Statistical Modeling 2
Exercise 3

February 20, 2017

Basic concepts

$$\begin{aligned}
\mathrm{MSE}(\hat{f}, f) &= E\{[f(x) - \hat{f}(x)]^2\} \\
&= E[f(x)^2 + \hat{f}(x)^2 - 2f(x)\hat{f}(x)] \\
&= f(x)^2 + E[\hat{f}(x)^2] - E[\hat{f}(x)]^2 + E[\hat{f}(x)]^2 - 2f(x)E[\hat{f}(x)] \\
&= \{E[\hat{f}(x)^2] - E[\hat{f}(x)]^2\} + \{f(x)^2 + E[\hat{f}(x)]^2 - 2f(x)E[\hat{f}(x)]\} \\
&= V + B^2
\end{aligned}$$

Curve fitting by linear smoothing

A

In the least square estimate, we minimize:

$$L = \sum_{i=1}^{n} (\hat{\beta} x_i - y_i)^2$$

We take the derivative and set to 0:

$$\begin{aligned}
\partial L / \partial \hat{\beta} &= \sum_{i=1}^{n} 2(\hat{\beta} x_i - y_i) x_i = 0 \\
&\implies \hat{\beta}(\sum_{i=1}^{n} x_i^2) - \sum_{i=1}^{n} y_i x_i = 0 \\
&\implies \hat{\beta} = \frac{\sum_{i=1}^{n} y_i x_i}{\sum_{i=1}^{n} x_i^2}
\end{aligned}$$

We have that the prediction is:

$$\begin{aligned}\hat{f}(x^*) &= \hat{\beta}x^* \\ &= \frac{\sum_{i=1}^n x_i x^* y_i}{\sum_{i=1}^n x_i^2} \\ &= \sum_{i=1}^n w(x_i, x^*) y_i\end{aligned}$$

where

$$w(x_i, x^*) = \frac{x_i x^*}{\sum_{i=1}^n x_i^2}$$

This weight function weights examples by the product with predictors, using all the points in the dataset. The K nearest function weights the K nearest neighbors equally and ignore all other points.

B

Code: kernel.r
I use the function sin from -5 to 5 with 100 samples and try three bandwidths: $h = 0.01, h = 0.1, h = 1$.

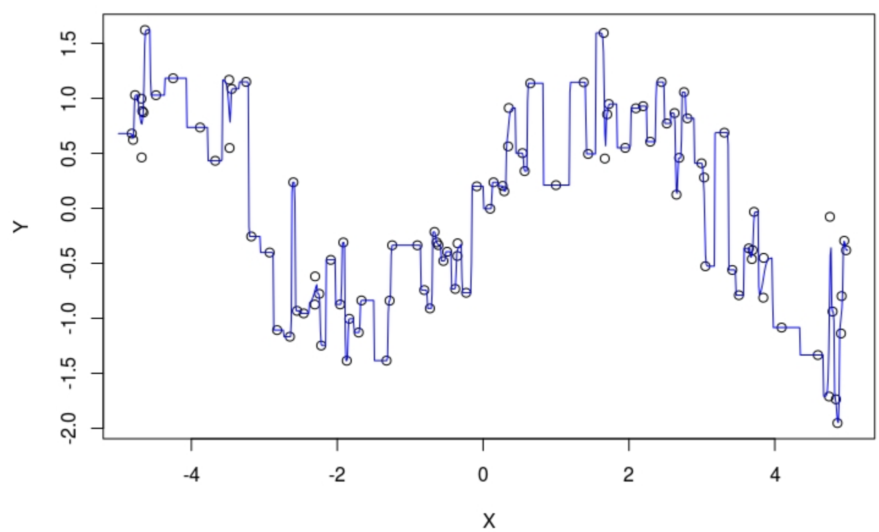

Figure 1: Bandwidth $h = 0.01$

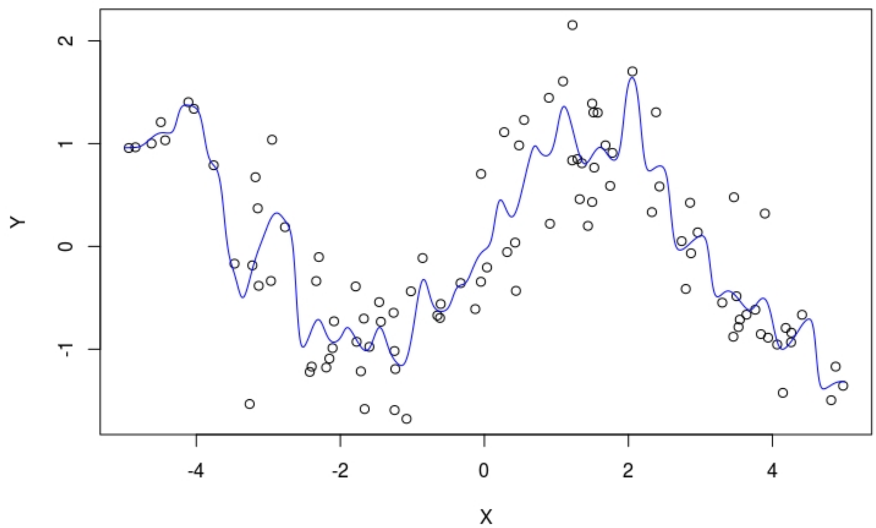

Figure 2: Bandwidth $h = 0.1$

The bandwidth h affects the smoothness of the prediction line. With smaller h, more weight is given to neighbor points and the line fluctuates. With larger h, more weight is given to points that are further away, resulting in a smoother line.

