SDS 383D, Exercises 3: Linear smoothing and Gaussian processes

Tyler Buffington

March 5, 2019

Curve fitting by linear smoothing

(A) We begin with the linear regression equation:

$$y_i = \beta x_i + \epsilon_i$$

Recall that we derived the least-squares estimator for multiple regression:

$$\hat{\beta} = (X^T X)^{-1} X^T y$$

For the case with the means subtracted, this is equivalent to

$$\frac{\sum_{i=1}^n x_i y_i}{\sum_{i=1}^n x_i^2}$$

Back to our regression equation for one value, but for an new point, x^*

$$\hat{y}^* = \hat{\beta} x^*$$

Note that the error term is excluded because it has mean zero. Substituting in what we derived for $\hat{\beta}$:

$$\hat{y^*} = \frac{\sum_{i=1}^n x_i y_i}{\sum_{i=1}^n x_i^2} x^*$$

which can then be written like:

$$\hat{y}^* = \sum_{i=1}^n w(x_i, x^*) y_i$$

where

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

This smoothes the data by collapsing it to the least-squares regression line. The K-nearest-neighbor smoothing will conversely take the average of the K nearest points (by x). It is worth noting that in the linear smoother, data points with x values very far away from $\bar{x}$ are assigned large weights.

(B) The code for the analysis can be found in kernel_smooth.py. The resulting plot is shown below:

This plot illustrates the bias-variance tradeoff: as h increases, the fit becomes less noisy, but more biased. In the limit as h goes to inf, the fit becomes a flat line at the mean because this would mean that all points have equal weight. On the other hand, as h goes to zero, the fit would match the data exactly, which is also not useful because it would pick up the noise from the raw data as well.

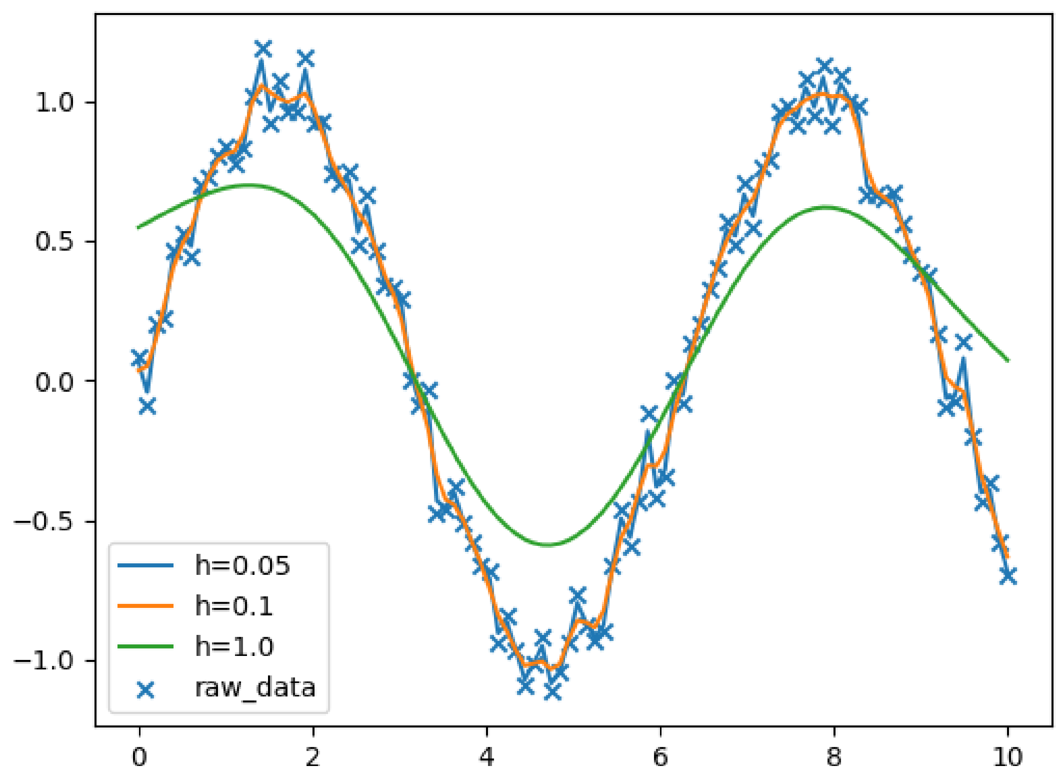


Figure 1: The fit for different bandwidth values.

