Learning theory from first principles

Lecture 5: Local averaging methods

Francis Bach

October 23, 2020

Class summary

-Partition estimators
-Nadaraya-Watson estimators
-K-nearest-neighbors
-Universal consistency

1 Introduction - Review of supervised learning

- 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 $dp(x, y)$.
- We consider a fixed (testing) distribution dp on $\mathcal{X} \times \mathcal{Y}$ and a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$; $\ell(y, z)$ is the loss of predicting z while the true label is y.

  ⚠ Like in the rest of the course, we assume that the testing distribution is the same as the training distribution.
- Our goal is to minimize the risk, or generalization performance of a prediction function $f : \mathcal{X} \to \mathcal{Y}$:

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

  ⚠ 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 Lecture 1, 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 Lecture 1, minimizing the expected risk leads to an optimal "target function", called the Bayes predictor $f^* \in \arg\min \mathcal{R}(f) = \mathbb{E}\Big[\ell(y, f(x))\Big]$.

  Proposition 1 (Bayes predictor) *The risk is minimized at a* Bayes predictor $f^* : \mathcal{X} \to \mathcal{Y}$ *satisfying for all* $x \in \mathcal{X}$, $f^*(x) \in \arg\min_{z \in \mathcal{Y}} \mathbb{E}(\ell(y, z)|x)$. *The Bayes risk* $\mathcal{R}^*$ *is the risk of all Bayes predictors and is equal to*

$$\mathcal{R}^* = \mathbb{E}_{x \sim dp(x)} \inf_{z \in \mathcal{Y}} \mathbb{E}(\ell(y, z)|x) = \mathbb{E}_{x \sim dp(x)} \inf_{z \in \mathcal{Y}} \mathbb{E}_{y \sim dp(y|x)}(\ell(y, z)|x).$$

  Note that (a) the Bayes predictor is not unique, but that all lead to the same Bayes risk, and (b) that the Bayes risk is usually non zero (unless the dependence between x and y is deterministic).

  - For binary classification: $\mathcal{Y} = \{0, 1\}$ and $\ell(y, z) = 1_{y \neq z}$, the Bayes predictor is $f^*(x) \in \arg\max_{i \in \{0, \dots, 1\}} \mathbb{P}(y = i|x)$. This extends to multi-category classification as $f^*(x) \in \arg\max_{i \in \{1, \dots, k\}} \mathbb{P}(y = i|x)$.
  - For regression: $\mathcal{Y} = \mathbb{R}$ and $\ell(y, z) = (y - z)^2$, the Bayes predictor is $f^*(x) = \mathbb{E}(y|x)$.

- The goal of supervised machine learning is thus to estimate f^*, knowing only the training data $\mathcal{D} = \{(x_1, y_1), \dots, (x_n, y_n)\}$ *and* the loss ℓ, with the goal of minimizing the risk or the following excess risk.

  Definition 1 (Excess risk) *The excess risk of a function from* $f : \mathcal{X} \to \mathcal{Y}$ *is equal to* $\mathcal{R}(f) - \mathcal{R}^*$ *(it is always non-negative).*

  For least-squares, we have $\mathcal{R}(f) - \mathcal{R}^* = \displaystyle\int_{\mathcal{X}} (f(x) - f^*(x))^2 dp(x) = \|f - f^*\|^2_{L_2(dp(x))}$.

- In Lectures 2 and 3 (and Lecture 6), we explored methods based on empirical risk minimization. In this lecture, we focus on local averaging methods.

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 $dp(y|x)$ of y given x, by some $d\widehat{p}(y|x)$.
- We then replace $f^*(x) \in \arg\min_{z\in\mathcal{Y}} \int_{\mathcal{Y}} \ell(y,z) dp(y|x)$ 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: $\hat{f}(x) \in \arg\max_{j\in\{1,\dots,k\}} \widehat{\mathbb{P}}(y=j|x)$.
  - For regression: $\mathcal{Y}=\mathbb{R}$: $\hat{f}(x) = \int_{\mathcal{Y}} y \; d\widehat{p}(y|x)$.

2.1 Linear estimators

In this lecture we will consider "linear" estimators, where

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

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

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

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

In most cases, for any i, the weight function $\hat{w}_i(x)$ is close to 1 for test points x which are close to x.

In the usual cases, 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 $\hat{w}_i(x)$.
- 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.

