STATS205: Introduction to Nonparametric Statistics

Lecturer: Tengyu Ma
Scribe: Anna Shors

Lecture # 2
April 9th, 2021

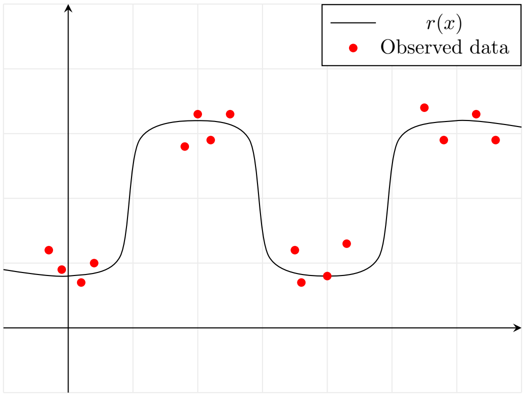
2.1 Review and overview

In the last lecture, we introduced the idea of nonparametric regression including the regressogram, local averaging, and Naradaya-Watson kernel estimators.

In this lecture, we expand on the ideas introduced in the first lecture. We discuss the bias-variance trade-off and its implications in the nonparametric setting, citing a formal theorem. We next introduce the concept of linear smoothers as a way to unify many common nonparametric regression methods and conclude with an introduction to another approach to nonparametric regression: local linear regression. We discuss how local linear regression overcomes some of the challenges faced by kernel estimators.

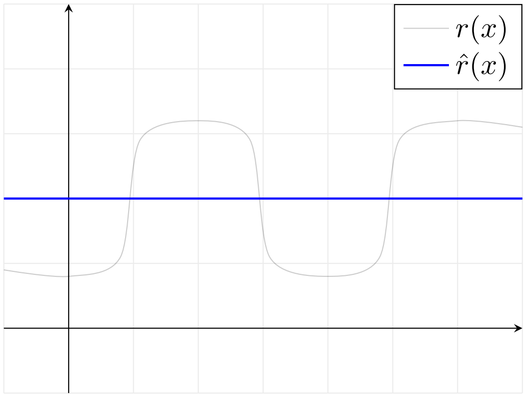
2.2 Example: the optimal bandwidth

Consider the function $r(x)$, along with the given data points, shown in the plot below:



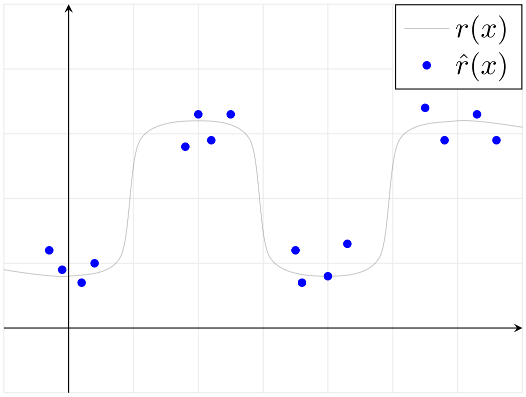
We would like to use local averaging to estimate $r(x_i)$ at each data point x_i. We consider three choices for the bandwidth h.

1. $h = \infty$: when h is infinite, we simply average over all data points in the dataset when making a prediction for a given point x. Thus, the resulting estimator $\hat{r}$ is simply a constant function, as shown in blue below:

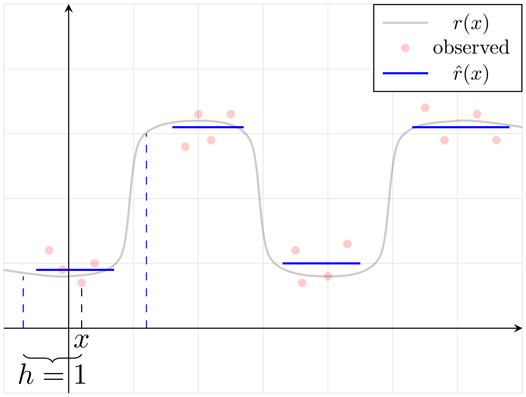


$h = \infty$ is clearly undesirable in this example, because $\hat{r}(x)$ carries no information about the underlying function $r(x)$.

