Chapter 6

Local averaging methods

Chapter summary

- First chapter on non-parametric methods that are not based on parametric models and can adapt to complex target functions.
- "Linear" estimators: These are based on assigning weight functions to each observation so that each observation can vote for its label with the corresponding weight (typically non-linear in the input variables).
- Partitioning estimates: the input space is cut into non-overlapping cells, and the predictor is piecewise-constant.
- Nadaraya-Watson estimators (a.k.a. kernel regression): each observation assigns a weight proportional to its distance in input space.
- k-nearest-neighbors: each observation assigns an equal weight to its k nearest neighbors.
- Consistency: All of these methods can provably learn complex Lipschitz-continuous non-linear functions with a convergence rate of the form $O(n^{-2/(d+2)})$, where d is the underlying input dimension, leading to the curse of dimensionality.

6.1 Introduction

Like in previous chapters, we consider the supervised learning set-up, where we are being given a training set: observations $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$, $i = 1, \dots, n$, of inputs/outputs, features/variables are assumed independent and identically distributed (i.i.d.) random variables with common distribution p. We consider a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, where $\ell(y, z)$ is the loss of predicting z while the true label is y.

Our goal is to minimize the expected risk, that is, the generalization performance of a prediction function f from $\mathcal{X}$ to $\mathcal{Y}$:

$$\mathcal{R}(f) = \mathbb{E}\big[\ell(y, f(x))\big],$$

where the expectation is taken with respect to the distribution p.

⚠ Like in the rest of the book, we assume that the testing distribution is the same as the training distribution.

⚠ Be careful with the randomness or lack thereof of f: The estimator $\hat{f}$ we will use depends on the training data and not on the testing data, and thus $\mathcal{R}(\hat{f})$ is random because of the dependence on the training data.

As seen in Chapter 2, the two classical cases are:

- Binary classification: $\mathcal{Y} = \{0, 1\}$ (or often $\mathcal{Y} = \{-1, 1\}$), and $\ell(y, z) = 1_{y \neq z}$ ("0-1" loss). Then $\mathcal{R}(f) = \mathbb{P}(f(x) \neq y)$.
- Regression: $\mathcal{Y} = \mathbb{R}$ and $\ell(y, z) = (y - z)^2$ (square loss). Then $\mathcal{R}(f) = \mathbb{E}(y - f(x))^2$.

As seen in Chapter 2, minimizing the expected risk leads to an optimal "target function," called the Bayes predictor $f^* \in \arg\min \mathcal{R}(f) = \mathbb{E}\big[\ell(y, f(x))\big]$. As shown in Section 2.2.3, the optimal predictor can be obtained from the conditional distribution of $y|x$ as

$$f^*(x) \in \arg\min_{z \in \mathcal{Y}} \ \mathbb{E}(\ell(y, z)|x).$$

Note that (a) the Bayes predictor is not unique but that all Bayes predictors lead to the same Bayes risk, and (b) the Bayes risk is usually non-zero (unless the dependence between x and y is deterministic). The goal of supervised machine learning is thus to estimate f^*, knowing only the training data $\mathcal{D} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ *and* the loss ℓ, with the goal of minimizing the risk or the excess risk $\mathcal{R}(f) - \mathcal{R}^*$. We have the following special cases:

- For binary classification: $\mathcal{Y} = \{0, 1\}$ and $\ell(y, z) = 1_{y \neq z}$, the Bayes predictor is equal to $f^*(x) \in \arg\max_{i \in \{0,1\}} \mathbb{P}(y = i|x)$. This extends naturally to multi-category classification with the Bayes predictor $f^*(x) \in \arg \max\limits_{i \in \{1,\ldots,k\}} \mathbb{P}(y = i|x)$.

  If a convex surrogate from Section 4.1.1 is used, such as the logistic loss $\ell(y, z) = \log(1 + \exp(-yz))$ for $z \in \mathbb{R}$, then the target function is $f^*(x) = \log \frac{\mathbb{P}(y=1|x)}{\mathbb{P}(y=-1|x)}$.
- For regression: $\mathcal{Y} = \mathbb{R}$ and $\ell(y, z) = (y - z)^2$, the Bayes predictor is $f^*(x) = \mathbb{E}(y|x)$. Moreover, we have $\mathcal{R}(f) - \mathcal{R}^* = \int_{\mathcal{X}} (f(x) - f^*(x))^2 dp(x) = \|f - f^*\|^2_{L_2(dp(x))}$.

In Chapter 3 and Chapter 4, we explored methods based on empirical risk minimization, with explicit finite-dimensional models (often linear in their parameters) that may not be flexible enough to adapt to complex target functions. We now explore methods that can, starting with local averaging methods, which are not based on empirical risk minimization. In subsequent chapters, we will study kernel methods (Chapter 7) and neural networks (Chapter 9).

6.2 Local averaging methods

In local averaging methods, we aim at approximating the target function f^* directly *without any form of optimization*. This will be done by approximating the conditional distribution $p(y|x)$ of y given x, by some $\widehat{p}(y|x)$. We then replace the target function $f^*(x) \in \arg\min_{z\in\mathcal{Y}} \int_{\mathcal{Y}} \ell(y,z)dp(y|x)$ by $\hat{f}(x) \in \arg\min_{z\in\mathcal{Y}} \int_{\mathcal{Y}} \ell(y,z)d\widehat{p}(y|x)$. These are often called "plug-in" estimators.

In the usual cases, this leads to the following predictions:

- For classification with the 0-1 loss: $\hat{f}(x) \in \arg\max_{j\in\{1,\dots,k\}} \widehat{\mathbb{P}}(y=j|x)$.
- For regression with the square loss: $\hat{f}(x) = \int_{\mathcal{Y}} y \ d\widehat{p}(y|x)$.

6.2.1 Linear estimators

In this chapter, we will consider "linear" estimators, where the conditional distribution is of the form

$$\widehat{p}(y|x) = \sum_{i=1}^{n} \hat{w}_i(x)\delta_{y_i}(y),$$

where δ_{y_i} is the Dirac probability distribution at y_i (putting a unit mass at y_i), and the weight functions $\hat{w}_i : \mathcal{X} \to \mathbb{R}$, $i = 1,\dots,n$, depends on the input data only (for simplicity) and satisfy (almost surely in x):

$$\forall x \in \mathcal{X}, \quad \forall i \in \{1,\dots,n\},\ \hat{w}_i(x) \geqslant 0, \text{ and } \sum_{i=1}^{n} \hat{w}_i(x) = 1.$$

These conditions ensure that for all $x \in \mathcal{X}$, $\widehat{p}(y|x)$ is a probability distribution.

⚠ Some references allow for the weights not to sum to 1.

For our running examples, this leads to the following predictions:

- For classification: $\hat{f}(x) \in \arg\max_{j\in\{1,\dots,k\}} \sum_{i=1}^{n} \hat{w}_i(x) 1_{y_i=j}$, that is, each observation (x_i, y_i) votes for its label with weight $\hat{w}_i(x)$, a strategy often called "majority vote".
- For regression: $\mathcal{Y} = \mathbb{R}$: $\hat{f}(x) = \sum_{i=1}^{n} \hat{w}_i(x) y_i$. This is why the terminology "linear estimators" is sometimes used, since, as a function of the response vector in $\mathbb{R}^n$, the estimator is linear (note that this is the case as well for kernel ridge regression in Chapter 7). If we only consider predictions $\hat{f}(x_i)$ at the observed inputs, the vector $\hat{y} \in \mathbb{R}^n$ of predictions $\hat{y}_i = \hat{f}(x_i)$, for $i \in \{1,\dots,n\}$ is of the form $\hat{y} = Hy$, where the matrix $H \in \mathbb{R}^{n\times n}$, often called the smoothing matrix or the "hat matrix", is such that $H_{ij} = \hat{w}_j(x_i)$.

  Note that on top of being a linear estimator, the estimator satisfies additional properties: if the same constant is added to all outputs, the exact same constant

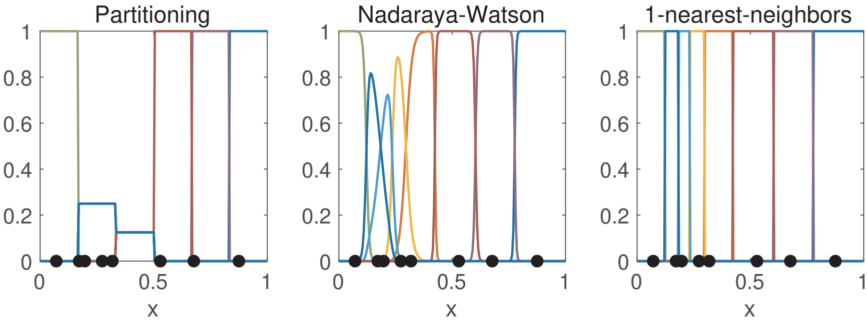


Figure 6.1: Weights of linear estimators in $d = 1$ dimension for the three types of local averaging estimators. The $n = 8$ weight functions $x \mapsto \hat{w}_i(x)$ are plotted with the observations in black.

is added to the prediction function; moreover, given two vectors of outputs y and $y' \in \mathbb{R}^n$, with two prediction functions $\hat{f}$ and $\hat{f}'$, if $y_i \leqslant y_i'$ for all $i \in \{1, \dots, n\}$, then $\hat{f}(x) \leqslant \hat{f}'(x)$ for all $x \in \mathcal{X}$.

Construction of weight functions. In most cases, for any i, the weight function $\hat{w}_i(x)$ is close to 1 for training points x_i which are close to x. We now show three classical ways of building them: (1) partition estimators, (2) Nearest-neighbors, and (3) Nadaraya-Watson estimator (a.k.a. kernel regression). See examples in Figure 6.1.