2.2 Partition estimators

If $\mathcal{X} = \bigcup_{j\in J} A_j$ is a partition (such that for all $j, j' \in J$, $A_j \cap A_{j'} = \varnothing$) of $\mathcal{X}$ with a countable index set J, then we can consider for any x the 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

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

with the convention that if no data point lies in $A(x)$, then $\hat{w}_i(x)$ is equal to $1/n$ for each $i \in \{1, \dots, n\}$.

Note that this estimator can be seen as a least-squares estimator with feature vector $(1_{x \in A_j})_{j \in J}$: this feature vector leads to a diagonal (empirical or not) non-centered covariance matrix, for which we have a closed form estimation, which is equal to the estimator above (with a proper convention when no data point lies in $A(x)$).

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

There are two standard applications of partition estimators:

- Fixed partitions: for example, when $\mathcal{X} = [0,1]^d$, then we 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 below).

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 (see more details in `https://en.wikipedia.org/wiki/Decision_tree`). Note here that the partition depends on the labels (so the analysis below does not apply, unless the partitioning is learned on a different data than the one used for the estimation).

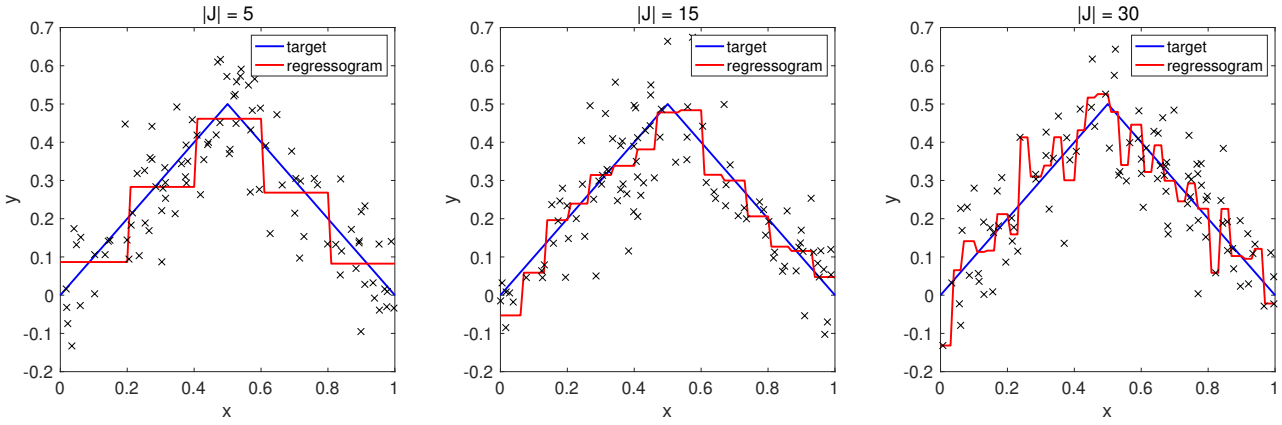


Figure 1: Regressograms in $d = 1$ dimension, with three values of $|J|$ (the number of sets in the partition). We can observe both underfitting ($|J|$ too small), or overfitting ($|J|$ too large). Note that the target function f^* is piecewise affine, and that on the affine parts, the estimator is far from linear, that is, the estimator cannot take advantage of extra-regularity (see Section 5 for more details).

2.3 Nearest-neighbors

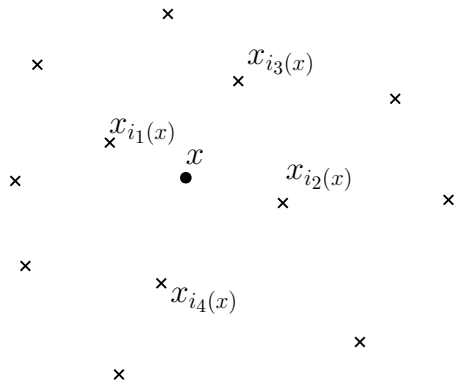
Given an integer $k \geqslant 1$, and a distance d on $\mathcal{X}$, for any $x \in \mathcal{X}$, we can order the n observations so that

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

where $\{i_1(x), \ldots, i_n(x)\} = \{1, \ldots, n\}$, and ties are broken randomly (that is, by sampling priorities randomly for each i once for all $x \in \mathcal{X}$). We then define

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

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 $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ 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).



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. When n is large, this is costly in time and memory. There exists indexing techniques for (potentially approximate) nearest-neighbor search, such as "k-d-trees", with typically a logarithmic complexity in n (but with some additional compiling time) (see `https://en.wikipedia.org/wiki/K-d_tree`).