2. $h = 0$: the only data point x_i that satisfies $|x - x_i| \leq h$ is x itself. Thus, for a given x_i in the training set, the predicted value of $r(x_i)$ will be precisely equal to the observed outcome Y_i (i.e. $\hat{r}(x_i) = Y_i$), so that we fail to denoise the dataset at all.



3. $h = 1$: This is the best choice for this example. $h = 1$ has the effect of averaging only the nearby points when making a prediction on an example x_i. This is desirable because points substantially far from x_i exhibit quite different behavior and thus should not influence the estimate of $r(x_i)$. $h = 1$ has the effect of adequately denoising the data without smoothing the estimator $\hat{r}$ too severely, as the plot below illustrates.



2.3 The bias-variance trade-off

2.3.1 Motivation for using MSE

We begin with a motivation for why we use MSE to measure performance of an estimator. We claim that MSE is very closely related to predictive risk, which measures how well $\hat{r}(x)$ performs on a new observation (with fresh randomness). After creating our estimator $\hat{r}(x)$, let $Z_i = r(x_i) + \xi_i$, where ξ_i has mean 0 and variance σ^2. Then,

$$\begin{aligned}
\text{predictive risk}(\hat{r}) &\triangleq \mathbb{E}_{Z_i}\left[\frac{1}{n}\sum_{i=1}^{n}(\hat{r}(x_i) - Z_i)^2\right] \\
&= \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}_{Z_i}\left[(\hat{r}(x_i) - Z_i)^2\right] \\
&= \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left[(\hat{r}(x_i) - r(x_i) - \xi_i)^2\right] \\
&= \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left[(\hat{r}(x_i) - r(x_i))^2 - 2(\hat{r}(x_i) - r(x_i))\xi_i + \xi_i^2\right] \\
&= \frac{1}{n}\sum_{i=1}^{n}\left((\hat{r}(x_i) - r(x_i))^2 - 2(\hat{r}(x_i) - r(x_i))\,\mathbb{E}[\xi_i] + \mathbb{E}[\xi_i^2]\right) \\
&= \frac{1}{n}\sum_{i=1}^{n}\left((\hat{r}(x_i) - r(x_i))^2 + \mathbb{E}[\xi_i^2]\right) \\
&= \frac{1}{n}\sum_{i=1}^{n}(\hat{r}(x_i) - r(x_i))^2 + \sigma^2 \\
&= \text{MSE}(\hat{r}) + \text{constant}. \qquad (2.1)
\end{aligned}$$

We thus have that MSE and predictive risk are identical up to a constant. Note that in this derivation, we have used the fact that the expectation is taken with respect to Z_i to pull the constants with respect to Z_i (in particular, $\hat{r}(x_i) - r(x_i)$) out of the expectation.

2.3.2 Bias-variance decomposition for MSE

Recall that our estimator $\hat{r}$ is random, because it is a function of the observed outcomes Y_i, which are assumed to be random due to the noise in the observations. In contrast, the true underlying function r is fixed. Thus,

$$\begin{aligned}\text{MSE} &:= \mathbb{E}_{Y_i}[\text{MSE}(\hat{r})] \\ &= \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}(\hat{r}(x_i) - r(x_i))^2\right] \\ &= \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}[(\hat{r}(x_i) - r(x_i))^2]. \qquad (2.2)\\ \mathbb{E}[(\hat{r}(x_i) - r(x_i))^2] &= (\mathbb{E}[\hat{r}(x_i) - r(x_i)])^2 + \text{Var}(\hat{r}(x_i) - r(x_i)) \\ &= (\mathbb{E}[\hat{r}(x_i) - r(x_i)])^2 + \text{Var}(\hat{r}(x_i)), \qquad (2.3)\end{aligned}$$

