STAT40810 — Stochastic Models

Brendan Murphy

Week 3

Kernel Smoothing

Kernel Smoothing

- Suppose we have a dataset with values $(x_1, y_1), \ldots, (x_n, y_n)$.
- We want to study the relationship between the x and the y values.
- In other words, we want to estimate the function $f(x_i)$, where $y_i = f(x_i) + \epsilon_i$.
- A crucial input into kernel smoothing is the kenrel and the *bandwidth* (h).
- The bandwidth is a number greater than 0; it controls the smoothness of the estimate $\hat{f}(x)$.

Kernel Function

- A kernel is a function $K(x)$ such that

$$K(x) > 0 \text{ and } \int_{\mathcal{X}} K(x)dx = 1.$$

- Most kernels are also symmetric, so $K(x) = K(-x)$.
- Some commonly used kernels are:

$$K(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) \qquad \text{Gaussian Kernel}$$

$$K(x) = \begin{cases} 1 - |x| & \text{for } |x| < 1 \\ 0 & \text{otherwise} \end{cases} \qquad \text{Triangular Kernel}$$

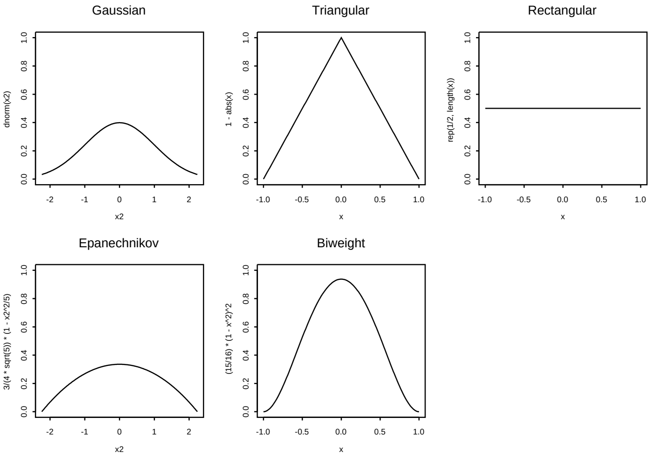
Yet More Kernels

$$K(x) = \begin{cases} \frac{1}{2} & \text{for } |x| < 1 \\ 0 & \text{otherwise} \end{cases}$$ Rectangular Kernel

$$K(x) = \begin{cases} \frac{3}{4\sqrt{5}}\left(1 - \frac{x^2}{5}\right) & \text{for } |x| < \sqrt{5} \\ 0 & \text{otherwise} \end{cases}$$ Epanechnikov Kernel

$$K(x) = \begin{cases} \frac{15}{16}\left(1 - x^2\right)^2 & \text{for } |x| < 1 \\ 0 & \text{otherwise} \end{cases}$$ Biweight Kernel

Various Kernels



Bandwidth

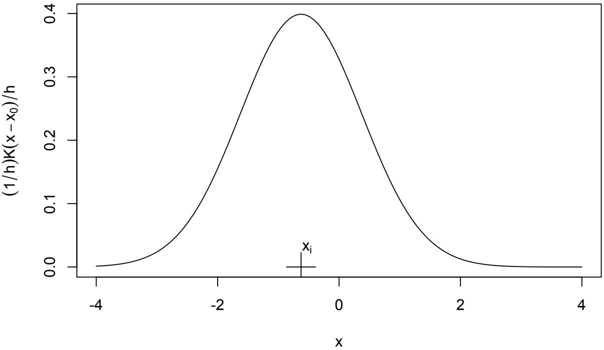
- If we take a kernel function $K(x)$, any x_0 and a bandwidth $h > 0$ and compute
$$\frac{1}{h} K\left(\frac{x - x_0}{h}\right).$$
- This has the effect of shifting the kernel to being centred on x_0 instead of 0.
- It also has the effect of:
  - Making the kernel narrower and taller, if $h < 1$.
  - Making the kernel wider and lower, if $h > 1$.
- The value of h is called the bandwidth.