6.2.2 Partition estimators

If $\mathcal{X} = \bigcup_{j \in J} A_j$ is a partition (such that for all distinct $j, j' \in J$, $A_j \cap A_{j'} = \varnothing$) of $\mathcal{X}$ with a countable index set J (which we will assume finite for simplicity, equal to $\{1, \dots, |J|\}$), then we can consider for any $x \in \mathcal{X}$ the corresponding element $A(x)$ of the partition (that is, $A(x)$ is the unique A_j, $j \in J$, such that $x \in A_j$), and define

$$\hat{w}_i(x) = \frac{1_{x_i \in A(x)}}{\sum_{i'=1}^n 1_{x_{i'} \in A(x)}}, \tag{6.1}$$

with the convention that if no training data point lies in $A(x)$, then $\hat{w}_i(x)$ is equal to $1/n$ for each $i \in \{1, \dots, n\}$. This implies that each $\hat{w}_i$ is piecewise constant with respect to the partition, that is, for any non-empty cell A_j (that is, for which at least one observation falls in A_j), for any $x \in A_j$, the vectors $(\hat{w}_i(x))_{i \in \{1,\dots,n\}}$ has weights equal to $1/n_{A_j}$ for $i \in A_j$, where n_{A_j} is the number of training points in the set A_j, and 0 otherwise.

Equivalence with least-squares regression. When applied to regression where the estimator is $\hat{f}(x) = \sum_{i=1}^n \hat{w}_i(x) y_i$, then using a partition estimator can be seen as a least-squares estimator with feature vector $\binom{\varphi(x)}{1} = \binom{(1_{x \in A_j})_{j \in J}}{1} \in \mathbb{R}^{|J|+1}$, as we now show.

Indeed, we then aim to estimate $\binom{\theta}{\eta} \in \mathbb{R}^{|J|+1}$ for the prediction function

$$\hat{f}(x) = \sum_{j \in J} \theta_j 1_{x \in A_j} + \eta.$$

From training data $(x_1, y_1), \dots, (x_n, y_n)$, as shown in Chapter 3, we can directly estimate the constant term as $\eta = \bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$, while for the other components, we need to solve the normal equations

$$\sum_{i=1}^n \varphi(x_i)\varphi(x_i)^\top \theta = \sum_{i=1}^n (y_i - \bar{y})\varphi(x_i).$$

It turns out that the matrix $n\widehat{\Sigma} = \sum_{i=1}^n \varphi(x_i)\varphi(x_i)^\top$ is diagonal where for each $j \in J$, $n\widehat{\Sigma}_{jj}$ is equal to n_{A_j} the number of data points lying in cell A_j. This implies that for a non-empty cell A_j, θ_j is the average of all $y_i - \bar{y}$, for all i such that x_i lies in A_j. Thus, for all $x \in A_j$, the prediction is exactly $\theta_j + \bar{y}$, as obtained from weights in Eq. (6.1). For empty cells, θ_j is not determined by the normal equations above, and if we set it to zero, we recover our convention of predicting as the mean of all labels.

⚠ Other conventions exist (such as all zero weights when no data point lies in $A(x)$).

This equivalence with least-squares estimation with a diagonal (empirical or not) non-centered covariance matrix makes it attractive for theoretical purposes, as the inversion of the population and expected covariance matrices could be done in closed form.

Choice of partitions. There are two standard applications of partition estimators:

- **Fixed partitions**: for example, when $\mathcal{X} = [0, 1]^d$, we can consider cubes of length h, with $|J| = h^{-d}$ (see example below in $d = 2$ dimension with $|J| = 25$). Note here that the computation time for each $x \in \mathcal{X}$ is not necessarily proportional to $|J|$ but to n (by simply considering the bins where the data lie). This estimator is sometimes called a "regressogram". We need then to choose the bandwidth h (see analysis in Section 6.3.1). See Figure 6.2 for an illustration in one dimension.

A_1	A_2	A_3	A_4	A_5
A_6	A_7	A_8	A_9	A_{10}
A_{11}	A_{12}	A_{13}	A_{14}	A_{15}
A_{16}	A_{17}	A_{18}	A_{19}	A_{20}
A_{21}	A_{22}	A_{23}	A_{24}	A_{25}

- **Decision trees**: for data in a hypercube, we can recursively partition it by selecting a variable to split, leading to a maximum reduction in errors when defining the partitioning estimate.[1] A model selection criterion is then needed to control

[1] See more details in https://en.wikipedia.org/wiki/Decision_tree_learning.

the number of cells in the partition (see, e.g., Friedman et al., 2009, Section 9.2). Note here that the partition depends on the labels (so the analysis below does not apply unless the partitioning is learned on data different from the one used for the estimation).

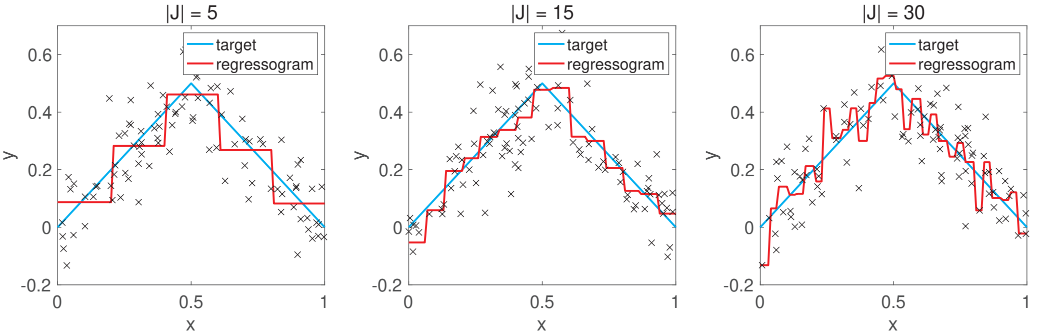


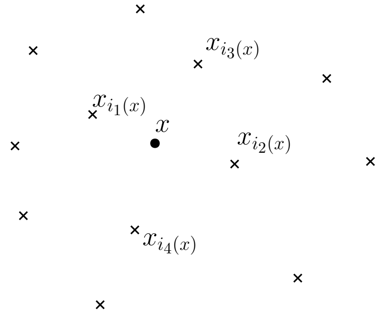
Figure 6.2: Regressograms in $d = 1$ dimension, with three values of $|J|$ (the number of sets in the partition). The $n = 100$ input data points are distributed uniformly on $[0, 1]$, and, for $i \in \{1, \dots, n\}$, the outputs y_i are equal to $\frac{1}{2} - |x_i - \frac{1}{2}| + \varepsilon_i$, where ε_i is a Gaussian with mean zero and variance $\sigma^2 = \frac{1}{100}$. We can observe both underfitting ($|J|$ too small) and overfitting ($|J|$ too large). Note that the target function f^* is piecewise affine and that on the affine parts, the estimator is far from linear; that is, the estimator cannot take advantage of extra-regularity (see Section 6.5 for more details).

6.2.3 Nearest-neighbors

Given an integer $k \geqslant 1$, and a distance d on $\mathcal{X}$, for any $x \in \mathcal{X}$, we can order the n observations so that

$$d(x_{i_1(x)}, x) \leqslant d(x_{i_2(x)}, x) \leqslant \cdots \leqslant d(x_{i_n(x)}, x),$$

where $\{i_1(x), \dots, i_n(x)\} = \{1, \dots, n\}$, and ties are broken randomly[2] (that is, for all $x \in \mathcal{X}$, by sampling randomly once, which indices should come first for each i). See the illustration below.

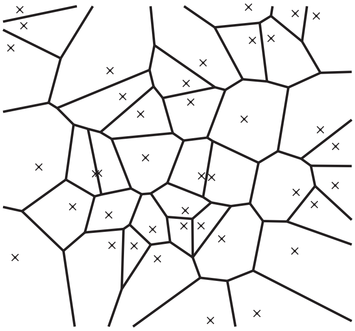


[2] Other conventions share the weights among all ties.

We then define

$$\hat{w}_i(x) = 1/k \ \text{ if } \ i \in \{i_1(x), \dots, i_k(x)\}, \ \text{and 0 otherwise.}$$

Given a new input $x \in \mathbb{R}^d$, the nearest neighbor predictor looks at the k nearest points x_i in the data set $\{(x_1, y_1), \dots, (x_n, y_n)\}$ and predicts a majority vote among them (for classification) or simply the averaged response (for regression). The number of nearest neighbors is the hyperparameter, which needs to be estimated (typically by cross-validation); see Section 6.3.2 for an analysis. See a one-dimensional example in Figure 6.3. For $k = 1$, the prediction function is piecewise constant, with each constant piece corresponding to a region where a given observation is the nearest neighbor, leading, in two dimensions to the Voronoi diagram below, with all regions displayed.



Algorithms. Given a test point $x \in \mathcal{X}$, the naive algorithm looks at all training data points for computing the predicted response. Thus the complexity is $O(nd)$ per test point in $\mathbb{R}^d$. When n is large, this is costly in time and memory. Indexing techniques exist for (potentially approximate) nearest-neighbor search, such as “k-d-trees”,[3] with typically a logarithmic complexity in n (but with some additional compiling time), with a memory footprint that can grow exponentially in dimension (see, e.g., Shakhnarovich et al., 2005).

Exercise 6.1 *What is the pattern of non-zeros of the smoothing matrix $H \in \mathbb{R}^{n \times n}$?*

6.2.4 Nadaraya-Watson estimator a.k.a. kernel regression (♦)

Given a “kernel” function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$, which is pointwise non-negative, we define

$$\hat{w}_i(x) = \frac{k(x, x_i)}{\sum_{i'=1}^n k(x, x_{i'})},$$

with the convention that if $k(x, x_i) = 0$ for all $i \in \{1, \dots, n\}$, then $\hat{w}_i(x)$ is equal to $1/n$ for each i (which is the same convention used for estimators based on partitions in Section 6.2.2).

[3] See https://en.wikipedia.org/wiki/K-d_tree.