where we used the fact that for any random variable X, $\mathbb{E}[X^2] = \mathbb{E}[X]^2 + \text{Var}(X)$ and the fact that shifting a random variable by constant does not change the variance. Combining (2.2) and (2.3) and using the fact that $r(x)$ is fixed, we see that

$$\text{MSE} = \boxed{\frac{1}{n}\sum_{i=1}^{n}(\mathbb{E}[\hat{r}(x_i)] - r(x_i))^2} + \boxed{\frac{1}{n}\sum_{i=1}^{n}\text{Var}(\hat{r}(x_i))}, \qquad (2.4)$$

where the expression in the blue box is called the bias and the expression in the red box is the variance. Note that all results we have derived thus far hold for all estimators, parametric and non-parametric. We will next discuss results related to the bias-variance trade-off in the nonparametric setting.

2.3.3 Bias-variance trade-off in the nonparametric setting

Let us look at the bias and variance of kernel estimators. Note that the bias is closely related to $\mathbb{E}[\hat{r}(x)]$. Thus, we compute $\mathbb{E}[\hat{r}(x)]$ to investigate the bias.

$$\hat{r}(x) = \frac{\sum_{j=1}^{n} w_j Y_j}{\sum_{i=1}^{n} w_j} = \frac{1}{\sum_{j=1}^{n} w_j}\left(\sum_{j=1}^{n} w_j(r(x_j) + \xi_j)\right).$$

By linearity of expectation,

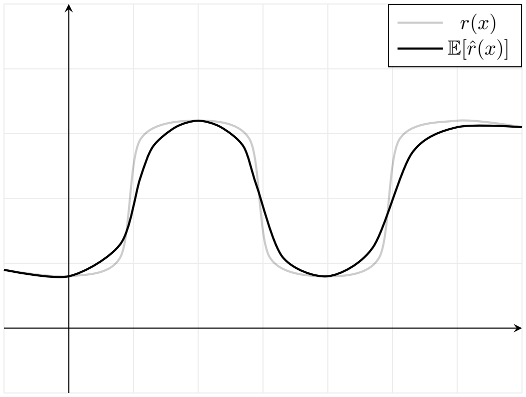
$$\mathbb{E}[\hat{r}(x)] = \frac{1}{\sum_{j=1}^{n} w_j}\left(\sum_{j=1}^{n} w_j(r(x_j) + \mathbb{E}[\xi_j])\right) = \frac{1}{\sum_{j=1}^{n} w_j}\left(\sum_{j=1}^{n} w_j(r(x_j))\right) = \frac{\sum_{j=1}^{n} w_j r(x_j)}{\sum_{j=1}^{n} w_j},$$

which gives us that

$$\text{Bias} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{\sum_{j=1}^{n} w_j r(x_j)}{\sum_{j=1}^{n} w_j} - r(x_i)\right)^2. \qquad (2.5)$$

where $w_j = K\left(\frac{|x_j - x|}{h}\right)$.

Thus, we see that $\mathbb{E}[\hat{r}(x)]$ is equivalent to applying the estimator to the "clean data" (i.e. data with no noise). $\mathbb{E}[\hat{r}(x)]$ is a "smoother" version of $r(x)$, as demonstrated in the plot below. The bias provides a measure of how much information is lost in the process of smoothing the initial function. Note that as bandwidth h increases, $\hat{r}(x)$ becomes smoother because the higher the value of h, the more weight we give to points further from x. Thus, as h increases, bias increases.