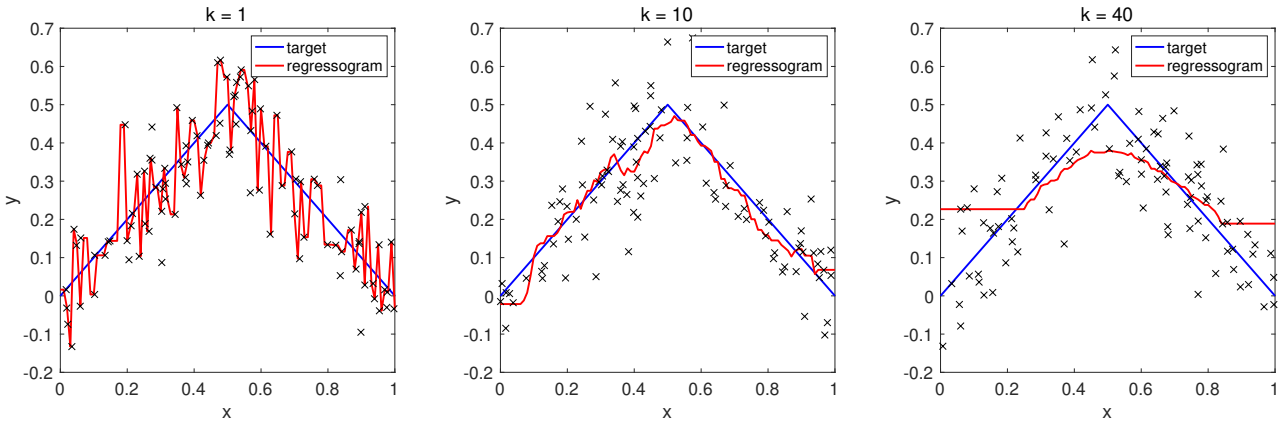


Figure 2: k-nearest neighbor regression in $d = 1$ dimension, with three values of k (the number of neighbors). We can observe both underfitting (k too large), or overfitting (k too small).

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

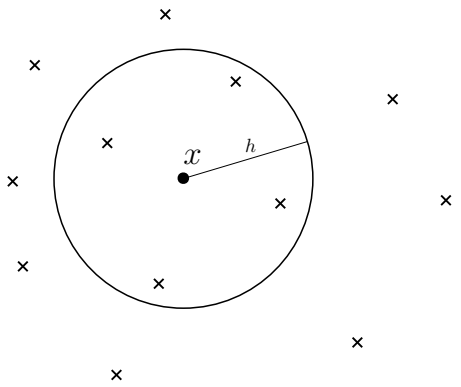
Given a "kernel" function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$, we define

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

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. In most cases where $\mathcal{X} \subset \mathbb{R}^d$, we take $k(x, x') = q\big(\frac{1}{h}(x - x_i)\big)$ for a certain function $q : \mathbb{R}^d \to \mathbb{R}$, and $h > 0$ a "bandwidth" parameter to be selected.

Typical examples are:

- Box kernel: $q(x) = 1_{\|x\|_2 \leqslant 1}$. See below for an illustration in $d = 2$ dimensions.



- Gaussian kernel $q(x) = e^{-\|x\|^2/2}$, where we use the fact it is non-negative *pointwise* (as opposed to positive-definiteness in Lecture 6, see `https://francisbach.com/cursed-kernels/`).