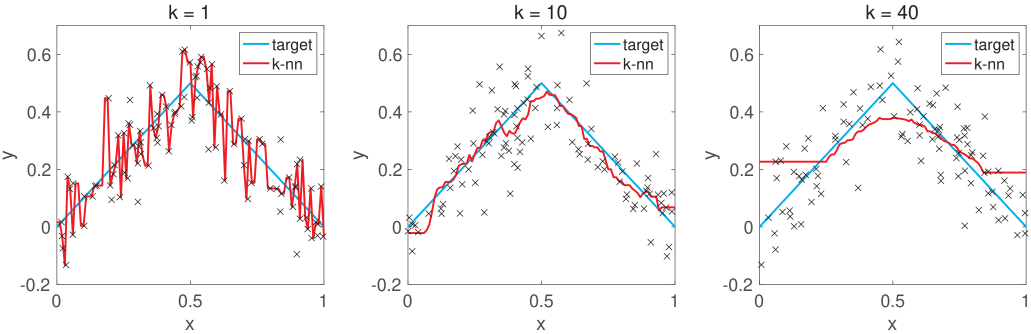
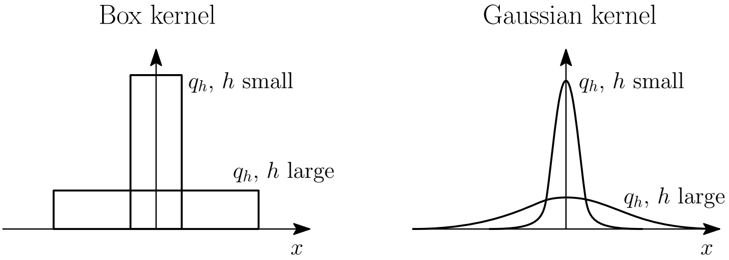


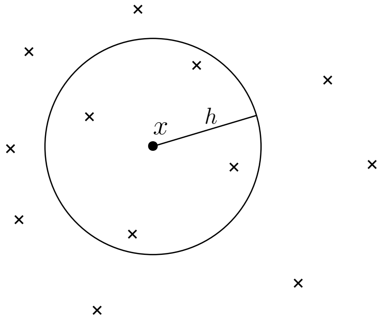
Figure 6.3: k-nearest neighbor regression in $d = 1$ dimension, with three values of k (the number of neighbors), with the same data as Figure 6.2. We can observe both underfitting (k too large) and overfitting (k too small).

In most cases where $\mathcal{X} \subset \mathbb{R}^d$, we take $k(x, x') = h^{-d} q\big(\frac{1}{h}(x - x')\big)$ for a certain function $q : \mathbb{R}^d \to \mathbb{R}_+$ that has large values around 0, and $h > 0$ a "bandwidth" parameter to be selected (see analysis in Section 6.3.3). If we assume that q is integrable with an integral equal to one, then $k(\cdot, x')$ is a probability density with mass around x', which gets more concentrated as h goes to zero. See the illustration below for the two typical kernel functions (sometimes called "windows").



Typical examples are:

- Box kernel: $q(x) \propto 1_{\|x\|_2 \leqslant 1}$, which leads to a weight functions $\hat{w}_i$ with many zeros. See below for an illustration in $d = 2$ dimensions.



- Gaussian kernel $q(x) \propto e^{-\|x\|^2/2}$, where we use the fact it is non-negative *pointwise*, as opposed to positive-definiteness in Chapter 7.[4] See a one-dimensional experiment in Figure 6.4.

In terms of algorithms, with a naive algorithm, for every test point, all the input data have to be considered, that is, a complexity proportional to n. The same techniques used for efficient k-nearest-neighbor search (e.g., k-d-trees) can also be applied here. Algorithms based on the fast Fourier transform can also be used (Silverman, 1982).

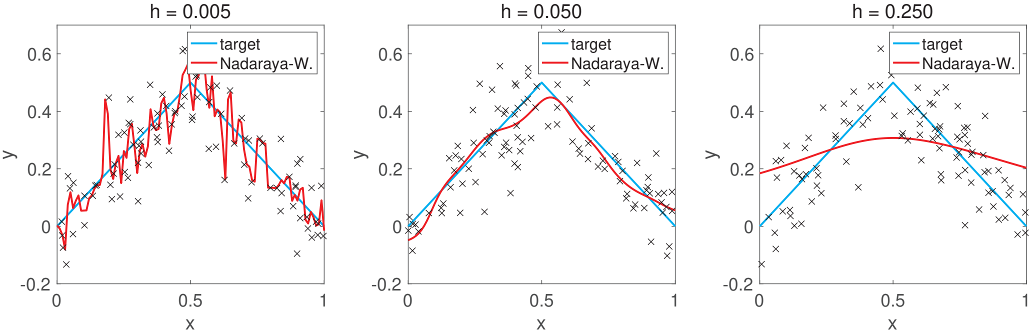


Figure 6.4: Nadaraya-Watson regression in $d = 1$ dimension, with three values of h (the bandwidth), for the Gaussian kernel, with the same data as Figure 6.2. We can observe both underfitting (h too large), or overfitting (h too small).

6.3 Generic "simplest" consistency analysis

We consider for simplicity the regression case. For classification, convex surrogate techniques such as those used in Section 4.1 can be used, with, for example, the square loss or the logistic loss (with then a square root calibration function on top of the least-squares excess risk, see Exercise 6.2 below). Still, better rates can be obtained directly (see, e.g., Chen and Shah, 2018; Biau and Devroye, 2015; Audibert and Tsybakov, 2007; Chaudhuri and Dasgupta, 2014).

We make the following generic simplifying assumptions (weaker ones could be considered with more involved proofs):

(H-1) Bounded noise: There exists $\sigma \geqslant 0$ such that $(y - \mathbb{E}(y|x))^2 \leqslant \sigma^2$ almost surely. We could also consider a weaker assumption that the conditional variance $\mathbb{E}\big[(y - \mathbb{E}(y|x))^2|x\big]$ is bounded by σ^2 almost surely.

(H-2) Regular target function: The target function $f^*(x) = \mathbb{E}(y|x)$ is B-Lipschitz-continuous with respect to a distance d. For weaker assumptions, see Section 6.4.

We have, with the target function $f^*(x) = \mathbb{E}(y|x)$, at a test point $x \in \mathcal{X}$ (and using that

[4] See also https://francisbach.com/cursed-kernels/

the weights $\hat{w}_i(x)$ sum to one):

$$\begin{aligned}
\hat{f}(x) - f^*(x) &= \sum_{i=1}^n y_i \hat{w}_i(x) - \mathbb{E}(y|x) \\
&= \sum_{i=1}^n \hat{w}_i(x)\big[y_i - \mathbb{E}(y_i|x_i)\big] + \sum_{i=1}^n \hat{w}_i(x)\big[\mathbb{E}(y_i|x_i) - \mathbb{E}(y|x)\big] \\
&= \sum_{i=1}^n \hat{w}_i(x)\big[y_i - \mathbb{E}(y_i|x_i)\big] + \sum_{i=1}^n \hat{w}_i(x)\big[f^*(x_i) - f^*(x)\big].
\end{aligned}$$

Given $x_1, \dots, x_n$ (and *because we have assumed the weight functions do not depend on the labels*), the left term has zero expectation, while the right term is deterministic. We thus have, using the independence of all (x_i, y_i), $i = 1, \dots, n$, and for x fixed:

$$\begin{aligned}
& \mathbb{E}\big[(\hat{f}(x) - f^*(x))^2\big|x_1, \dots, x_n\big] \\
= \; & (\mathbb{E}(\hat{f}(x)\big|x_1, \dots, x_n) - f^*(x))^2 + \operatorname{var}\big[\hat{f}(x)\big|x_1, \dots, x_n\big] \\
= \; & \Big[\sum_{i=1}^n \hat{w}_i(x)\big[f^*(x_i) - f^*(x)\big]\Big]^2 + \sum_{i=1}^n \hat{w}_i(x)^2 \mathbb{E}\big[\big(y_i - \mathbb{E}(y_i|x_i)\big)^2\big|x_i\big] \\
= \; & \qquad\qquad \text{bias} \qquad\qquad + \quad \text{variance},
\end{aligned}$$

with a “bias” term which is zero if f^* is constant,[5] and a “variance” term which is zero, when y is a deterministic function of x (i.e., $\sigma = 0$). Note that at this point, we only had equalities in the argument; we can now upper-bound as:

$$\begin{aligned}
& \mathbb{E}\big[(\hat{f}(x) - f^*(x))^2\big|x_1, \dots, x_n\big] \\
\leqslant \; & \Big[\sum_{i=1}^n \hat{w}_i(x)\big|f^*(x_i) - f^*(x)\big|\Big]^2 + \sigma^2 \sum_{i=1}^n \hat{w}_i(x)^2 \text{ using (H-1)}, \qquad (6.2) \\
\leqslant \; & \Big[\sum_{i=1}^n \hat{w}_i(x) B d(x_i, x)\Big]^2 + \sigma^2 \sum_{i=1}^n \hat{w}_i(x)^2 \text{ using (H-2)}, \\
\leqslant \; & B^2 \sum_{i=1}^n \hat{w}_i(x) d(x_i, x)^2 + \sigma^2 \sum_{i=1}^n \hat{w}_i(x)^2 \text{ using Jensen's inequality.}
\end{aligned}$$

Note that in the last inequality, having the weight vector $\hat{w}(x)$ in the simplex is crucial. We then have for the expected excess risk this generic bound we will use for all three cases (partitions, k-nn, and Nadaraya-Watson):

$$\int_{\mathcal{X}} \mathbb{E}[(\hat{f}(x) - f^*(x))^2] dp(x) \leqslant B^2 \int_{\mathcal{X}} \mathbb{E}\Big[\sum_{i=1}^n \hat{w}_i(x) d(x_i, x)^2\Big] dp(x) + \sigma^2 \sum_{i=1}^n \int_{\mathcal{X}} \mathbb{E}[\hat{w}_i(x)^2] dp(x). \qquad (6.3)$$

[5] What we call bias in this book is sometimes referred to as the squared bias.