To compute the variance of $\hat{r}(x)$, assume the ξ_i are independent with mean 0 and variance σ^2. Then

$$\operatorname{Var}(\hat{r}(x)) = \operatorname{Var}\left(\frac{\sum_{i=1}^n w_j Y_j}{\sum_{i=1}^n w_j}\right) = \frac{1}{\left(\sum_{j=1}^n w_j\right)^2} \sum_{j=1}^n w_j^2 \operatorname{Var}(Y_j) = \left(\frac{\sum_{j=1}^n w_j^2}{\left(\sum_{j=1}^n w_j\right)^2}\right) \sigma^2. \tag{2.6}$$

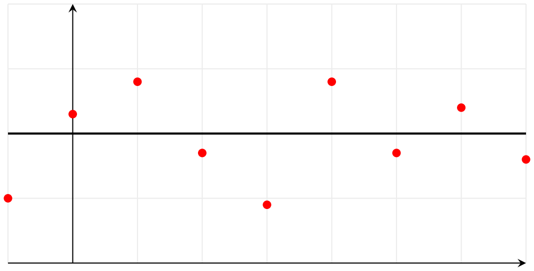
Let us compute the value of the variance in the case of local averaging. Recall that the kernel used for local averaging is the Boxcar kernel: $w_j = \mathbb{1}\{|x_j - x| \leq h\}$, so that $(w_1, \ldots, w_n) = (0, \ldots, 0, 1, \ldots, 1, 0, \ldots, 0)$, where the number of 1's is equal to the number of elements that are at most a distance of h from x.
Suppose we take $x = x_i$. Let $B_{x_i} = \{x_j \colon |x_j - x_i| \leq h\}$ and let $n_{x_i} = |B_{x_i}|$. Then

$$\operatorname{Var}(\hat{r}(x_i)) = \left(\frac{\sum_{j=1}^n w_j^2}{\left(\sum_{j=1}^n w_j\right)^2}\right) \sigma^2 = \left(\frac{\sum_{x_j \in B_{x_i}} 1^2}{\left(\sum_{x_j \in B_{x_i}} 1\right)^2}\right) \sigma^2 = \left(\frac{n_{x_i}}{n_{x_i}^2}\right) \sigma^2 = \frac{\sigma^2}{n_{x_i}}. \tag{2.7}$$

We thus see that, in the case of local averaging, the variance of $\hat{r}(x_i)$ decreases as a function of the number of points in the neighborhood of x_i. As $h \to \infty$, the number of points being averaged over increases. So $h = \infty$ gives the smallest possible variance of $\frac{\sigma^2}{n}$ and $h = 0$ gives the largest variance of σ^2.

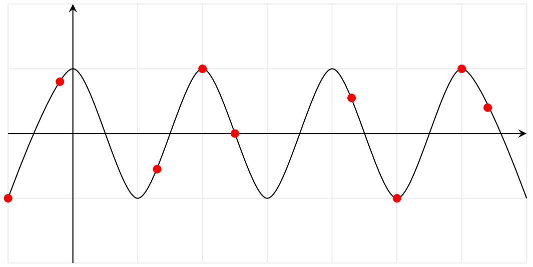
2.3.4 Case studies

Consider the constant function $r(x) = c$ shown below.



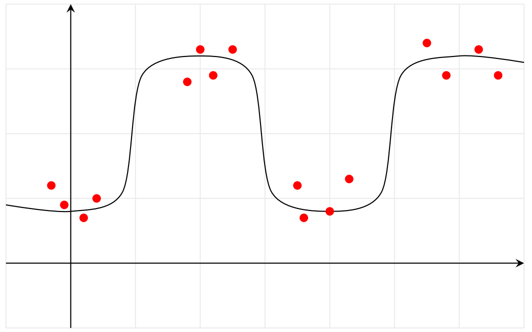
Because $r(x)$ is a constant function, bias will be equal to 0 for any choice of h ($\mathbb{E}[\hat{r}(x)] = \frac{\sum_{i=1}^{n} w_i r(x_i)}{\sum_{i=1}^{n} w_i} = \frac{\sum_{i=1}^{n} w_i c}{\sum_{i=1}^{n} w_i} = c$, so that $\mathbb{E}[\hat{r}(x)] = c = r(x)$ for any choice of kernel and h). However, the variance is sensitive to the bandwidth. In this case, we should pick $h = \infty$ to minimize variance and thus minimize MSE.