Kernel Plot

- An example of what

$$\frac{1}{h}K\left(\frac{x-x_0}{h}\right)$$

looks like is:



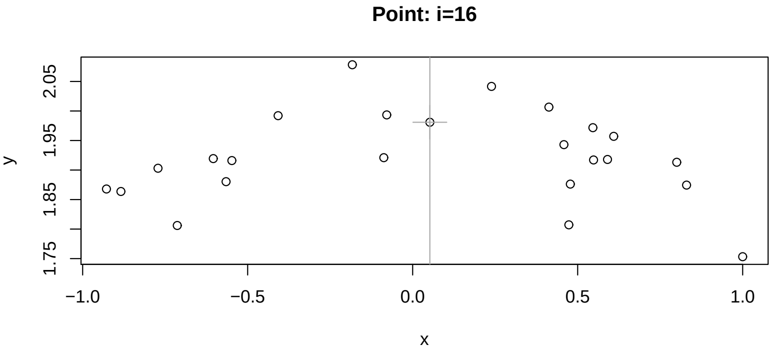
Kernel Smoothing

- Kernel smoothing requires us to specify a kernel and a bandwidth.
- The value of the response variable is predicted using $\hat{f}(x_i) = \hat{y}_i = \sum_{j=1}^{n} w_{ij} y_j$ where

$$w_{ij} = \frac{\frac{1}{h} K\left(\frac{x_i - x_j}{h}\right)}{\frac{1}{h} \sum_{k=1}^{n} K\left(\frac{x_i - x_k}{h}\right)}$$

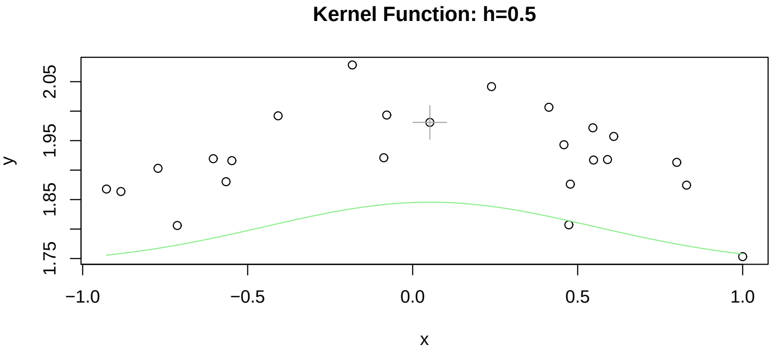
Kernel Smoothing Steps (1)

- Suppose we want to compute $\hat{f}(x_i)$. Suppose $h = 0.5$ and $n = 24$.



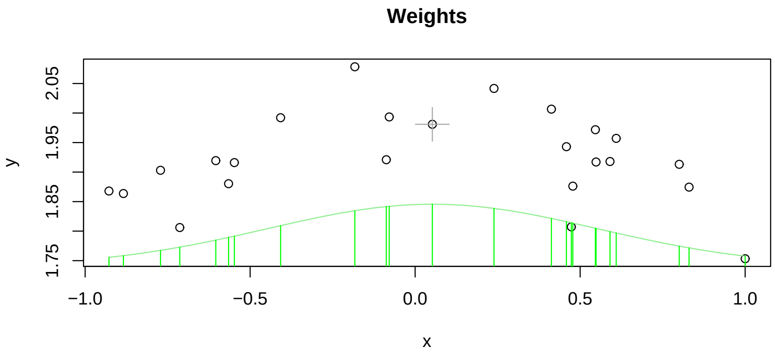
Kernel Smoothing Steps (2)

- The kernel function $(1/h)K\left(\frac{x-x_i}{h}\right)$ is computed. A Gaussian kernel is shown.



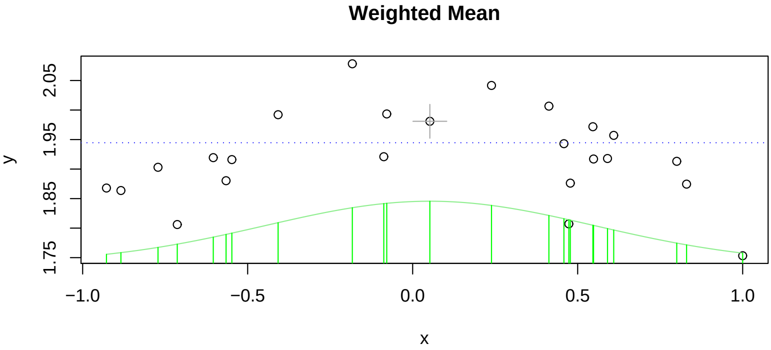
Kernel Smoothing Steps (3)

- The weights are computed.