⚠ The expectation is with respect to the training data. The expectation with respect to the testing point x is kept as an integral to avoid confusion.

This upper bound can be divided into two terms:

- A variance term $\sigma^2 \sum_{i=1}^n \int_{\mathcal{X}} \mathbb{E}[\hat{w}_i(x)^2] dp(x)$, that depends on the noise on top of the optimal predictions. Since the weights sum to one, we can write $\sum_{i=1}^n \mathbb{E}[\hat{w}_i(x)^2] = \sum_{i=1}^n \mathbb{E}[(\hat{w}_i(x) - 1/n)^2] + 2/n - 1/n^2$, that is, up to vanishing constant, the variance term measures the deviation to uniform weights.
- A bias term $B^2 \int_{\mathcal{X}} \mathbb{E}\Big[\sum_{i=1}^n \hat{w}_i(x) d(x_i, x)^2 \Big] dp(x)$, which depends on the regularity of the target function through the constant B. It will be small if the weight vectors $\hat{w}(x)$ put most of its mass on observations x_i that are close to x.

This leads to two conditions: variance and bias have to go to zero when n grows, corresponding to two simple expressions that depend on the weights. For the variance, the worst case scenario is that $\hat{w}_i(x)^2 \approx \hat{w}_i(x)$, that is, weights are putting all the mass into a single label (usually different for different testing point), thus leading to overfitting. For the bias, the worst-case scenario is that weights are uniform (leading to underfitting).

In the following, we will specialize it for $\mathcal{X}$ a subset of $\mathbb{R}^d$, with a distribution with a density with some minor regularity properties (all will have compact support, that is, $\mathcal{X}$ compact), where we show that a proper setting of the hyperparameters leads to "good" predictions. This will be done for all three cases of local averaging methods.

We look at universal consistency in Section 6.4, where we will list assumption (H-2).

Exercise 6.2 *For the binary classification problem, with $\mathcal{Y} = \{-1, 1\}$, assume that $f^*(x) = \mathbb{E}(y|x)$ is B-Lipschitz-continuous. Using Section 4.1.4, show that the excess risk of the majority vote is upper-bounded by*

$$\bigg(B^2 \int_{\mathcal{X}} \mathbb{E}\Big[\sum_{i=1}^n \hat{w}_i(x) d(x_i, x)^2 \Big] dp(x) + \sigma^2 \sum_{i=1}^n \int_{\mathcal{X}} \mathbb{E}[\hat{w}_i(x)^2] dp(x) \bigg)^{1/2}.$$

6.3.1 Fixed partition

For the partitioning estimate defined in Section 6.2.2, we can prove the following convergence rate.

Proposition 6.1 (Convergence rate for partition estimates) *Assume bounded noise (H-1) and a Lipschitz-continuous target function (H-2), and a partition of the bounded support $\mathcal{X}$ of p, as $\mathcal{X} = \bigcup_{j \in J} A_j$; then for the partitioning estimate $\hat{f}$, we have:*

$$\int_{\mathcal{X}} \mathbb{E}[(\hat{f}(x) - f^*(x))^2] dp(x) \leqslant \Big(8\sigma^2 + \frac{B^2}{2} \mathrm{diam}(\mathcal{X})^2 \Big) \frac{|J|}{n} + B^2 \max_{j \in J} \mathrm{diam}(A_j)^2. \qquad (6.4)$$

Optimal trade-off between bias and variance. Before we look at the proof (which is based on Eq. (6.3)), we can look at the consequence of the bound in Eq. (6.4). We need to balance the terms (up to constants) $\max_{j \in J} \mathrm{diam}(A_j)^2$ and $\frac{|J|}{n}$. In the simplest

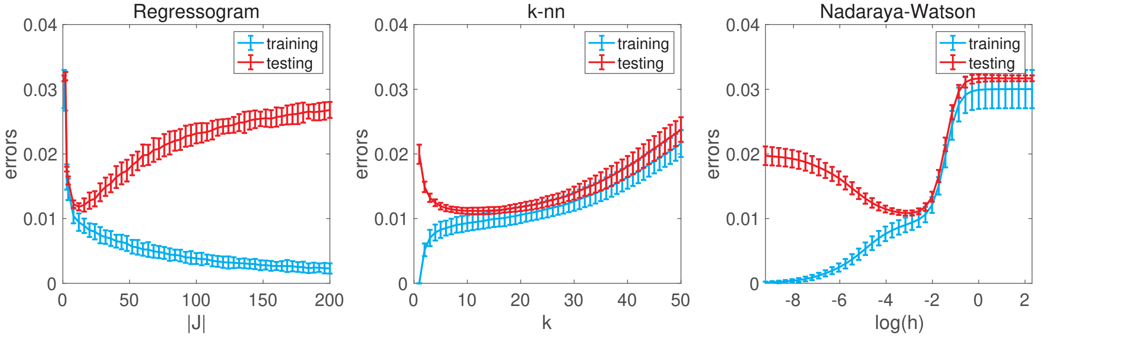


Figure 6.5: Learning curves for all three local averaging methods as a function of the corresponding hyperparameter. Left: regressogram (hyperparameter = number $|J|$ of sets in the partition), middle: k-nearest-neighbors (hyperparameter = number of neighbors k), right: Nadaraya-Watson (hyperparameter = bandwidth h). In all three cases, we see a trade-off between under-fitting and over-fitting.

situation of the unit-cube $[0,1]^d$, with $|J| = h^{-d}$ cubes of length h, we get $\frac{|J|}{n} = \frac{1}{nh^d}$ and $\max_{j\in J} \operatorname{diam}(A_j)^2 = h^2$, which, with $h = n^{-1/(2+d)}$ to make them equal, leads to a rate proportional to $n^{-2/(2+d)}$. As shown by Györfi et al. (2006), this rate is optimal for the estimation of Lipschitz-continuous functions (see Chapter 15).

While optimal, this is a very slow rate and a typical example of the curse of dimensionality. For this rate to be small, n has to be exponentially large in dimension. This is unavoidable with so little regularity (only bounded first-order derivatives). In Chapter 7 (and also in Section 6.5), we show how to leverage the smoothness of the target function to get significantly improved bounds (local averaging cannot take strong advantage of such smoothness). In Chapter 8, we will leverage dependence on a small number of variables.

Experiments. For the problem shown in Section 6.2, we plot in Figure 6.5 (left plot) training and testing errors averaged over 32 replications (with error bars showing the standard deviations), where we clearly see the trade-off in the choice of $|J|$.

Proof of Proposition 6.1 (♦) We consider an element A_j of the partition with at least one observation in it (a non-empty cell). Then for $x \in A_j$, and i among the indices of the points lying in A_j, $\hat{w}_i(x) = 1/n_{A_j}$ where $n_{A_j} \in \{1, \dots, n\}$ is the number of data points lying in A_j.

Variance. From Eq. (6.3), the variance term is bounded from above by σ^2 times

$$\sum_{i=1}^{n} \hat{w}_i(x)^2 = n_{A_j} \frac{1}{n_{A_j}^2} = \frac{1}{n_{A_j}}.$$

If A_j contains no input observations, then all weights are equal to $1/n$, and this sum is equal to $n \times \frac{1}{n^2} = 1/n$ for all $x \in A_j$. Thus, we get

$$\begin{aligned}\int_{\mathcal{X}} \mathbb{E}\Big[\sum_{i=1}^n \hat{w}_i(x)^2\Big] dp(x) &= \int_{\mathcal{X}} \sum_{j \in J} 1_{x \in A_j} \mathbb{E}\Big[\frac{1}{n_{A_j}} 1_{n_{A_j}>0} + \frac{1}{n} 1_{n_{A_j}=0}\Big] dp(x) \\ &= \sum_{j \in J} \mathbb{P}(A_j) \cdot \mathbb{E}\Big[\frac{1}{n_{A_j}} 1_{n_{A_j}>0} + \frac{1}{n} 1_{n_{A_j}=0}\Big].\end{aligned}$$

Intuitively, by the law of large numbers, n_{A_j}/n tends to $\mathbb{P}(A_j)$, so the variance term is expected to be of the order $\sigma^2 \sum_{j \in J} \mathbb{P}(A_j) \frac{1}{n\mathbb{P}(A_j)} = \sigma^2 \frac{|J|}{n}$, which is to be expected as this is essentially equivalent to least-squares regression with $|J|$ features $(1_{x \in A_j})_{j \in J}$.