Next, let $r(x)$, and the observed data points, be as shown below. Note that $r(x)$ fluctuates quite substantially but the observed points have no noise ($\sigma = 0$).



In this case, variance $= \frac{\sum_{j=1}^{n} w_j^2}{\left(\sum_{j=1}^{n} w_j\right)^2}\sigma^2 = 0$ for all h. However, because the function fluctuates, the choice of bandwidth has a large effect on the bias, as it effects how "smooth" our estimates will be. Thus, in this case, the bandwidth should be chosen to minimize bias, i.e. $h = 0$.

Finally, consider the function from the beginning of this lecture, shown again below.



As discussed in section 2, it is desirable to pick a value of h between 0 and ∞, because we would like to average over the noise in the dataset (decrease variance) without smoothing our estimate too substantially (which would increase bias).

2.3.5 Effect of dataset size on bias-variance trade-off

In general, when we have more data, we can assume that there will be more observations within a given neighborhood of x. Because variance depends on the number of data points being averaged over, this means that increasing the number of observations will decrease variance. However, note that $\text{Bias}(\hat{r}(x)) = \frac{1}{n}\sum_{i=1}^{m}(\mathbb{E}[\hat{r}(x_i)] - r(x))^2$ depends only on the estimator's performance on the clean data, not on the noise in the dataset. Thus, increasing the number of observations does not change the bias.

Suppose you were at a "sweet spot" with the bias-variance trade-off. Then, you observe more data points. This has the effect of decreasing variance while keeping bias constant. Thus, to "rebalance" the bias and variance, you should decrease h slightly as to decrease the bias and increase the variance.

2.3.6 Formal theorem (bias-variance characterization)

We now make the previous discussions of the bias-variance trade-off in the context of kernel estimators more precise with a formal theorem. While the MSE of an estimator does not depend on the noise in the dataset, it does depend on the (arbitrary) positions of $x_1, \ldots, x_n$. For theoretical feasibility, we assume $x_1, \ldots, x_n \overset{\text{iid}}{\sim} P$, where P has density $f(x)$. Additionally, the theorem has the following setup:

- Assume we have an estimator $\hat{r}_n$ with n samples and that n tends to ∞.
- For a fixed n, the bandwidth is h_n.
- We use integrated risk to measure the quality of an estimator: $R(\hat{r}_n, r) \triangleq \int (\hat{r}_n(x) - r(x))^2 dx$

Theorem 2.1. *The risk of a Nadaraya-Watson kernel estimator is*

$$\begin{aligned} R(\hat{r}_n, r) &= \boxed{\frac{h_n^4}{4}\left(\int x^2 K(x)dx\right)^2 \int \left(r''(x) + 2r'(x)\frac{f'(x)}{f(x)}\right)^2 dx} \tag{2.8}\\ &+ \boxed{\sigma^2 \frac{\int K(x)^2 dx}{nh_n}\left(\int \frac{1}{f(x)}dx\right)} \tag{2.9}\\ &+ \boxed{o\left((nh_n)^{-1}\right) + o\left(h_n^4\right)}. \tag{2.10} \end{aligned}$$

The term in blue is the bias of the estimator and the term in red is the variance. The terms in green are higher order terms that go to 0 as $nh_n \to \infty$ and $h_n \to 0$.

We will next try to decompose this theorem to understand each of the individual parts.

Bias The theorem tells us that the bias depends on the following quantities:

- bandwidth h_n: The smaller the bandwidth, the lower the bias.
- $\int x^2 K(x)dx$: This quantity is a measure of how flat the kernel is. The flatter the kernel, the larger the value of $\int x^2 K(x)dx$ and thus the higher the bias. This aligns with our intuition, as the flatter the kernel, the more we are weighting further away points and thus the more smooth our estimator.
- $r''(x)$: This is a measure of the curvature of $r(x)$. The smoother the function, the lower the value of $r''(x)$ and thus the lower the bias.