Cross validation

(A) See cross_validaiton.py.

(B) The methodology is detailed in bandwidth_select.py. The smooth function was chosen to be $y = x^3$ and tthe wiggly function was chosen to be $y = sin(50x)$. The error as a function of bandwidth, h, is plotted below:

the main features are that the noisy functions have higher errors at the optimal h, and the noise also increases the value for the optimal h. the optimal fits are shown for the four cases below:

Local polynomial regression

(A) For points u in a neighborhood of the target point x, define the polynomial

$$g_x(u; a) = a_0 + \sum_{k=1}^{D} a_k (u - x)^k$$

where

$$\hat{a} = \arg\min_{R^{D+1}} \sum_{i=1}^{n} w_i \{y_i - g_x(x_i; a)\}^2$$

substitute in $g_x(x_i; a)$

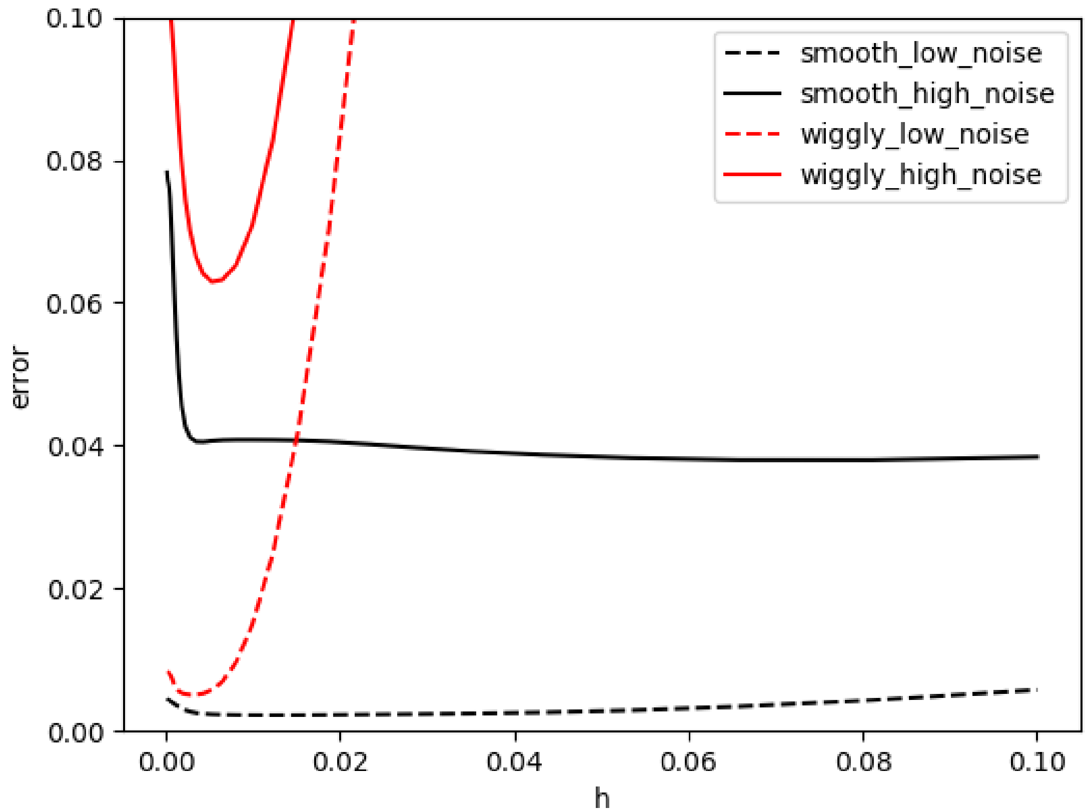


Figure 2: the error for different bandwidth values.