More formally, we have $\mathbb{P}(n_{A_j} = 0) = (1 - \mathbb{P}(A_j))^n$, and, using Bernstein's inequality (see Section 1.2.3) for the random variables $1_{x_i \in A_j}$, which have mean and variance upper bounded by $\mathbb{P}(A_j)$, we have: $\mathbb{P}\big(\frac{n_{A_j}}{n} \leqslant \frac{1}{2}\mathbb{P}(A_j)\big) = \mathbb{P}\big(\frac{n_{A_j}}{n} \leqslant \mathbb{P}(A_j) - \frac{1}{2}\mathbb{P}(A_j)\big) \leqslant \exp\big(- \frac{n\mathbb{P}(A_j)^2/4}{2\mathbb{P}(A_j)+2(\mathbb{P}(A_j)/2)/3}\big) \leqslant \exp(-n\mathbb{P}(A_j)/10) \leqslant \frac{5}{n\mathbb{P}(A_j)}$, where we have used $\alpha e^{-\alpha} \leqslant 1/2$ for any $\alpha \geqslant 0$. This leads to the bound

$$\begin{aligned}\sum_{j \in J} \mathbb{P}(A_j)\, \mathbb{E}\Big[\frac{1_{n_{A_j}>0}}{n_{A_j}} + \frac{1_{n_{A_j}=0}}{n}\Big] &\leqslant \sum_{j \in J} \mathbb{P}(A_j)\, \mathbb{E}\Big[\frac{1_{1 \leqslant n_{A_j} \leqslant \frac{n}{2}\mathbb{P}(A_j)}}{n_{A_j}} + \frac{1_{n_{A_j} > \frac{n}{2}\mathbb{P}(A_j)}}{n_{A_j}} + \frac{1_{n_{A_j}=0}}{n}\Big] \\ &\leqslant \sum_{j \in J} \mathbb{P}(A_j) \Big[\mathbb{P}\Big(\frac{n_{A_j}}{n} \leqslant \frac{\mathbb{P}(A_j)}{2}\Big) + \frac{2}{n\mathbb{P}(A_j)} + \frac{1}{n}\mathbb{P}(n_{A_j}=0)\Big] \\ &\leqslant \sum_{j \in J} \mathbb{P}(A_j)\, \mathbb{E}\Big[\frac{5}{n\mathbb{P}(A_j)} + \frac{2}{n\mathbb{P}(A_j)} + \frac{1}{n\mathbb{P}(A_j)}\Big] \leqslant \frac{8|J|}{n}.\end{aligned}$$

Bias. We have, for $x \in A_j$ and a non-empty cell,

$$\sum_{i=1}^n \hat{w}_i(x) d(x, x_i)^2 \leqslant \operatorname{diam}(A_j)^2,$$

with $\sum_{i=1}^n \hat{w}_i(x) d(x, x_i)^2 = \frac{1}{n}\sum_{i=1}^n d(x, x_i)^2 \leqslant \operatorname{diam}(\mathcal{X})^2$ for empty-cells. Thus, separating the cases $n_{A_j} = 0$ and $n_{A_j} > 0$:

$$\begin{aligned}\int_{\mathcal{X}} \mathbb{E}\Big[\sum_{i=1}^n \hat{w}_i(x) d(x, x_i)^2\Big] dp(x) &\leqslant \sum_{j \in J} \mathbb{P}(A_j)\, \mathbb{E}\Big[\operatorname{diam}(A_j)^2 1_{n_{A_j}>0} + 1_{n_{A_j}=0} \operatorname{diam}(\mathcal{X})^2\Big] \\ &\leqslant \sum_{j \in J} \mathbb{P}(A_j) \Big[\operatorname{diam}(A_j)^2 + (1 - \mathbb{P}(A_j))^n \operatorname{diam}(\mathcal{X})^2\Big] \\ &= \sum_{j \in J} \mathbb{P}(A_j) \operatorname{diam}(A_j)^2 + \sum_{j \in J} \mathbb{P}(A_j)(1 - \mathbb{P}(A_j))^n \cdot \operatorname{diam}(\mathcal{X})^2.\end{aligned}$$

Using that $u(1-u)^n \leqslant 1/(2n)$ for $u \geqslant 0$, we get

$$\int_{\mathcal{X}} \mathbb{E}\Big[\sum_{i=1}^n \hat{w}_i(x) d(x,x_i)^2\Big] dp(x) \leqslant \sum_{j \in J} \mathbb{P}(A_j) \mathrm{diam}(A_j)^2 + \frac{1}{2}\frac{|J|}{n} \times \mathrm{diam}(\mathcal{X})^2,$$

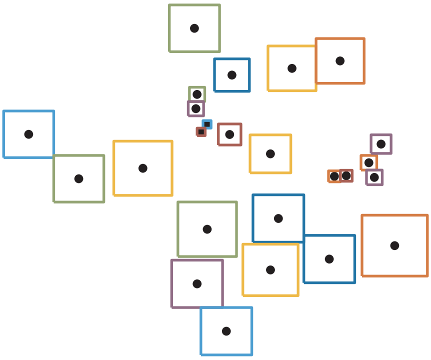
which leads to the desired term. ■

6.3.2 k-nearest neighbor

Here, since all weights are equal to zero except k of them, which are equal to $\frac{1}{k}$, we have $\sum_{i=1}^n \hat{w}_i(x)^2 = \frac{1}{k}$, so the variance term will go down as soon as k tends to infinity. For the bias term, the needed term $\sum_{i=1}^n \hat{w}_i(x) d(x_i, x)^2$ is equal to the average of the squared distances between x and its k-nearest neighbors within $\{x_1, \dots, x_n\}$, and this is less than the expected distance to the k-th nearest neighbor $x_{i_k(x)}$, for which the two following lemmas, taken from Biau and Devroye (2015, Theorem 2.4), give an estimate for the ℓ_∞-distance, and thus for all distances by equivalence of norms on $\mathbb{R}^d$.

Lemma 6.1 (distance to nearest neighbor) *Consider a probability distribution with compact support in $\mathcal{X} \subset \mathbb{R}^d$. Consider $n+1$ points $x_1, \dots, x_n, x_{n+1}$ sampled i.i.d. from $\mathcal{X}$. Then the expected squared ℓ_∞-distance between x_{n+1} and its first-nearest-neighbor is less than $4\frac{\mathrm{diam}(\mathcal{X})^2}{n^{2/d}}$ for $d \geqslant 2$, and less than $\frac{2}{n}\mathrm{diam}(\mathcal{X})^2$ for $d = 1$.*

Proof We denote by $x_{(i)}$ a nearest neighbor of x_i among the other n points. Since all $n+1$ points are i.i.d., we can permute the indices without changing the distributions, and all $\|x_i - x_{(i)}\|_\infty^2$ have the same distribution as $\|x_{n+1} - x_{(n+1)}\|_\infty^2$; thus, we can instead compute $\frac{1}{n+1}\sum_{i=1}^{n+1} \mathbb{E}\big[\|x_i - x_{(i)}\|_\infty^2\big]$. We denote by $R_i = \|x_i - x_{(i)}\|_\infty$ and assume for simplicity $R_i > 0$ for all i (the general case is left as an exercise). Then the sets $B_i = \{x \in \mathbb{R}^d,\ \|x - x_i\|_\infty < \frac{R_i}{2}\}$ are disjoint since for $i \neq j$, $\|x_i - x_j\|_\infty \geqslant \max\{R_i, R_j\}$. See the illustration below in two dimensions, with squares representing the sets B_i centered as x_i (represented as dots).



Moreover, their union has diameter less than $\mathrm{diam}(\mathcal{X}) + \mathrm{diam}(\mathcal{X}) = 2\mathrm{diam}(\mathcal{X})$. Thus, the volume of the union of all sets B_i, which is equal to the sum of their volumes, is less than $\big(2\mathrm{diam}(\mathcal{X})\big)^d$. Thus, we have: $\sum_{i=1}^{n+1} R_i^d \leqslant \big(2\mathrm{diam}(\mathcal{X})\big)^d$. Therefore, by Jensen's inequality, for $d \geqslant 2$,

$$\Big(\frac{1}{n+1}\sum_{i=1}^{n+1} R_i^2\Big)^{d/2} \leqslant \frac{1}{n+1}\sum_{i=1}^{n+1}(R_i)^d \leqslant \frac{2^d \mathrm{diam}(\mathcal{X})^d}{n+1},$$

leading to the desired result. For $d = 1$, we have $\big(\frac{1}{n+1}\sum_{i=1}^{n+1} R_i^2\big) \leqslant \mathrm{diam}(\mathcal{X})\big(\frac{1}{n+1}\sum_{i=1}^{n+1} R_i\big)$, which is less than $\frac{2}{n+1}\mathrm{diam}(\mathcal{X})^2$. ■

Lemma 6.2 (distance to k-nearest-neighbor) *Let $k \geqslant 1$. Consider a probability distribution with compact support in $\mathcal{X} \subset \mathbb{R}^d$. Consider $n+1$ points $x_1, \dots, x_n, x_{n+1}$ sampled i.i.d. from $\mathcal{X}$. Then the expected squared ℓ_∞-distance between x_{n+1} and its k-nearest-neighbor is less than $8\mathrm{diam}(\mathcal{X})^2\big(\frac{2k}{n}\big)^{2/d}$ for $d \geqslant 2$, and less than $\frac{8k}{n}\mathrm{diam}(\mathcal{X})^2$ for $d = 1$.*

Proof (♦) Without loss of generality, we assume $2k \leqslant n$ (otherwise, the proposed bounds are trivial). We can then divide randomly (and independently) the n first points into $2k$ sets of size approximately $\frac{n}{2k}$. We denote $x_{(k)}^j$ a 1-nearest neighbor of x_{n+1} within the j-th set. The squared distance from x_{n+1} to the k-nearest neighbor among all first n points is less than the k-th smallest of the distances $\|x_{n+1} - x_{(k)}^j\|_\infty^2$, $j \in \{1, \dots, 2k\}$, because we take a k-nearest neighbor over a smaller set. This k-th smallest distance is less than $\frac{1}{k}\sum_{j=1}^{2k}\|x_{n+1} - x_{(k)}^j\|_\infty^2$ (this is a general fact that the k-smallest element among non-negative p elements, is less than their sum divided by $p - k$, applied here for $p = k$).

Thus, using the lemma above on the 1-nearest-neighbor from $\frac{n}{2k}$ points, we get that the desired averaged distance is less than, for $d \geqslant 2$:

$$\frac{1}{k}\sum_{j=1}^{2k} 4\frac{\mathrm{diam}(\mathcal{X})^2}{(\frac{n}{2k})^{2/d}} = 8\frac{\mathrm{diam}(\mathcal{X})^2}{n^{2/d}}(2k)^{2/d}.$$

A similar argument can be extended to $d = 1$ (left as an exercise). ■

Putting things together, we get the following result for the consistency of k-nearest-neighbors.