- $r'(x)\frac{f'(x)}{f(x)}$: Note that this quantity is equal to $r'(x)(\log f(x))'$. This term is called the **design bias** because it depends on the "design" (i.e., the distribution of the x_i's). The design bias is small when $(\log f(x))'$ is small (i.e., when the density of X doesn't change too quickly, or X is "close to uniform").

Variance According to the theorem, the variance depends on the following quantities:

- σ^2: The larger the value of σ^2 (i.e. the variance of the random noise), the higher the variance.
- h_n: The higher the bandwidth, the lower the variance.
- n: The larger the value of n, the smaller the smaller the variance.

Implications on bandwidth h_n Let us treat K, f, and r as constants. We are interested in seeing how the optimal bandwidth h_n^* changes as a function of σ^2 and n. Holding K, f, and r constant, we can express the risk as

$$R(\hat{r}(x), r(x)) = h_n^4 c_1 + \frac{\sigma^2}{nh_n} c_2 + \text{higher order terms}, \tag{2.11}$$

where c_1 and c_2 are constants. We are thus interested in the choice of h_n that minimizes this risk. By taking the derivative of the risk and setting it equal to zero, we see that

$$h_n^* = c_3 \left(\frac{\sigma^2}{n}\right)^{1/5}. \tag{2.12}$$

This tells us that the optimal bandwidth decreases at a rate proportional to $n^{-1/5}$.

Plugging in h_n^* to the risk equation, we see that

$$\min_{h_n} \left(h_n^4 c_1 + \frac{\sigma^2 c_2}{nh_n}\right) = c_4 \left(\frac{\sigma^2}{n}\right)^{4/5}, \tag{2.13}$$

so that the lowest risk is on the order of $n^{-4/5}$. Note that the risk for most parametric models is on the order of n^{-1}, a slight improvement over the risk for the nonparametric models we have discussed.

2.4 Linear smoothers: a unified view

We now take a brief detour to introduce the concept of linear smoothers, which serve as a way to unify many of the aforementioned nonparametric regression methods.

Definition 2.2. $\hat{r}$ is a **linear smoother** if there exists a vector valued function $x \longrightarrow (l_1(x), \ldots, l_n(x))$ such that

$$\hat{r}(x) = \sum_{i=1}^{m} l_i(x) Y_i. \tag{2.14}$$

Note that the l_i's can depend on $x_1, \ldots, x_n$ but not on $Y_1, \ldots, Y_n$. Additionally, we must have that $\sum_{i=1}^{n} l_i(x) = 1$.

Proposition 2.3. The regressogram and kernel estimator are both instances of linear smoothers.

- Regressogram:

$$\hat{r}(x) = \frac{1}{|B_x|}\sum_{i \in B_x} Y_i = \sum_{i=1}^{n}\left(\frac{\mathbb{1}\{x_i \in B_x\}}{|B_x|}\right) Y_i = \sum_{i=1}^{m} l_i(x) Y_i, \tag{2.15}$$

  where B_x is the bin containing x and $l_i(x) = \frac{\mathbb{1}\{x_i \in B_x\}}{|B_x|}$.

- Kernel estimator:

$$\hat{r}(x) = \frac{\sum_{i=1}^{n} w_i Y_i}{\sum_{i=1}^{n} w_i} = \sum_{i=1}^{n}\left(\frac{w_i}{\sum_{j=1}^{n} w_j}\right) Y_i = \sum_{i=1}^{m} l_i(x) Y_i, \tag{2.16}$$

  where $l_i(x) = \frac{w_i}{\sum_{j=1}^{n} w_j}$

Thus, linear smoothers provide a "unified view" in that they provide a category into which many different types of estimators fall. In the next section, we will introduce the method of local linear regression, which we show is yet another instance of a linear smoother. We now conclude this section with a few facts about linear smoothers.