$$\hat{a} = \arg\min_{R^{D+1}} \sum_{i=1}^{n} w_i \{y_i - a_0 - \sum_{k=1}^{D} a_k (x_i - x)^k\}^2$$

Then define matrix R such that $R_{i,j} = (x_i - x)^{j-1}$, so then we get

$$\hat{a} = \arg\min_{R^{D+1}} \sum_{i=1}^{n} w_i \{y_i - Ra\}^2$$

If we define W to be a diagonal matrix where $W_{i,i} = w_i$, and all off diagonal terms are zero, we get

$$\hat{a} = \arg\min_{R^{D+1}} \{y - Ra\}^T W \{y - Ra\}$$

Then evaluate derivative and set equal to zero to minimize:

$$\frac{d}{da}\{y - Ra\}^T W \{y - Ra\}$$

$$= \frac{d}{da}[y^T W y - 2y^T W Ra + a^T R^T W Ra]$$

$$= -2R^T W y + 2R^T W Ra = 0$$

$$R^T W Ra = R^T W y$$

$$a = (R^T W R)^{-1} R^T W y$$

We can then define a matrix $M = (R^T W R)^{-1} W$ so that we get the final clean form:

$$a = My$$

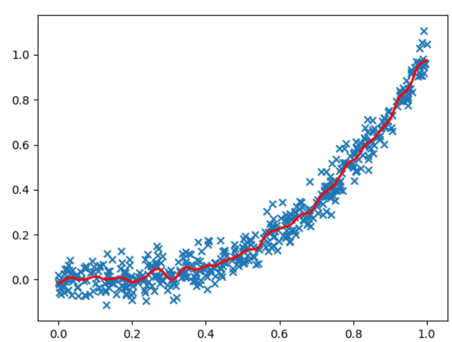
(B) Beginning with result from part A:

$$a = (R^T W R)^{-1} R^T W y$$

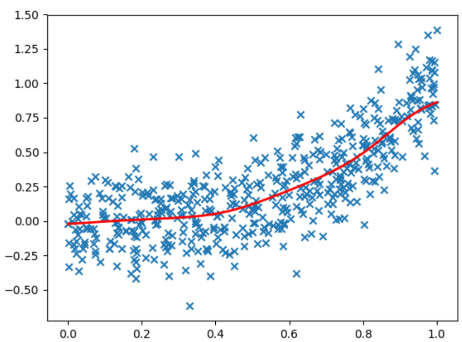
$$= \left(\begin{bmatrix} 1 & \dots & 1 \\ (x_1 - x) & \dots & (x_n - x) \end{bmatrix} \begin{bmatrix} w_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & w_n \end{bmatrix} \begin{bmatrix} 1 & (x_1 - x) \\ \vdots & \vdots \\ 1 & (x_n - x) \end{bmatrix} \right)^{-1} \begin{bmatrix} w_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & w_n \end{bmatrix} \begin{bmatrix} 1 & (x_1 - x) \\ \vdots & \vdots \\ 1 & (x_n - x) \end{bmatrix} \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$$

$$= \begin{bmatrix} \sum_{i=1}^n w_i & \sum_{i=1}^n w_i(xi - x) \\ \sum_{i=1}^n w_i(x_i - x) & \sum_{i=1}^n w_i(x_i - x)^2 \end{bmatrix}^{-1} \begin{bmatrix} \sum_{i=1}^n y_i w_i \\ \sum_{i=1}^n y_i w_i (x_i - x) \end{bmatrix}$$