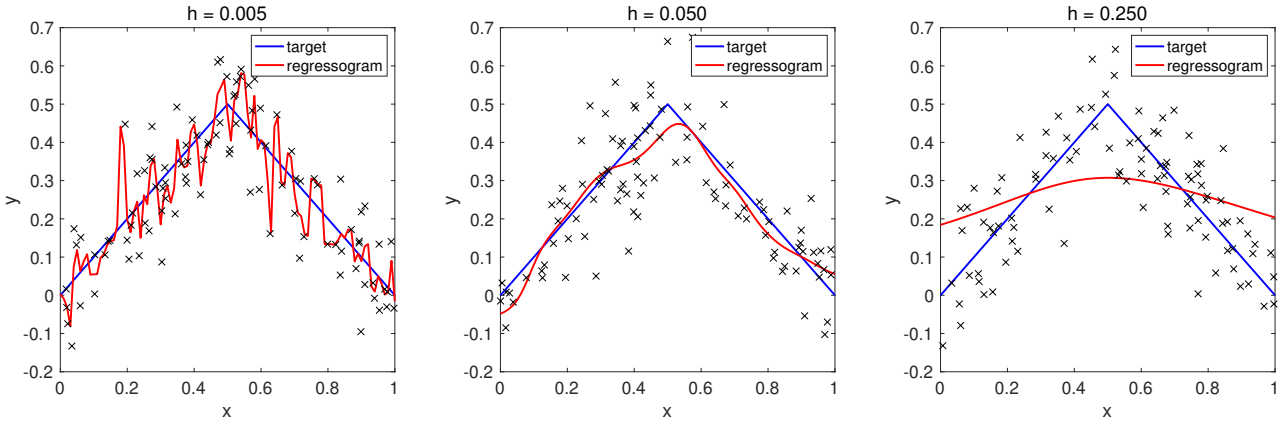


Figure 3: Nadaraya-Watson regression in $d = 1$ dimension, with three values of h (the bandwidth), for the Gaussian kernel. We can observe both underfitting (h too large), or overfitting (h too small).

3 Generic "simplest" consistency analysis

We consider for simplicity the regression case. For classification, calibration techniques like used in Lecture 3 can be used (with then a square root calibration function on top of the least-squares excess risk), but better rates can be obtained directly (see, e.g., [1, 2, 3, 4]).

We make the following generic assumptions:

(H1) Bounded noise: There exists $\sigma \geqslant 0$ such that $|y - \mathbb{E}(y|x)| \leqslant \sigma^2$ almost surely.

(H2) Regular target function: The target function f^* is B-Lipschitz-continuous with respect to the distance d. For weaker assumptions, see Section 4.

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

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

With respect to $x_1, \dots, x_n$, 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:

$$\begin{aligned}
\mathbb{E}\big[(\hat{f}(x) - f^*(x))^2 \big| x_1, \dots, x_n\big] &= (\mathbb{E}(\hat{f}(x)\big|x_1, \dots, x_n) - f^*(x))^2 + \mathrm{var}\big[\hat{f}(x)\big|x_1, \dots, x_n\big] \\
&= \Big[\sum_{i=1}^n \hat{w}_i(x)\big[f^*(x_i) - f^*(x)\big]\Big]^2 + \sum_{i=1}^n \hat{w}_i(x)^2 \mathbb{E}\big[\big(y_i - \mathbb{E}(y_i|x_i)\big)^2 \big| x_i\big] \\
&\leqslant \Big[\sum_{i=1}^n \hat{w}_i(x)\big|f^*(x_i) - f^*(x)\big|\Big]^2 + \sigma^2 \sum_{i=1}^n \hat{w}_i(x)^2 \text{ using (H1)}, \\
&\leqslant \Big[\sum_{i=1}^n \hat{w}_i(x) B d(x_i, x)\Big]^2 + \sigma^2 \sum_{i=1}^n \hat{w}_i(x)^2 \text{ using (H2)}, \\
&\leqslant B^2 \sum_{i=1}^n \hat{w}_i(x) d(x_i, x)^2 + \sigma^2 \sum_{i=1}^n \hat{w}_i(x)^2 \text{ using Jensen's inequality.}
\end{aligned}$$

Thus, the expected excess risk, which is equal to $\int_{\mathcal{X}} \mathbb{E}[(\hat{f}(x) - f^*(x))^2] dp(x)$, is less than

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

This upper bound can be divided into:

- A variance term $\sigma^2 \sum_{i=1}^n \mathbb{E}[\hat{w}_i(x)^2]$, 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 measure 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.

This leads to two conditions: both variance and bias have to go to zero when n grows, and this corresponds to two simple quantities 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 density $dp(x)$ with some minor regularity properties (all will have compact suppport, that is, $\mathcal{X}$ compact), where we show that a proper setting of the hyperparameters leads to "good" predictions.

We look at universal consistency in Section 4.