- We can write any linear smoother in the matrix multiplication form $\hat{\boldsymbol{r}} = \boldsymbol{LY}$, where $\hat{\boldsymbol{r}} = \begin{bmatrix} \hat{r}_1(x) \\ \vdots \\ \hat{r}_n(x) \end{bmatrix}$, $\boldsymbol{L} = \begin{bmatrix} l_1(x_1) & \cdots & l_n(x_1) \\ \vdots & \ddots & \vdots \\ l_1(x_n) & \cdots & l_n(x_n), \end{bmatrix}$, and $\boldsymbol{Y} = \begin{bmatrix} Y_1 \\ \vdots \\ Y_n \end{bmatrix}$.

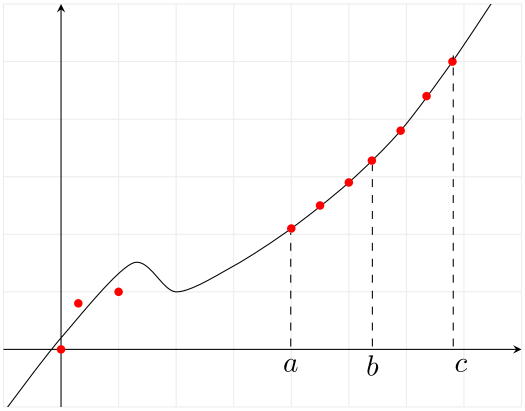
- For all linear smoothers,

$$\mathbb{E}[\hat{r}(x)] = \mathbb{E}\left[\sum_{i=1}^{n} l_i(x) Y_i\right] = \sum_{i=1}^{n} l_i(x)\,\mathbb{E}[Y_i] = \sum_{i=1}^{n} l_i(x) r(x), \tag{2.17}$$

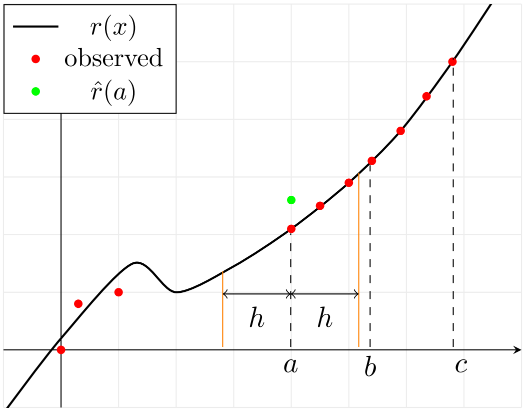
  so $\mathbb{E}[\hat{r}(x)]$ is equal to the estimate when the estimator is run on clean data. Like in the particular case of kernel estimators, the bias is an indicator of how much we damage the clean data by smoothing.

2.5 Local linear regression

Local linear regression provides an alternative to kernel estimators. To motivate why we might want such an alternative, let us revisit the idea of design bias. Consider the following function $r(x)$, along with the observed data points, shown below.



If we were to use a kernel estimator to attempt to predict $r(a)$, we would overestimate the true value. This is because all observed points that are close to a are to the right of a, so we would average primarily over points whose y-values are larger than a's. The predicted value for $r(a)$ using local averaging is marked in green in the plot below.



Similarly, using a kernel estimator to predict $r(c)$ would result in an underestimate. In contrast, because b has nearby points on either side, the prediction for $r(b)$ would be reasonable. This problem of over/underestimating $r(x)$ for points that have no observations on one side (i.e. points on the boundary of a given bin) is closely related to the design bias. It occurs because, when using kernel estimators, we make the assumption that $r(x)$ is locally constant.

