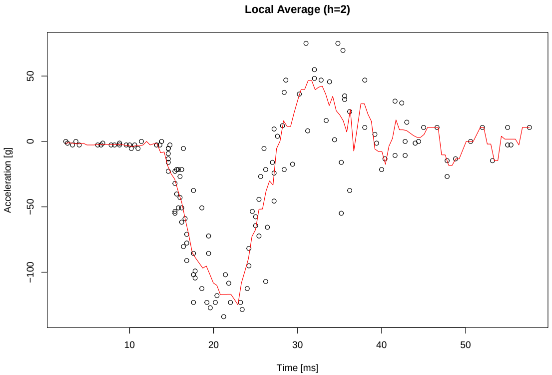
The local average estimator is

$$\widehat{m}_n(x) = \frac{\sum_{i=1}^n I(x - h < X_i \leq x + h) Y_i}{\sum_{i=1}^n I(x - h < X_i \leq x + h)}$$

$$= \frac{\sum_{i=1}^n \frac{1}{2} \mathbb{1}_{[-1,1)}\left(\frac{x - X_i}{h}\right) Y_i}{\sum_{i=1}^n \frac{1}{2} \mathbb{1}_{[-1,1)}\left(\frac{x - X_i}{h}\right)},$$

for $h > 0$

Local average estimator



Local Constant Regression

Since $m(x) = \int_{\mathbb{R}} y f_{Y|X}(y|x) dy = \frac{\int_{\mathbb{R}} y f_{X,Y}(x,y) dy}{f_X(x)}$ and we can now estimate densities, let's plug in those estimators

$$\hat{f}_X(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right)$$

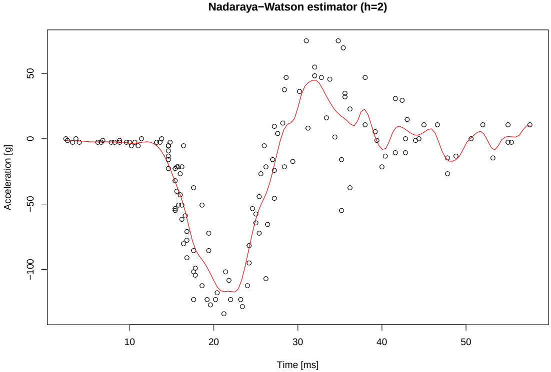
$$\hat{f}_{X,Y}(x, y) = \frac{1}{nh^2} \sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right) K\left(\frac{y - Y_i}{h}\right)$$

to obtain

$$\widehat{m}(x) = \frac{\sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right) Y_i}{\sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right)}$$

$\Rightarrow$ The "box" kernel is replaced by a general kernel and yields the so-called Nadaraya–Watson kernel estimator

Local Constant Regression



Local Constant Regression

The Nadaraya–Watson kernel estimator

$$\widehat{m}(x) = \frac{\sum_{i=1}^{n} K\left(\frac{X_i - x}{h}\right) Y_i}{\sum_{i=1}^{n} K\left(\frac{X_i - x}{h}\right)} = \sum_{i=1}^{n} W_i^0(x) Y_i$$

is a weighted mean of the Y_i. Thus, it must be a solution to a weighted least squares:

$$\widehat{m}(x) = \arg\min_{\beta_0 \in \mathbb{R}} \sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right) (Y_i - \beta_0)^2$$

For a fixed x, this is a weighted intercept-only regression, with weights given by the kernel $\Rightarrow$ estimate suffers from boundary bias

What if we went for better than intercept-only regression?

Local Polyomial Regression

The aim is to find the local regression parameters $\beta(x)$ s.t.

$$\hat{\beta}(x) = \arg\min_{\beta \in \mathbb{R}^{p+1}} \sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right)\{Y_i - \beta_0 - \beta_1(X_i - x) - ... - \beta_p(X_i - x)^p\}^2$$

Local Polyomial Regression

The aim is to find the local regression parameters $\beta(x)$ s.t.

$$\hat{\beta}(x) = \arg\min_{\beta \in \mathbb{R}^{p+1}} \sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right) \{Y_i - \beta_0 - \beta_1 (X_i - x) - ... - \beta_p (X_i - x)^p\}^2$$

Why does this make sense?