Proposition 6.2 (Convergence rate for k-nearest-neighbors) *Assume bounded noise (H-1) and a Lipschitz-continuous target function (H-2) with an input distribution with bounded support $\mathcal{X}$. Then for the k-nearest-neighbor estimate $\hat{f}$ with the ℓ_∞-norm, we have, for $d \geqslant 2$:*

$$\int_{\mathcal{X}} \mathbb{E}[(\hat{f}(x) - f^*(x))^2]dp(x) \leqslant \frac{\sigma^2}{k} + 8B^2\mathrm{diam}(\mathcal{X})^2\Big(\frac{2k}{n}\Big)^{2/d}. \qquad (6.5)$$

Balancing the two terms above is obtained with $k \propto n^{2/(2+d)}$, and we get the same result as for the other local averaging schemes. See more details by Chen and Shah (2018) and Biau and Devroye (2015).

Exercise 6.3 *Show that if the Bayes rate is* 0 *(that is,* $\sigma = 0$*), then* 1*-nearest-neighbor is consistent.*

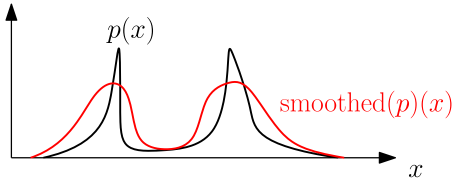
Experiments. For the problem shown in Section 6.2, we plot in Figure 6.5 (middle) training and testing errors averaged over 32 replications (with error bars showing the standard deviations), where we clearly see the trade-off in the choice of k.

6.3.3 Kernel regression (Nadaraya-Watson) (♦)

In this section, we assume that $\mathcal{X} = \mathbb{R}^d$, and for simplicity, we assume that the distribution of the inputs has a density (also denoted p) with respect to the Lebesgue measure. We also assume that the kernel is of the form $k(x, x') = q_h(x - x') = h^{-d} q(\frac{1}{h}(x - x'))$ for a probability density $q : \mathbb{R}^d \to \mathbb{R}_+$. The function q_h is also a density, which is the density of hz when z has density $q(z)$ (it thus gets more concentrated around 0 as h tends to zero). With these notations, the weights can be written:

$$\hat{w}_i(x) = \frac{q_h(x - x_i)}{\sum_{j=1}^n q_h(x - x_j)}.$$

Smoothing by convolution. When performing kernel regression, quantities of the form $\frac{1}{n}\sum_{i=1}^n q_h(x - x_i) g(x_i)$ naturally appear. When the number n of observations goes to infinity, and x is fixed, by the law of large numbers, it tends almost surely to $\int_{\mathbb{R}^d} q_h(x - z) g(z) p(z) dz$, which is exactly the convolution between the function q_h and the function $x \mapsto p(x)g(x)$, which we can denote $[(pg) * q_h](x)$. The function q_h is a probability density that puts most of its weights at a range of values of order h, e.g., for kernels like the Gaussian kernel or the box kernel. Thus, convolution will smooth the function pg by averaging values at range h. Therefore, when h goes to zero, it converges to the function pg itself. See an example below for $g = 1$.



Note that for this limit to hold, we need to ensure the factors in n and h^d are present.

We can now look at the generalization bound from Eq. (6.3) and see how it applies to kernel regression. We now consider the ℓ_2-distance for simplicity and consider the variance and bias terms separately, first with an asymptotic informal result where both h tends to zero and n tends to infinity, and then a formal result.

Variance term. We have, for a fixed $x \in \mathcal{X}$:

$$n \sum_{i=1}^{n} \hat{w}_i(x)^2 \quad = \quad \frac{\frac{1}{n}\sum_{i=1}^{n} q_h(x - x_i)^2}{\big(\frac{1}{n}\sum_{i=1}^{n} q_h(x - x_i)\big)^2}.$$

Using the law of large numbers and the smoothing reasoning above, this sum $n\sum_{i=1}^{n} \hat{w}_i(x)^2$ is converging almost surely to

$$\frac{\int_{\mathbb{R}^d} q_h(x-z)^2 p(z)dz}{\big(\int_{\mathbb{R}^d} q_h(x-z)p(z)dz\big)^2} = \frac{[q_h^2 * p](x)}{[q_h * p](x)^2}.$$

When h tends to zero, then the denominator above $[q_h * p](x)^2$ tends to $p(x)^2$ because the bandwidth of the smoothing goes to zero. The numerator above corresponds, up to a multiplicative constant, to the smoothing of p by the density $x \mapsto \frac{q_h(x)^2}{\int_{\mathbb{R}^d} q_h(u)^2 du}$, and is thus asymptotically equivalent to $p(x)\int_{\mathbb{R}^d} q_h(u)^2 du = p(x) h^{-d}\int_{\mathbb{R}^d} q(u)^2 du$.

Overall, when n tends to infinity, and h tends to zero, we get, asymptotically for x fixed:

$$\sum_{i=1}^{n} \hat{w}_i(x)^2 \sim \frac{1}{nh^d}\frac{1}{p(x)}\int_{\mathbb{R}^d} q(u)^2 du,$$

and thus, still asymptotically,

$$\int_{\mathcal{X}} \Big[\sum_{i=1}^{n} \hat{w}_i(x)^2\Big] p(x)dx \sim \frac{1}{nh^d}\text{vol}(\text{supp}(p)) \int_{\mathbb{R}^d} q(u)^2 du,$$

where vol(supp(p)) is the volume of the support of p in $\mathbb{R}^d$ (the closure of all x for which $p(x) > 0$), which we assume bounded.

Bias. With the same intuitive reasoning, we get when n tends to infinity:

$$\sum_{i=1}^{n} \hat{w}_i(x) d(x_i, x)^2 \quad \rightarrow \quad \frac{\int_{\mathbb{R}^d} q_h(x-z)\|x-z\|_2^2 p(z)dz}{\int_{\mathbb{R}^d} q_h(x-z)p(z)dz}.$$

The denominator has the same shape as for the variance term and tends to $p(x)$ when h tends to zero. With the change of variable $u = \frac{1}{h}(x-z)$, the numerator is equal to $\int_{\mathbb{R}^d} q_h(x-z)\|x-z\|_2^2 p(z)dz = h^2\int_{\mathbb{R}^d} q(u)\|u\|_2^2 p(x-uh)du$, which is equivalent to $h^2 p(x)\int_{\mathbb{R}^d} q(u)\|u\|_2^2 du$ when h tends to zero. Overall, when n tends to infinity, and h tends to zero, we get:

$$\int_{\mathcal{X}} \Big[\sum_{i=1}^{n} \hat{w}_i(x) d(x_i, x)^2\Big] p(x)dx \sim h^2 \int_{\mathbb{R}^d} q(u)\|u\|_2^2 du.$$

Therefore, overall we get an *asymptotic* bound proportional to (up to constants depending on q):

$$\frac{\sigma^2}{nh^d} + B^2 h^2,$$

leading to the same upper bound as for partitioning estimates by setting $h \propto n^{-1/(d+2)}$.

Formal reasoning (♦♦). We can make the informal reasoning above more formal using concentration inequalities, leading to non-asymptotic bounds of the same nature (simply more complicated) that make explicit the joint dependence on n and h. We will prove the following result:

Proposition 6.3 (Convergence rate for Nadaraya-Watson estimation) *Assume bounded noise (H-1) and a Lipschitz-continuous target function (H-2), and a function $q : \mathbb{R}^d \to \mathbb{R}$ such that $\int_{\mathbb{R}^d} q(z)dz = 1$, and $\|q\|_\infty = \sup_{z \in \mathbb{R}^d} q(z)$ is finite. Moreover, assume that p has bounded support $\mathcal{X}$ and density upper-bounded by $\|p\|_\infty$. Then for the Nadaraya-Watson estimate $\hat{f}$, we have:*

$$\int_{\mathcal{X}} \mathbb{E}[(\hat{f}(x) - f^*(x))^2]dp(x) \leqslant \Big[\frac{8\|q\|_\infty}{nh^d}\big(1 + \frac{1}{2}\mathrm{diam}(\mathcal{X})^2\big) + 2Bh^2\|p\|_\infty c\Big] \cdot C_h, \tag{6.6}$$

where $c = \int_{\mathbb{R}^d} q(u)\|u\|_2^2 du$ and $C_h = \int_{\mathcal{X}} \frac{p(x)}{[q_h * p](x)}dx$.

With additional assumptions, we can show that the constant C_h remains bounded when h tends to zero (see exercise below). Before giving the proof, we note that the optimal bandwidth parameter is indeed proportional to $h \propto n^{-1/(d+2)}$, with an overall excess risk proportional to $n^{-2/(d+2)}$, like the two other types of estimators.

Proof of Proposition 6.3 (♦) As for the proof for partitioning estimates, to deal with the denominator in the definition of the weights, we can first use Bernstein's inequality (see Section 1.2.3), applied to the random variables $q_h(x - x_i)$ which is almost surely in $[0, h^{-d}\|q\|_\infty]$, to bound

$$\mathbb{P}\Big(\frac{1}{n}\sum_{i=1}^n q_h(x - x_i) \leqslant \mathbb{E}[q_h(x-z)] - \varepsilon\Big) \leqslant \exp\Big(-\frac{n\varepsilon^2}{2\mathbb{E}[q_h^2(x-z)] + 2\|q\|_\infty h^{-d}\varepsilon/3}\Big).$$

We get with $\varepsilon = \frac{1}{2}\mathbb{E}[q_h(x-z)]$, using $\mathbb{E}[q_h^2(x-z)] \leqslant \|q\|_\infty h^{-d}\mathbb{E}[q_h(x-z)]$, for the event $\mathcal{A}(x) = \{\frac{1}{n}\sum_{i=1}^n q_h(x-x_i) \leqslant \frac{1}{2}\mathbb{E}[q_h(x-z)]\}$:

$$\begin{aligned}\mathbb{P}\big(\mathcal{A}(x)\big) \leqslant\ & \exp\Big(-\frac{\frac{n}{4}(\mathbb{E}[q_h(x-z)])^2}{2\mathbb{E}[q_h^2(x-z)] + \mathbb{E}[q_h(x-z)]h^{-d}\|q\|_\infty/3}\Big)\\ \leqslant\ & \exp\Big(-\frac{\frac{n}{4}\mathbb{E}[q_h(x-z)]}{(7/3)h^{-d}\|q\|_\infty}\Big) \leqslant \frac{\|q\|_\infty}{nh^d\mathbb{E}[q_h(x-z)]}\cdot\frac{1}{e}\frac{28}{3} \leqslant \frac{4\|q\|_\infty}{nh^d\mathbb{E}[q_h(x-z)]},\end{aligned} \tag{6.7}$$

where we have used $\alpha e^{-\alpha} \leqslant 1/e$ for $\alpha \geqslant 0$. We can now bound bias and variance.