Local linear regression provides a solution to this problem. Rather than assuming $r(x)$ is locally constant, we assume that $r(x)$ is locally linear. For a given data point x, we would like to approximate $r(x)$ by locally fitting a line based at x to our data. Let $\tilde{r}_x(u) = a_1(u - x) + a_0$. Then the algorithm for local linear regression is as follows.

For a given data point x, let

$$\hat{a}_0, \hat{a}_1 = \underset{a_0,a_1}{\operatorname{argmin}} \sum_{i=1}^{n} w_i(Y_i - \tilde{r}_x(x_i))^2 = \underset{a_0,a_1}{\operatorname{argmin}} \sum_{i=1}^{n} w_i(Y_i - (a_1(x_i - x) + a_0))^2, \tag{2.18}$$

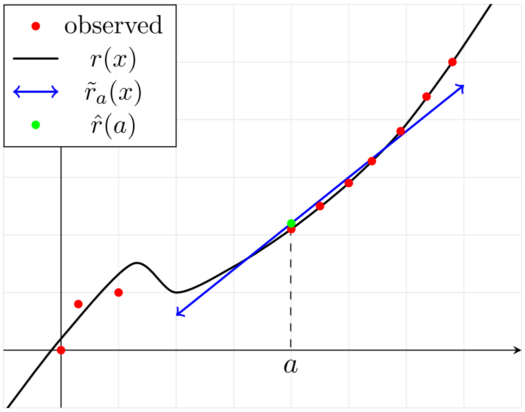
where $w_i = K\left(\frac{(x_i - x)}{h}\right)$ for some kernel function K. Then, we let our estimate $\hat{r}(x)$ equal the intercept term:

$$\hat{r}(x) \overset{\Delta}{=} \hat{a}_0. \tag{2.19}$$

Theorem 2.4. *The integrated risk of local linear regression consists of a bias term and variance term. The variance is the same as that from Theorem 2.1. The bias is*

$$\frac{h_n^4}{4}\left(\int x^2K(x)dx\right)^2\int r''(x)^2dx. \tag{2.20}$$

If we compare this bias to the bias from the kernel estimators ($\frac{h_n^4}{4}(\int x^2K(x)dx)^2\cdot \int(r''(x)+2r'(x)\frac{f'(x)}{f(x)})^2dx$), we see that the bias for local linear regression does not have the $2r'(x)\frac{f'(x)}{f(x)}$ term (i.e. no design bias!). Local linear regression thus mitigates the problem of design bias that we encounter when using kernel estimators. The diagram below provides some intuition as to why this is the case.



We see that the local linear assumption allows us to better approximate $r(x)$ for values of x that lie on the boundary. We conclude by proving that local linear regression is another instance of a linear smoother.

$$\hat{a}_0, \hat{a}_1 = \underset{a_0,a_1}{\text{argmin}}\sum_{i=1}^n w_i(Y_i-(a_1(x_j-x)+a_0))^2 = \underset{a_0,a_1}{\text{argmin}}\ \ g(a_0,a_1).$$

$$\frac{\partial g}{\partial a_0} = 2\sum_{i=1}^n w_i(a_1(x_i-x)+a_0-Y_i)=0$$

$$\frac{\partial g}{\partial a_1} = 2\sum_{i=1}^n w_i(a_1(x_i-x)+a_0-Y_i)(x_i-x)=0.$$

By solving this system of equations for a_0 and a_1, we get that

$$\hat{r}(x)=\hat{a}_0=\frac{\sum_{i=1}^n(w_iCY_i-w_i(x_i-x)BY_i)}{AC-B^2}=\sum_{i=1}^n\left(\frac{w_iC-w_i(x_i-x)B}{AC-B^2}\right)Y_i, \tag{2.21}$$

where $A=\sum_{i=1}^n w_i$, $B=\sum_{i=1}^n w_i(x_i-x)$ and $C=\sum_{i=1}^n w_i(x_i-x)^2$. We thus see that $\hat{r}(x)$ is a linear combination of the Y_i's, so local linear averaging is indeed a linear smoother.