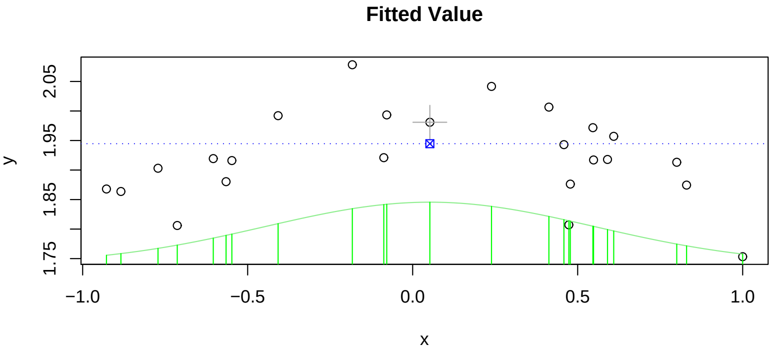
Kernel Smoothing Steps (4)

- A weighted mean is calculated



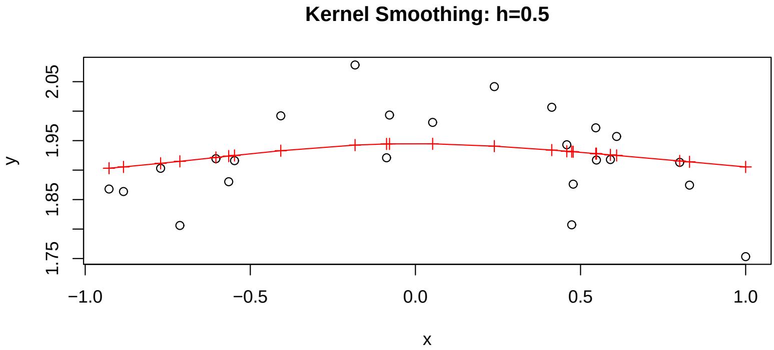
Kernel Smoothing Steps (5)

- The fitted value is found (it's just the weighted mean!)



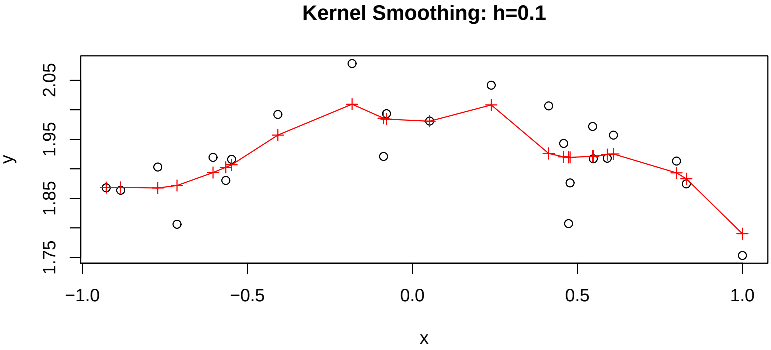
Kernel Smoothing Fits

- The steps are repeated for each $i = 1, 2, \ldots, n$. This gives the kernel smoothed curve.



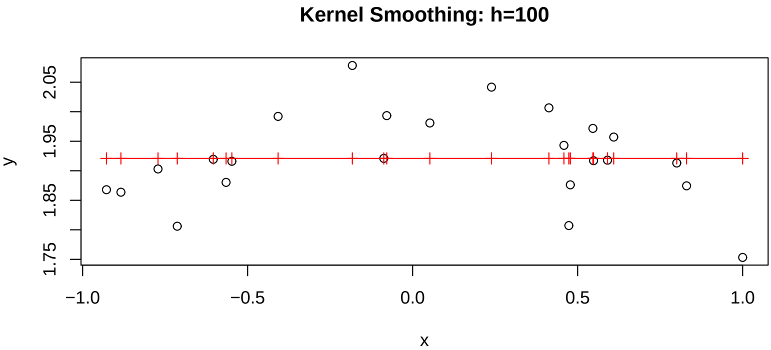
Kernel Smoothing (h=0.1)

- If the bandwidth is smaller, then we get a more “wiggly” fit.



Kernel Smoothing (h=100)

- If the span is very large, then we get a constant function, equal to the mean of the ys.



Variants

- Variants of kernel smoothing exist which use weighted polynomial regression instead of weighted means.
- Can you see the similarity with LOWESS?

Kernel Smoothing

- Kernel Smoothing is very easy to implement in R.
- Here's code to model the motorcycle data.

```
# Load the data
library(MASS)
data(mcycle)

# Plot the data
plot(mcycle)

# Fit a kernel smoothing
# The bandwidth needs to be specified
fit <- ksmooth(mcycle$times,mcycle$accel,kernel="normal",bandwidth=3)

# Add the fitted values to the plot
points(fit,pch=3,col="red")

# Assess fit using mean squared error (MSE)
MSE <- mean((mcycle$accel-fit$y)^2)
MSE
```