Variance. For a fixed $x \in \mathcal{X}$, we get

$$\begin{aligned}\mathbb{E}\Big[\sum_{i=1}^n \hat{w}_i(x)^2\Big] &= \mathbb{E}\Big[1_{\mathcal{A}(x)}\sum_{i=1}^n \hat{w}_i(x)^2\Big] + \mathbb{E}\Big[1_{\mathcal{A}(x)^c}\sum_{i=1}^n \hat{w}_i(x)^2\Big]\\ &\leqslant \mathbb{P}(\mathcal{A}(x)) + \frac{4}{\big(n\mathbb{E}[q_h(x-z)]\big)^2}\mathbb{E}\Big[\sum_{i=1}^n q\big(\frac{1}{h}(x-x_i)\big)^2\Big]\\ &\leqslant \frac{4\|q\|_\infty}{nh^d\mathbb{E}[q_h(x-z)]} + \frac{4\mathbb{E}\big[q_h(x-z)^2\big]}{n\big[\mathbb{E}q_h(x-z)\big]^2} \leqslant \frac{8\|q\|_\infty}{nh^d\mathbb{E}[q_h(x-z)]}.\end{aligned}$$

Moreover, we have $\mathbb{E}[q_h(x-z)] = \int_{\mathbb{R}^d} dp(x-hu)q(u)du = [p*q_h](x)$. This leads to an overall bound on the variance term as $\displaystyle\int_{\mathcal{X}} \mathbb{E}\Big[\sum_{i=1}^n \hat{w}_i(x)^2\Big]p(x)dx \leqslant \frac{8\|q\|_\infty}{nh^d}\int_{\mathcal{X}} \frac{p(x)}{[p*q_h](x)}dx.$

Bias term. We have a similar reasoning for the bias term. Indeed, we get for a given $x \in \mathcal{X}$, using the bound in Eq. (6.7):

$$\begin{aligned}
& \mathbb{E}\Big[\sum_{i=1}^n \hat{w}_i(x)\|x-x_i\|_2^2\Big] \\
= \ & \mathbb{E}\Big[1_{\mathcal{A}(x)}\sum_{i=1}^n \hat{w}_i(x)\|x-x_i\|_2^2\Big] + \mathbb{E}\Big[1_{\mathcal{A}(x)^c}\sum_{i=1}^n \hat{w}_i(x)\|x-x_i\|_2^2\Big] \\
\leqslant \ & \mathbb{P}(\mathcal{A}(x)) \cdot \operatorname{diam}(\mathcal{X})^2 + \frac{2}{n\mathbb{E}[q_h(x-z)]} \cdot n\mathbb{E}[q_h(x-z)\|x-z\|_2^2] \\
\leqslant \ & \frac{4\|q\|_\infty}{nh^d[q_h*p](x)} \cdot \operatorname{diam}(\mathcal{X})^2 + \frac{2h^2}{[q_h*p](x)} \cdot \int_{\mathbb{R}^d} q(u)\|u\|_2^2 p(x-uh)du \\
\leqslant \ & \frac{4\|q\|_\infty}{nh^d[q_h*p](x)} \cdot \operatorname{diam}(\mathcal{X})^2 + \frac{2h^2\|p\|_\infty}{[q_h*p](x)} \cdot \int_{\mathbb{R}^d} q(u)\|u\|_2^2 du.
\end{aligned}$$

This leads to an overall bound on the bias term equal to $\displaystyle\int_{\mathcal{X}} \mathbb{E}\Big[\sum_{i=1}^n \hat{w}_i(x)\|x-x_i\|_2^2\Big]p(x)dx \leqslant \int_{\mathcal{X}} \frac{p(x)}{[q_h*p](x)}dx \cdot \bigg[\frac{4\|q\|_\infty}{nh^d}\operatorname{diam}(\mathcal{X})^2 + 2h^2\|p\|_\infty\Big(\int_{\mathbb{R}^d} q(u)\|u\|_2^2 du\Big)\bigg].$

Putting things together, we get that the excess risk $\int_{\mathcal{X}} \mathbb{E}[(\hat{f}(x) - f^*(x))^2]dp(x)$ is less than

$$\bigg[\frac{8\|q\|_\infty}{nh^d}\big(1+\frac{1}{2}\operatorname{diam}(\mathcal{X})^2\big) + 2Bh^2\|p\|_\infty\Big(\int_{\mathbb{R}^d} q(u)\|u\|_2^2 du\Big)\bigg] \cdot \int_{\mathcal{X}} \frac{p(x)}{[q_h*p](x)}dx,$$

which is exactly the desired bound. ■

Exercise 6.4 *Assume that the support $\mathcal{X}$ of the density p of inputs is bounded and that p is strictly positive and continuously differentiable on $\mathcal{X}$. Show that for h small enough (with an explicit upper-bound), then $C_h = \int_{\mathcal{X}} \frac{p(x)}{[q_h*p](x)}dx \leqslant \frac{1}{2}\mathrm{vol}(\mathcal{X})$.*

Experiments. For the problem shown in Section 6.2, we plot in Figure 6.5 (right) training and testing errors averaged over 32 replications (with error bars showing the standard deviations), where we clearly see the trade-off in the choice of h.

6.4 Universal consistency (♦)

Above, we have required the following conditions on the weights:

- $\displaystyle\int_{\mathcal{X}} \mathbb{E}\Big[\sum_{i=1}^n \hat{w}_i(x) d(x_i,x)^2\Big] dp(x) \to 0$ when n tends to infinity, to ensure that the bias goes to zero.
- $\displaystyle\int_{\mathcal{X}} \sum_{i=1}^n \mathbb{E}[\hat{w}_i(x)^2] dp(x) \to 0$ when n tends to infinity, to ensure that the variance goes to zero.

This was enough to show consistency when the target function is Lipschitz-continuous in $\mathbb{R}^d$. This also led to a precise rate of convergence, which turns out to be optimal for learning with target functions that are Lipschitz-continuous and for which the curse of dimensionality cannot be avoided (see Chapter 15).

To show universal consistency, that is, consistency for any square-integrable functions, we need an extra (technical) assumption, which was first outlined in Stone's theorem (Stone, 1977), namely that there exists $c > 0$ such that for any non-negative integrable function $h : \mathcal{X} \to \mathbb{R}$, then

$$\int_{\mathcal{X}} \sum_{i=1}^n \mathbb{E}\big[\hat{w}_i(x) h(x_i)\big] dp(x) \leqslant c \cdot \int_{\mathcal{X}} h(x) dp(x). \tag{6.8}$$

Below, h will be the squared deviation between two functions.

⚠ Above, we only take the expectation with respect to the training data, while we use the integral notation to take the expectation with respect to the training distribution.

Then for any $\varepsilon > 0$, and for any target function $f^* \in L_2(dp(x))$, we can find a function g which is $B(\varepsilon)$-Lipschitz-continuous and such that $\|f^* - g\|_{L_2(dp(x))} \leqslant \varepsilon$, because the set of Lipschitz-continuous functions is dense in $L_2(dp(x))$ (see, e.g., Ambrosio et al., 2013)).

Then we have, for a given $x \in \mathcal{X}$:

$$\begin{aligned}
& \mathbb{E}\Big(\Big[\sum_{i=1}^n \hat{w}_i(x)\big[f^*(x_i) - f^*(x)\big]\Big]^2\Big) \\
\leqslant \ & \mathbb{E}\Big(\Big[\sum_{i=1}^n \hat{w}_i(x)\Big(\big|f^*(x_i) - g(x_i)\big| + \big|g(x_i) - g(x)\big| + \big|g(x) - f^*(x)\big|\Big]^2\Big) \\
\leqslant \ & 3\mathbb{E}\Big(\Big[\sum_{i=1}^n \hat{w}_i(x)\big|f^*(x_i) - g(x_i)\big|\Big]^2\Big) + 3\mathbb{E}\Big(\Big[\sum_{i=1}^n \hat{w}_i(x)\big|g(x_i) - g(x)\big|\Big]^2\Big) \\
& +3\mathbb{E}\Big(\Big[\sum_{i=1}^n \hat{w}_i(x)\big|g(x) - f^*(x)\big|\Big]^2\Big) \text{ using the inequality } (a+b+c)^2 \leqslant 3a^2+3b^2+]!3c^2, \\
\leqslant \ & 3\mathbb{E}\Big(\Big[\sum_{i=1}^n \hat{w}_i(x)\big|f^*(x_i) - g(x_i)\big|\Big]^2\Big) + 3\mathbb{E}\Big(\Big[\sum_{i=1}^n \hat{w}_i(x) B(\varepsilon) d(x,x_i)\Big]^2\Big) \\
& +3\mathbb{E}\Big(\big|g(x) - f^*(x)\big|^2\Big) \text{ since weights sum to one, and } g \text{ is Lipschitz-continuous.}
\end{aligned}$$