3.1 Fixed partition

In this situation, we consider an element A_j of the partition with at least one observation in it. 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 $n_{A_j} \in \{0, \dots, n\}$ is the number of data points lying in A_j. Thus

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

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

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

Variance. Intuitively, by the law of large numbers, $\frac{n_{A_j}}{n}$ 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 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 for the random variables $1_{x_i \in A_j}$, which have mean and variance upper bounded by $\mathbb{P}(A_j)$, we have: $\mathbb{P}\Big(\frac{n_{A_j}}{n} \leqslant \mathbb{P}(A_j) - \frac{1}{2}\mathbb{P}(A_j)\Big) \leqslant \exp\big(-\frac{n\mathbb{P}(A_j)^2/4}{2\mathbb{P}(A_j) + 2(\mathbb{P}(A_j)/2)/3}\big) \leqslant \exp(-n\mathbb{P}(A_j)/10) \leqslant \frac{5}{n\mathbb{P}(A_j)}$, leading to a bound

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

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

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

with $\sum_{i=1}^{n} \hat{w}_i(x) d(x, x_i)^2 = \frac{1}{n} \sum_{i=1}^{n} d(x, x_i)^2 \leqslant \text{diam}(\mathcal{X})^2$ for empty-cells. Thus

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

An alternative bound is $\sum_{j \in J} \mathbb{P}(A_j) \text{diam}(A_j)^2 + \exp(-n \min_{j \in J} \mathbb{P}(A_j)) \times \text{diam}(\mathcal{X})^2$.

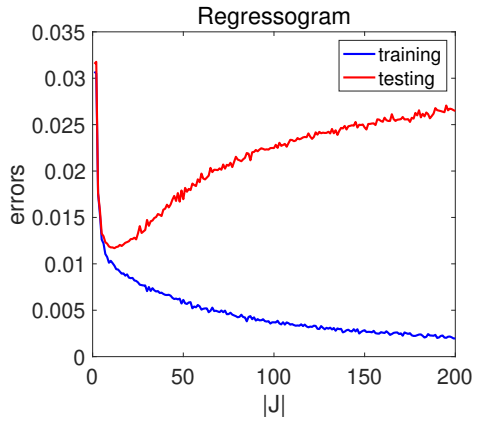
We thus need to balance the terms in $\sum_{j \in J} \mathbb{P}(A_j) \text{diam}(A_j)^2 \leqslant \max_{j \in J} \text{diam}(A_j)^2$ and in $\frac{|J|}{n}$,

In the simplest situation, of the unit-cube, with $|J| = h^{-d}$ cubes of length h, we get the two terms $\frac{|J|}{n} \propto \frac{1}{nh^d}$ and $\max_{j \in J} \text{diam}(A_j)^2 \propto h^2$, which, with $h \propto n^{-1/(2+d)}$ to make them equal, leads to a rate proportional to $n^{-2/(2+d)}$.

As shown in [5], this rate is optimal for estimation of Lipschitz-continuous function.

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 a single bounded derivatives). In Lecture 6 (and also in Section 5), we show how to leverage smoothness to get significantly improved bounds. In Lecture 7, we will leverage dependence on a small number of variables.

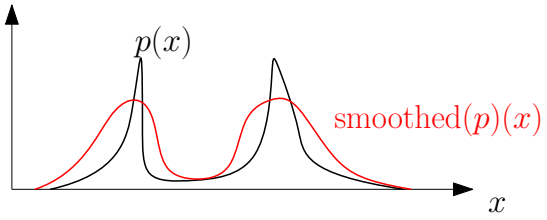
Experiments. For the problem shown in Section 2, below, we plot training and testing errors averaged over 10 replications, where we clearly see the trade-off in the choice of $|J|$.



3.2 Kernel regression (Nadaraya-Watson) (♦)

In this section, we assume that $\mathcal{X} = \mathbb{R}^d$, and for simplicity, we assume that $dp(x)$ has a density p with respect to the Lebesgue measure.

Smoothing by convolution. When performing kernel smoothing, quantities like $\frac{1}{n}\sum_{i=1}^n h^{-d} q\big(\frac{1}{h}(x - x_i)\big) g(x_i)$ naturally appear. When the number n of observations goes to infinity, by the law of large numbers, it tends almost surely to $\int_{\mathbb{R}^d} h^{-d} q\big(\frac{1}{h}(x-z)\big) g(z) p(z) dz$, which is exactly the convolution between the function $x \mapsto h^{-d} q(x/h)$ and the function $x \mapsto p(x)g(x)$. If we assume that $\int_{\mathbb{R}^d} q(z)dz = 1$, then $x \mapsto h^{-d} q(x/h)$ is a probability density that is putting all most its weights at range of values which are less than h, e.g., for kernels like the Gaussian kernel or the box kernel. Thus convolution will smooth the function pg by averaging values which are at range h. Thus, 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 make sure the factors in n and h^d are present.

Variance term. We now consider the ℓ_2-distance, and kernel regression. We have, for a fixed $x \in \mathcal{X}$:

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

For simplicity, we will assume that $dp(x)$ has a density on a compact support and that its density is in $[p_{\min}, p_{\max}]$.