Cross Validation

B

For the smooth function, I use x^2; for the wiggly function, I use $sin(10x)$. 'Not so noisy' has a Normal noise of 0.05, 'noisy' has a Normal noise of 0.25. I generate 100 points in train and 100 points in test. The bandwidth h is varied from 0.01 to 0.20. For the smooth function with less noise, the RMSE is stable across different settings of h. But for other cases, using the right h seems to improve RMSE significantly.

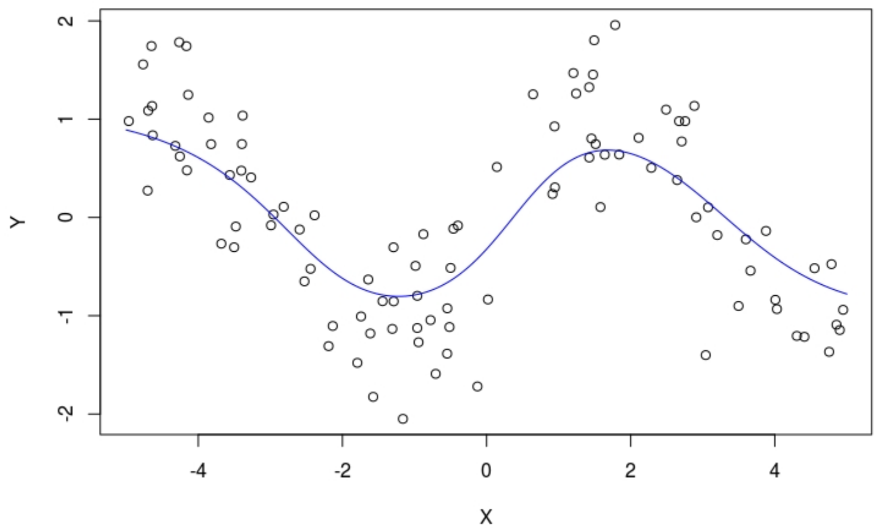

Figure 3: Bandwidth $h = 1$

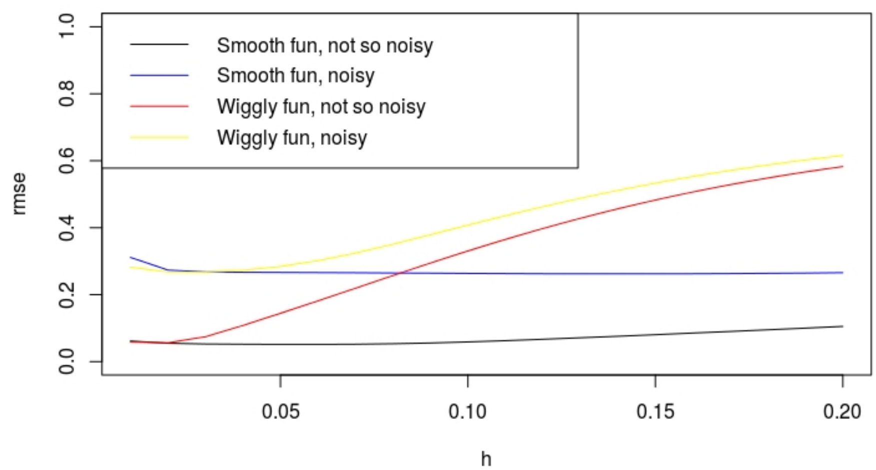

Figure 4: Cross Validation

Local polynomial regression

A

Let R_x be the matrix whose (i,j) entry is $(x_i - x)^{j-1}$. We have that

$$g_x(x_i; a) = R_{x,i}^T a$$

where a is the column vector of coefficients of the polynomial g and $R_{x,i}$ is the row i of R_x. We want to minimize:

$$\begin{aligned}
&\sum_{i=1}^{n} w_i \{y_i - g_x(x_i; a)\}^2 \\
&= \sum_{i=1}^{n} w_i \{y_i - R_{x,i}^T a\}^2 \\
&= \sum_{i=1}^{n} \frac{1}{h} K\left(\frac{x_i - x}{h}\right) \{y_i - R_{x,i}^T a\}^2 \\
&= (R_x a - y)^T K_x (R_x a - y) \\
&= a^T R_x^T K_x R_x a - a^T R_x^T K_x y - y^T K_x R_x a + y^T y \\
&= F_x
\end{aligned}$$

We take the derivative and set to zero:

$$\partial F_x / \partial a = (K_x + K_x^T)(R_x a - y) R_x$$