We can further upper-bound $\mathbb{E}\Big(\Big[\sum_{i=1}^n \hat{w}_i(x)\big[f^*(x_i)-f^*(x)\big]\Big]^2\Big)$ by

$$\begin{aligned}
& 3\mathbb{E}\Big[\sum_{i=1}^n \hat{w}_i(x)\big|f^*(x_i)-g(x_i)\big|^2\Big] + 3B(\varepsilon)^2\mathbb{E}\Big(\sum_{i=1}^n \hat{w}_i(x)d(x,x_i)^2\Big) \\
& +3\mathbb{E}\Big(\big|g(x)-f^*(x)\big|^2\Big) \text{ using Jensen's inequality on the second term,} \\
\leqslant \ & 3c\cdot\mathbb{E}\big[\big|f^*(x)-g(x)\big|^2\big] + 3B(\varepsilon)^2\mathbb{E}\Big(\sum_{i=1}^n \hat{w}_i(x)d(x,x_i)^2\Big) + 3\mathbb{E}\Big(\big|g(x)-f^*(x)\big|^2\Big),
\end{aligned}$$

using Eq. (6.8). We can now integrate with respect to x to get

$$\begin{aligned}
& \int_{\mathcal{X}} \mathbb{E}\Big(\Big[\sum_{i=1}^n \hat{w}_i(x)\big[f^*(x_i)-f^*(x)\big]\Big]^2\Big)dp(x) \\
& \qquad \leqslant 3c\cdot\varepsilon^2 + 3B(\varepsilon)^2\int_{\mathcal{X}}\mathbb{E}\Big(\sum_{i=1}^n \hat{w}_i(x)d(x,x_i)^2\Big)dp(x) + 3\varepsilon^2. \qquad (6.9)
\end{aligned}$$

Proving universal consistency. We can then combine the bound above (which gives a bound on the bias) with Eq. (6.2), starting from the excess risk $\int_{\mathcal{X}}\mathbb{E}\big[(\hat{f}(x)-f^*(x))^2\big]dp(x)$ less than

$$\int_{\mathcal{X}}\mathbb{E}\Big(\Big[\sum_{i=1}^n \hat{w}_i(x)\big|f^*(x_i)-f^*(x)\big|\Big]^2\Big)dp(x) + \sigma^2\int_{\mathcal{X}}\mathbb{E}\big[\sum_{i=1}^n \hat{w}_i(x)^2\big]dp(x),$$

which is the sum of a bias term and a variance term, and for which, together with Eq. (6.9), we can use the same tools for consistency as for Eq. (6.3).

To prove universal consistency, we fix a certain $\varepsilon > 0$, from which we obtain some Lipschitz constant $B(\varepsilon)$. For such a $B(\varepsilon)$, we know how to make the (squared) bias term $B(\varepsilon)^2\int_{\mathcal{X}}\mathbb{E}\Big(\sum_{i=1}^n \hat{w}_i(x)d(x,x_i)^2\Big)dp(x) + \sigma^2\int_{\mathcal{X}}\mathbb{E}\big[\sum_{i=1}^n \hat{w}_i(x)^2\big]dp(x)$ less than ε, by choosing appropriate hyperparameter and number of observations n (see previous sections). Thus, if the extra condition in Eq. (6.8) is satisfied, these three methods are universally consistent. Note that, in general, n has to grow unbounded when ε tends to zero without any a priori bound.

We can now look at the three cases:

- Partitioning: We have then $c = 2$, and we get universal consistency. Indeed, using the same notations as in Section 6.2.2, we have for any fixed $x \in A_j$, $j \in J$, and f a non-negative function:

$$\begin{aligned}
\sum_{i=1}^n \mathbb{E}\big[\hat{w}_i(x)f(x_i)\big] &= \mathbb{E}\Big[1_{n_{A_j}>0}\frac{1}{n_{A_j}}\sum_{i \text{ s.t. } x_i\in A_j} f(x_i) \ + 1_{n_{A_j}=0}\frac{1}{n}\sum_{i=1}^n f(x_i)\Big] \\
&= \mathbb{E}\Big[1_{n_{A_j}>0}\frac{1}{n_{A_j}}\sum_{i \text{ s.t. } x_i\in A_j} \mathbb{E}[f(x_i)|x_i\in A_j] + 1_{n_{A_j}=0}\frac{1}{n}\sum_{i=1}^n f(x_i)\Big] \\
&\leqslant \mathbb{E}[f(z)|z\in A_j] \ + \mathbb{E}[f(z)],
\end{aligned}$$

where z is distributed as x. Thus, integrating with respect to x and summing over $j \in J$, we get:

$$\int_{\mathcal{X}} \sum_{i=1}^{n} \mathbb{E}\big[\hat{w}_i(x) h(x_i)\big]\, dp(x) \leqslant \sum_{j \in J} \Big(\mathbb{P}(A_j)\mathbb{E}[f(z)|z \in A_j] + \mathbb{P}(A_j) \cdot \mathbb{E}[f(z)] \Big) = 2\mathbb{E}[f(z)],$$

which is exactly Eq. (6.8) with $c = 2$.

- Kernel regression: it can be shown using the same type of techniques outlined for consistency for Lipschitz-continuous functions.
- k-nearest neighbor: the condition in Eq. (6.8) is not easy to show and is often referred to as Stone's lemma. See Biau and Devroye (2015, Lemma 10.7).

6.5 Adaptivity (♦♦)

As shown above, all local averaging techniques achieve the same performance on Lipschitz-continuous functions, which is an unavoidable bad performance when d grows (curse of dimensionality). One extra order of smoothness, that is, on $\mathbb{R}^d$, two bounded derivatives, can be leveraged to lead to a convergence rate proportional to $n^{-4/(4+d)}$ (Wasserman, 2006, Section 5.4). However, the higher smoothness of the target function does not seem to be easy to leverage, that is, even if the target function is very smooth, the local averaging techniques will not be able to attain better convergence rates. The impossibility comes from the bias term which is the square of $\sum_{i=1}^n \hat{w}_i(x)\big[f^*(x_i) - f^*(x)\big]$ in Section 6.3: when f^* is once differentiable, $f^*(x_i) - f^*(x) = O(\|x_i - x\|)$ and this is what we leveraged in the proofs; when f^* is twice differentiable, by a Taylor expansion, $f^*(x_i) - f^*(x) = (x_i - x)^\top (f^*)'(x_i) + O(\|x_i - x\|^2)$, and we can choose weights so that $\sum_{i=1}^n \hat{w}_i(x)(x - x_i) = O(\|x - x_i\|^2)$ (this is possible because the components of $x - x_i$ may take positive and negative values, leading to potential cancellations, see exercise below); but when f is three-times differentiable or more, obtaining a term $O(\|x_i - x\|^3)$ that would come from a Taylor expansion, is only possible if the weights satisfy $\sum_{i=1}^n \hat{w}_i(x)(x - x_i)(x - x_i)^\top = O(\|x_i - x\|^3)$, which is not possible when the weights are non-negative as no cancellations are possible.

Positive-definite kernel methods will provide simple ways in Chapter 7, as well as neural networks in Chapter 9, to leverage smoothness. Among local averaging techniques, there are, however, ways to do it. For example, using locally linear regression, where one solves for any test point x,

$$\inf_{\beta_1 \in \mathbb{R}^d,\ \beta_0 \in \mathbb{R}} \sum_{i=1}^{n} \hat{w}_i(x)(y_i - \beta_1^\top x_i - \beta_0)^2.$$

(note that the regular regressogram corresponds to setting $\beta_1 = 0$ above). In other words we solve

$$\inf_{\beta_1 \in \mathbb{R}^d,\ \beta_0 \in \mathbb{R}} \int_{\mathcal{Y}} (y - \beta_1^\top x - \beta_0)^2 d\widehat{p}(y|x).$$

The running time is now $O(nd^2)$ per testing point as we have to solve a linear least-squares (see Chapter 3), but the performance, both empirical and theoretical (Tsybakov, 2008), improves. See an example with the regressogram weights in Figure 6.6.

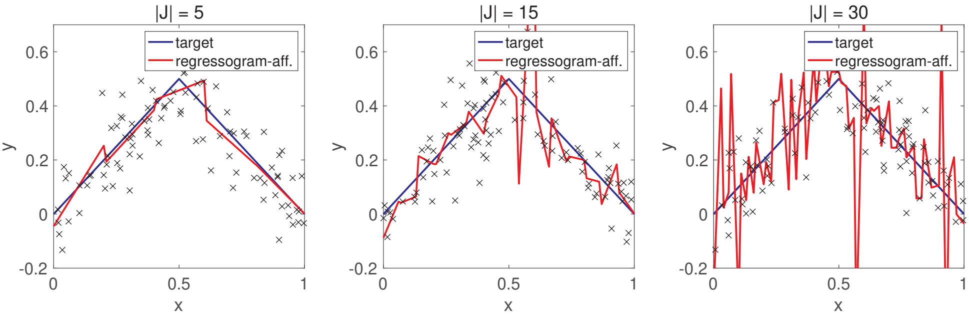


Figure 6.6: Locally linear regression, on the same data as Figure 6.2, for three values of the number $|J|$ of sets within in the partition. Notice the difference with Figure 6.2.

Exercise 6.5 *(♦) For the Nadaraya Watson estimator, show that when the target function and the kernel are twice continuously differentiable, then the bias term is bounded by a constant times h^4. Show that the optimal bandwidth selection leads to a rate proportional to $n^{-4/(4+d)}$.*

6.6 Conclusion

In this chapter, we have explored local averaging methods, which leverage the explicit formula for the Bayes predictor and explicitly aim at approximating it, without the need for optimization (as opposed to all other methods presented in this book). While they can potentially adapt to complex prediction functions, they suffer from the curse of dimensionality, that is, the number of observations has to be exponential in dimension for good predictions. Without further assumptions, this is unavoidable, but in the following chapters, we will see that other learning techniques can take advantage of extra assumptions, such as the smoothness of the prediction function (kernels in Chapter 7 and neural networks in Chapter 9), and dependence only a linear projection of the inputs (this will only be possible for neural networks).

Like all techniques presented in this book, local averaging methods can also be used within "ensemble methods" that combine several predictors learned on modifications of the original dataset (see Chapter 10).