Recall that the aim is to estimate $m(x) = \mathbb{E}(Y|X = x)$ and hence to minimize the RSS

$$\sum_{i=1}^{n} \{Y_i - m(X_i)\}^2$$

A Taylor expansion of m for x close to X_i is

$$m(X_i) \approx m(x) + (X_i - x)m'(x) + \frac{(X_i - x)^2}{2!} m''(x) + ... + \frac{(X_i - x)^p}{p!} m^p(x),$$

Local Polyomial Regression

The RSS can be rewritten as

$$\sum_{i=1}^{n} \left\{ Y_i - \sum_{j=0}^{p} \frac{m^j(x)}{j!} (X_i - x)^j \right\}^2$$

Thus, $\hat{\beta}_j(x)$ estimates $\frac{m^{(j)}(x)}{j!}$

- $\widehat{m}(x) = \hat{\beta}_0(x)$
- $\widehat{m'}(x) = \hat{\beta}_1(x)$

Finally, add a weighting kernel to make the contributions of X_i dependent on their distance to x

$\Rightarrow \hat{\beta}$ becomes the solution to a weighted least squares problem

$$\hat{\beta} = \arg\min_{\beta \in \mathbb{R}^{p+1}} (\mathbf{Y} - \mathbf{X}\beta)^\top \mathbf{W} (\mathbf{Y} - \mathbf{X}\beta) = (\mathbf{X}^\top \mathbf{W} \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{W} \mathbf{Y},$$

where $\mathbf{W}$ is a diagonal matrix with entries depending on the kernel!

Local Linear Regression

Choosing the order $p = 1$ leads to the local linear estimator

$$(\hat{\beta}_0(x), \hat{\beta}_1(x)) = \arg\min_{\beta \in \mathbb{R}^2} \sum_{i=1}^{n} \{Y_i - \beta_0 - \beta_1(X_i - x)\}^2 K\left(\frac{x - X_i}{h}\right),$$

It can be shown that $\hat{m}(x) = \hat{\beta}_0(x) = \sum_{i=1}^{n} w_{ni}(x) Y_i$, where

$$w_{ni}(x) = \frac{1}{nh} \frac{K\left(\frac{x - X_i}{h}\right)\left\{S_{n,2}(x) - (X_i - x)\, S_{n,1}(x)\right\}}{S_{n,0}(x) S_{n,2}(x) - S_{n,1}^2(x)}$$

with $S_{n,k}(x) = \frac{1}{nh} \sum_{i=1}^{n} (X_i - x)^k K\left(\frac{X_i - x}{h}\right)$

- $\sum_{i=1}^{n} w_{ni}(x) = 1$

$\Rightarrow \hat{m}$ is a linear smoother, i.e., $\forall x$, it can be defined by a weighted average: $\hat{m}(x) = \sum_{i=1}^{n} l_i(x) Y_i$ (valid for Nadaraya–Watson and any p)

Visualization

A shiny App can be found here

Bias and Variance

For local linear regression, similarly to KDE and under regularity assumptions on m, f, K, h, and nh,

$$\text{bias}\{\widehat{m}(x)\} = \frac{1}{2}m''(x)h_n^2 \int z^2 K(z)dz + o_P(h_n^2)$$
$$\text{var}\{\widehat{m}(x)\} = \frac{\sigma^2(x)}{f_X(x)} \frac{\int \{K(z)\}^2 dz}{nh_n} + o_P\left(\frac{1}{nh_n}\right)$$

where $\sigma^2(x) = \text{var}(Y_1|X_1 = x)$ is the conditional/local variance

This implies that

- the bias depends on the curvature of m: negative for concave and positive for convex regions
- the variance decreases at a rate inversely proportional to the effective sample size nh

For other orders, similar expressions can be obtained

Nadaraya–Watson vs Local Linear estimator

It can be shown that, under the same smoothing conditions on $f(x)$ and $m(x)$, the Nadaraya–Watson estimator $\tilde{m}$

- has the same variance as the local linear estimator $\hat{m}$
- has bias

$$\text{bias}\{\tilde{m}(x)\} = h_n^2 \left\{ \frac{1}{2} m''(x) + m'(x) \frac{f'(x)}{f(x)} \right\} \int z^2 K(z) dz + o_P(h_n^2)$$

$\Rightarrow$ At the boundary points, the NW estimator bears high value due to the large absolute value of $f'(x)/f(x)$

$\Rightarrow$ Local linear estimation has no boundary bias at it does not depend on $f(x)$ (no design bias)