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

$$\frac{\mathbb{E}_{z\sim dp(z)}\big[q\big(\frac{1}{h}(x-z)\big)^2\big]}{\Big(\mathbb{E}_{z\sim dp(z)}\big[q\big(\frac{1}{h}(x-z)\big)\big]\Big)^2} = \frac{\int_{\mathbb{R}^d} q\big(\frac{1}{h}(x-z)\big)^2 dp(z)}{\Big(\int_{\mathbb{R}^d} q\big(\frac{1}{h}(x-z)\big)dp(z)\Big)^2}$$

Using a change of variable $u = \frac{1}{h}(x-z)$, it is equal to

$$\frac{\int_{\mathbb{R}^d} h^{-d} q\big(\frac{1}{h}(x-z)\big)^2 dp(z)}{\Big(h^{-d}\int_{\mathbb{R}^d} q\big(\frac{1}{h}(x-z)\big)dp(z)\Big)^2} = h^{-d}\frac{\int_{\mathbb{R}^d} q\big(u\big)^2 dp(x-hu)}{\Big(\int_{\mathbb{R}^d} q\big(u\big)dp(x-hu)\Big)^2}.$$

When h tends to zero, and $dp(x)$ has a continuous density $p(x)$ with respect to the Lebesgue measure, it tends to $\frac{\int_{\mathbb{R}^d} q^2(y)dy}{(\int_{\mathbb{R}^d} q(y)dy)^2} h^{-d} p(x)^{-1}$, and when we take the expectation with respect to $p(x)$, we get a constant. Moreover, with the same intuitive reasoning, we get:

$$\sum_{i=1}^n \hat{w}_i(x) d(x_i, x)^2 \quad \to \quad h^2 \frac{\mathbb{E}_{z\sim dp(z)}\big[\|\frac{1}{h}(x-z)\|^2 q\big(\frac{1}{h}(x-z)\big)\big]}{\mathbb{E}_{z\sim dp(z)}\big[q\big(\frac{1}{h}(x-z)\big)\big]}$$

which tends to $h^2\|x\|^2$, leading to an expectation of $h^2\mathbb{E}\|x\|^2$.

Therefore, overall we gate a bound proportional to

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

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).

We can first use Bernstein inequality, applied to the random variables $q(\frac{1}{h}(x-x_i))$, to bound

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

If we assume that $dp(x)$ has a density $p(x)$ which is in $[p_{\min}, p_{\max}]$. We get:

$$\mathbb{P}\big(\frac{1}{n}\sum_{i=1}^n q(\frac{1}{h}(x-x_i)) \leqslant p_{\min}h^d\|q\|_1 - \varepsilon\big) \leqslant \exp\Big(-\frac{n\varepsilon^2}{2p_{\max}h^d\|q\|_2^2 + 2\|q\|_\infty\varepsilon/3}\Big),$$

which we can apply to $\varepsilon = \frac{1}{2}p_{\min}h^d\|q\|_1$, leading to

$$\mathbb{P}\big(\mathcal{A}(x)\big) \leqslant \exp\Big(-\frac{np_{\min}^2h^d\|q\|_1^2/4}{2p_{\max}\int_{\mathbb{R}^d} q^2(y)dy + \|q\|_\infty p_{\min}\|q\|_1/3}\Big),$$

where $\mathcal{A}(x)$ is the event $\mathcal{A} = \{\frac{1}{n}\sum_{i=1}^n q(\frac{1}{h}(x-x_i)) \leqslant p_{\min}h^d\|q\|_1/2\}$. In summary, we get, $\mathbb{P}(\mathcal{A}(x)) \leqslant \exp(-\square nh^d) \leqslant \frac{1}{2\square nh^d}$ for a certain positive number $\square$.