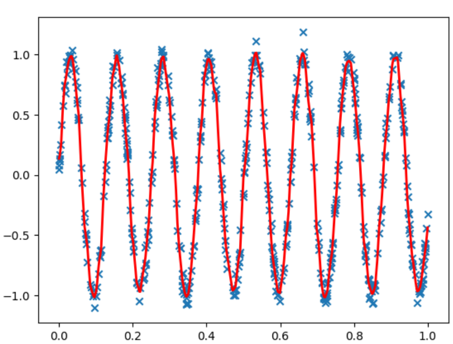
$$= \frac{1}{\sum_{i=1}^n w_i \sum_{i=1}^n w_i(x_i - x)^2 - (\sum_{i=1}^n w_i(x_i - x))^2} \begin{bmatrix} \sum_{i=1}^n w_i(x_i - x)^2 & -\sum_{i=1}^n w_i(x_i - x) \\ -\sum_{i=1}^n w_i(x_i - x) & \sum_{i=1}^n w_i \end{bmatrix} \begin{bmatrix} \sum_{i=1}^n y_i w_i \\ \sum_{i=1}^n y_i w_i (x_i - x) \end{bmatrix}$$



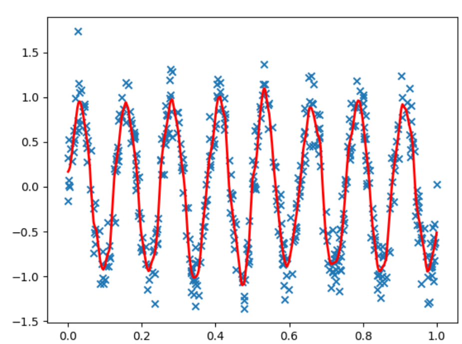
(a) smooth low noise



(b) smooth high noise



(c) wiggly low noise



(d) wiggly high noise

Figure 3: The sprinkler link temperatures and gas temperatures in the cell containing the sprinkler for the four cases presented in this abstract. The horizontal dashed lines represent the activation temperature of 70 ^{o}C.

$$= \frac{1}{\sum_{i=1}^n w_i \sum_{i=1}^n w_i(x_i - x)^2 - (\sum_{i=1}^n w_i(x_i - x))^2} \begin{bmatrix} \sum_{i=1}^n w_i(x_i - x)^2 \sum_{i=1}^n y_i w_i - \sum_{i=1}^n w_i(x_i - x) \sum_{i=1}^n y_i w_i(x_i - x) \\ - \sum_{i=1}^n w_i(x_i - x) \sum_{i=1}^n y_i w_i + \sum_{i=1}^n w_i \sum_{i=1}^n y_i w_i(x_i - x) \end{bmatrix}$$

It can be seen that at the target point x, $\hat{f} = a_0$, so we get

$$\hat{f} = \frac{\sum_{i=1}^n w_i(x_i - x)^2 \sum_{i=1}^n y_i w_i - \sum_{i=1}^n w_i(x_i - x) \sum_{i=1}^n y_i w_i(x_i - x)}{\sum_{i=1}^n w_i \sum_{i=1}^n w_i(x_i - x)^2 - (\sum_{i=1}^n w_i(x_i - x))^2}$$

Let $s_1 = \sum_{i=1}^n w_i(x_i - x)$ and $s_2 = \sum_{i=1}^n w_i(x_i - x)^2$. So then we get:

$$\hat{f} = \frac{s_2 \sum_{i=1}^n y_i w_i - s_1 \sum_{i=1}^n y_i w_i(x_i - x)}{\sum_{i=1}^n w_i s_2 - s_1 \sum_{i=1}^n w_i(x_i - x)}$$

$$= \frac{\sum_{i=1}^n y_i w_i \big(s_2 - s_1(x_i - x)\big)}{\sum_{i=1}^n w_i \big(s_2 - s_1(x_i - x)\big)}$$

Define $w_i^* = w_i\big(s_2 - s_1(x_i - x)\big)$, so then we get the desired form:

$$\hat{f} = \frac{\sum_{i=1}^n w_i^* y_i}{\sum_{i=1}^n w_i^*}$$