Bandwidth Selection

Similarly to what we did last week with KDEs, we consider

$$MSE\{\widehat{m}(x)\} = \text{var}\{\widehat{m}(x)\} + \left[\text{bias}\{\widehat{m}(x)\}\right]^2$$

and, dropping the little-o terms, we obtain

$$AMSE\{\widehat{m}(x)\} = \frac{\sigma^2(x)\int\{K(z)\}^2 dz}{f_X(x) n h_n} + \frac{1}{4}\{m''(x)\}^2 h_n^4 \left(\int z^2 K(z) dz\right)^2.$$

Now, a local bandwidth choice can be obtained by optimizing AMSE. Taking derivatives and setting them to zero, we obtain

$$h_{opt}(x) = n^{-1/5}\left[\frac{\sigma^2(x)\int\{K(z)\}^2 dz}{\left\{m''(x)\int z^2 K(z) dz\right\}^2 f_X(x)}\right]^{1/5}$$

Bandwidth Selection

$$h_{opt}(x) = n^{-1/5} \left[\frac{\sigma^2(x) \int \{K(z)\}^2 dz}{\left\{m''(x) \int z^2 K(z) dz\right\}^2 f_X(x)} \right]^{1/5}$$

This is somewhat more complicated compared to the KDE case, because we have to estimate

- the marginal density $f_X(x)$,
  - let's say that we already know how to do this, e.g., by KDE even though that requires a choice of yet another bandwidth
- the local variance function $\sigma^2(x)$, and
- the second derivative of the regression function $m''(x)$

Again, like in the case of KDEs, the global bandwidth choice can be obtained by integration:

- calculate $AMISE(\widehat{m}) = \int AMSE\{\widehat{m}(x)\}dx$, and
- set $h_{AMISE} = \arg\min_{h>0} AMISE(\widehat{m})$

Rule of Thumb Plug-in Algorithm

Replace the unknown quantities in

$$h_{AMISE} = n^{-1/5}\left[\frac{\int K^2(z)dz \int \sigma^2(x)dx}{\int z^2K(z)dz \int \{m''(x)\}^2 f_X(x)dx}\right]^{1/5}$$

by parametric OLS estimators

- Assume homoscedasticity and a quartic kernel, then

$$h_{AMISE} = n^{-1/5}\left(\frac{35\sigma^2|supp(X)|}{\theta_{22}}\right)^{1/5}, \quad \theta_{22} = \int \{m''(x)\}^2 f_X(x)dx$$

- Block the sample in N blocks and fit, in each block j, the model

$$y_i = \beta_{0j} + \beta_{1j}x_i + \beta_{2j}x_i^2 + \beta_{3j}x_i^3 + \beta_{4j}x_i^4 + \epsilon_i$$

  to obtain estimate $\hat{m}_j(x_i) = \hat{\beta}_{0j} + \hat{\beta}_{1j}x_i + \hat{\beta}_{2j}x_i^2 + \hat{\beta}_{3j}x_i^3 + \hat{\beta}_{4j}x_i^4$

Rule of Thumb Plug-in Algorithm

- Estimate the unknown quantities by

$$\hat{\theta}_{22}(N) = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{N}\hat{m}_j''(X_i)\hat{m}_j''(X_i)\mathbb{1}_{X_i\in\mathcal{X}_j}$$

$$\hat{\sigma}^2(N) = \frac{1}{n-5N}\sum_{i=1}^{n}\sum_{j=1}^{N}\{Y_i - \hat{m}_j(X_i)\}^2\mathbb{1}_{X_i\in\mathcal{X}_j}$$

Remark Unknown quantities can also be replaced by non-parametric estimates, using a pilot bandwidth (see lecture notes for details).

Why Local Linear?

Bias and variance can be calculated similarly also for higher order local polynomial regression estimators. In general:

- bias decreases with an increasing order
- variance increases with increasing order, but only for $p = 2k + 1 \rightarrow p + 1$, i.e., when increasing an odd order to an even one

For this reason, odd orders are preferred to even ones

- $p = 1$ is easy to grasp as it corresponds to locally fitted simple regression line
- increasing p has a similar effect to decreasing the bandwidth h
    - hence $p = 1$ is usually fixed and only h is tuned