This allows to show that for fixed x,

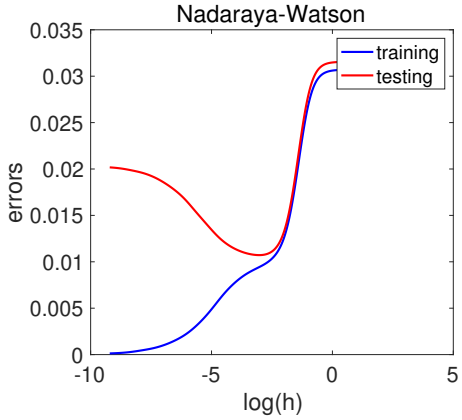
$$\begin{aligned}\mathbb{E}\Big[\sum_{i=1}^n \hat{w}_i(x)^2\Big] &= \mathbb{E}\Big[1_{\mathcal{A}(x)}\sum_{i=1}^n \hat{w}_i(x)^2\Big] + \mathbb{E}\Big[1_{\mathcal{A}(x)^c}\sum_{i=1}^n \hat{w}_i(x)^2\Big] \\ &\leqslant \mathbb{P}(\mathcal{A}(x)) + \frac{1}{p_{\min}^2 n^2 h^{2d}\|q\|_1^2/4}\mathbb{E}\Big[\sum_{i=1}^n q(\frac{1}{h}(x-x_i))^2\Big] \\ &= \mathbb{P}(\mathcal{A}(x)) + \frac{1}{p_{\min}^2 n^2 h^{2d}\|q\|_1^2/4} nh^d p_{\max}\int_{\mathbb{R}^d} q(y)^2 dy = O(1/(nh^d)).\end{aligned}$$

Bias term. We have a similar reasoning for the bias term. Indeed, using that the distance come from (any) norm, we get:

$$\begin{aligned}&\mathbb{E}\Big[\sum_{i=1}^n \hat{w}_i(x)\|x-x_i\|^2\Big] \\ =\ &\mathbb{E}\Big[1_{\mathcal{A}(x)}\sum_{i=1}^n \hat{w}_i(x) d(x_i,x)^2\Big] + \mathbb{E}\Big[1_{\mathcal{A}(x)^c}\sum_{i=1}^n \hat{w}_i(x) d(x_i,x)^2\Big] \\ \leqslant\ &\mathbb{P}(\mathcal{A}(x))\mathrm{diam}(\mathcal{X})^2 + \frac{1}{p_{\min} nh^d\|q\|_1/2} nh^d h^2 p_{\max}\int_{\mathbb{R}^d} q(y)^2\|x-y\|^2 dy = O(h^2).\end{aligned}$$

We indeed recover the same scalings as above, with explicit constants.

Experiments. For the problem shown in Section 2, below, we plot training and testing errors averaged over 10 replications, where we clearly see the trade-off in the choice of h.



3.3 k-nearest neighbor

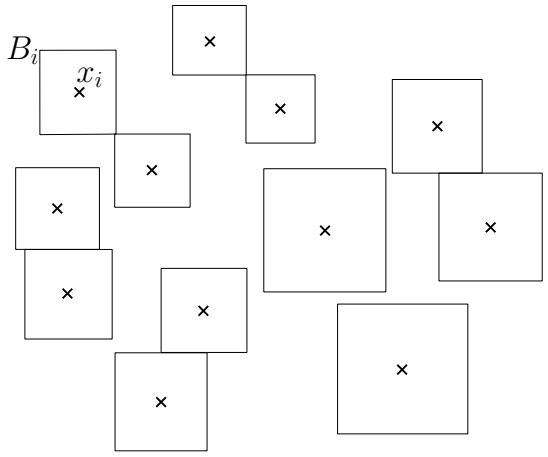
Here, by design, we have $\sum_{i=1}^n \hat{w}_i(x)^2 = \frac{1}{k}$, so the variance term will go down as soon as k tends to infinity. Moreover, for the bias term, we have that

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

is equal to the average of the squared distance between x and its k-nearest neighbors within $\{x_1, \dots, x_n\}$, and this is less than the expected distance to the k-nearest neighbors, for which the two following lemmas (taken from [2, Theorem 2.4]) give an estimate for the ℓ_∞-distance, and thus for all distances by equivalence of norms on $\mathbb{R}^d$.

Lemma 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 first-nearest-neighbor is less than $16\frac{\mathrm{diam}(\mathcal{X})^2}{n^{2/d}}$ for $d \geqslant 2$, and less than $\frac{4}{n}\mathrm{diam}(\mathcal{X})^2$ for $d = 1$.*

Proof By symmetry we aim at computing $\frac{1}{n+1}\sum_{i=1}^{n+1}\mathbb{E}\big[\|x_i - x_{(i)}\|_\infty^2\big]$, where $x_{(i)}$ is a nearest neighbor of x_i among the other n points. Denoting by $R_i = \|x_i - x_{(i)}\|_\infty$, then the sets $B_i = \{x \in \mathbb{R}^d,\ \|x - x_i\|_\infty < \frac{R_i}{2}\}$ are disjoint.