Section 2

Other Smoothers

Smoothing Splines

Consider the optimization problem

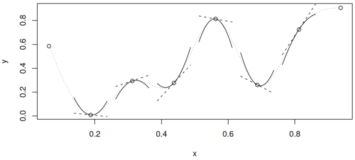
$$\arg\min_{g \in C^2} \sum_{i=1}^{n} \{Y_i - g(X_i)\}^2 + \lambda \int \{g''(x)\}^2 dx$$

- measure of closeness to the data, and
- smoothing penalty: $\lambda > 0$ controls the trade-off between fit and smoothness

Unique solution: the **natural cubic spline**

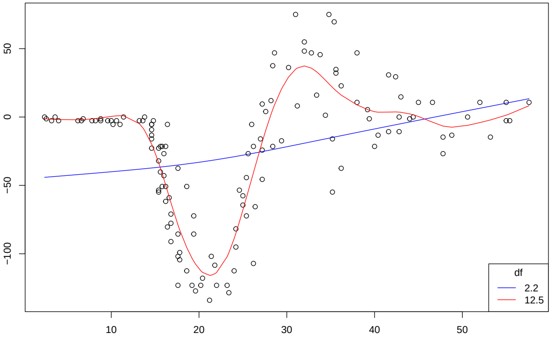
- piece-wise cubic polynomial between
- knots at X_i, $i = 1, \ldots, n$
- two continuous derivatives at the knots
- $\hat{m}''(x_1) = \hat{m}''(x_n) = 0$

$\Rightarrow$ it has n free parameters



Source: Wood (2017)

Smoothing Spline: Example



Smoothing Splines

A natural cubic spline g with n knots can be expressed w.r.t. the natural cubic spline basis $\{e_j\}$ (a set of basis functions) as

$$m(x) = \sum_{j=1}^{n} \gamma_j e_j(x)$$

Let

- $\gamma = (\gamma_1, \dots, \gamma_n)^\top \in \mathbb{R}^n$ be unknown coefficients
- $E := (e_{ij}) := \{e_j(x_i)\}_{i,j=1}^{n} \in \mathbb{R}^{n \times n}$ and
- $\Omega = (\omega_{ij}) \in \mathbb{R}^{n \times n}$ with $\omega_{ij} = \int e_i''(x) e_j''(x) dx$

Then, the optimisation problem becomes

$$\sum_{i=1}^{n} \{Y_i - m(X_i)\}^2 + \lambda \int \{m''(x)\}^2 dx \quad \equiv \quad (Y - E\gamma)^\top (Y - E\gamma) + \lambda \gamma^\top \Omega \gamma$$

Smoothing Splines

The solution is obtained in closed form

$$\hat{\gamma} = (E^\top E + \lambda\Omega)E^\top Y$$

The fitted values are

$$\widehat{Y} = (\hat{m}_\lambda(x_1), \dots, \hat{m}_\lambda(x_n))^\top = E\hat{\gamma} = S_\lambda Y, \quad \text{with } S_\lambda = E(E^\top E{+}\gamma\Omega)E^\top$$

$\Rightarrow$ smoothing splines are linear smoothers

The matrix S_λ is the hat matrix and $tr(S_\lambda)$ plays the role of the degrees of freedom (how many effective parameters you have in the model)

- Although there are n unknown coefficients, many are shrunken towards zero through the smoothness/roughness penalty
- λ encodes the bias-variance trade-off ($\lambda = 0:$ very rough, $\lambda = \infty$: very smooth)
- λ is chosen by CV (next week's lecture)

Orthogonal Series: Regression Splines

- take a pre-defined set of orthogonal functions $\{e_j\}_{j=1}^{\infty}$
  - customarily some basis, e.g. Fourier basis, B-splines, etc.
- truncate it to $\{e_j\}_{j=1}^{p}$
- approximate $m(x) \approx \sum_{j=1}^{p} \gamma_j e_j(x)$

Then estimate m by least-squares:

$$\arg\min_{\gamma \in \mathbb{R}^p} \sum_{i=1}^{n} \left[Y_i - \gamma_1 e_1(x_i) - \ldots - \gamma_p e_p(x_i) \right]^2$$

- just a single linear regression
- no penalty term, simplicity achieved via truncation
  - bias-variance trade-off controlled by the choice of p
  - can be related to smoothness when e_j's get more wiggly with increasing j (which is typical for most bases)
  - choice of location of knots is critical but tricky too (not needed with smoothing splines)

Assignment 3 [5 %]

Go to Assignment 3 for details.