Moreover, their union has diameter less than $\mathrm{diam}(X) + \mathrm{diam}(X) = 2\mathrm{diam}(X)$. Thus by comparing volumes, we have: $\sum_{i=1}^{n+1}(R_i/2)^d \leqslant \big(2\mathrm{diam}(X)\big)^d$. Thus, by Jensen's inequality, for $d \geqslant 2$,

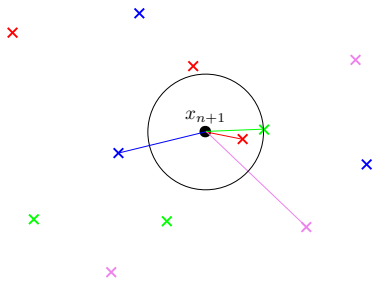
$$\Big(\frac{1}{n+1}\sum_{i=1}^{n+1} R_i^2\Big)^{d/2} \leqslant \frac{1}{n+1}\sum_{i=1}^{n+1}(R_i)^d \leqslant \frac{4^d \mathrm{diam}(X)^d}{n+1},$$

leading to the desired result. For $d = 1$, we simply have $\Big(\frac{1}{n+1}\sum_{i=1}^{n+1} R_i^2\Big) \leqslant \mathrm{diam}(\mathcal{X})\Big(\frac{1}{n+1}\sum_{i=1}^{n+1}\Big) \leqslant \frac{4}{n+1}\mathrm{diam}(\mathcal{X})^2$. ■

Lemma 2 *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 $32\mathrm{diam}(\mathcal{X})^2\Big(\frac{2k}{n}\Big)^{2/d}$ 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 bound is trivial). We can then divide randomly (and independently) the n first points into $2k$ sets of size approximately $\frac{n}{2k}$. Denoting $x_{(k)}^j$ a 1-nearest neighbor of x_{n+1} within the j-th set, the squared distance from x_{n+1} to the k-nearest

neighbors among all first n points, is less than $\frac{1}{k}\sum_{j=1}^{2k} \|x_{n+1} - x_{(k)}^j\|_\infty^2$, because among the $2k$ points $x_{(k)}^j$, at least k have to be distant from x_{n+1} more than the k-nearest-neighbor of x. See illustration below for the ℓ_2-distance, with $k = 2$ (with one color for each of the $2k = 4$ sets, and the black circle indicating the distance to the k-nearest neighbor).



Thus, using the lemma above, we get that the desired averaged distance is less than.

$$\frac{1}{k}\sum_{j=1}^{2k} 16\frac{\mathrm{diam}(\mathcal{X})^2}{(\frac{n}{2k})^{2/d}} = 32\frac{\mathrm{diam}(\mathcal{X})^2}{n^{2/d}}(2k)^{2/d}.$$

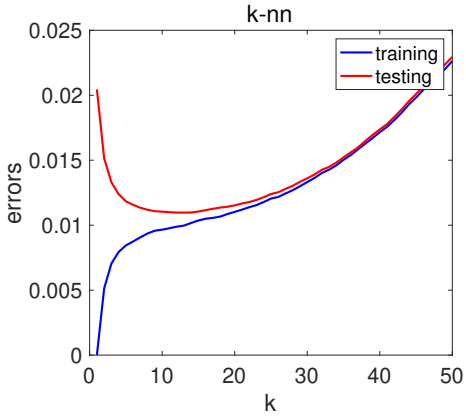
A similar argument can be extended to $d = 1$. ■

Thus the bias term is less than a constant times $B^2(k/n)^{2/d})$. Thus by balancing it with the variance term σ^2/k, with $k \propto n^{2/(2+d)}$, we obtain the same result as for the other local averaging schemes.

- **Exercise**: show that if the Bayes rate is 0, then 1-nearest-neighbor is consistent.

See more details in [1] and [2].

Experiments. For the problem shown in Section 2, below, we plot training and testing errors averaged over 10 replications, where we clearly see the trade-off in the choice of k.



4 Universal consistency (⧫)

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

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

In order 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 [6], namely that there exists $c > 0$ such that for any non-negative integrable function $h : \mathcal{X} \to \mathbb{R}$, then

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

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 expectation with respect to the training distribution.

Then for any $\varepsilon > 0$, and for any $f^* \in L_2(dp(x))$, we can find a function g which is $B(\varepsilon)$-Lipschitz-continuous and such that $\|f - g\|_{L_2(dp(x))} \leqslant \varepsilon$ (because of Lipschitz-continuous functions is dense in $L_2(dp(x))$ (see, e.g., [7]).

Then we have

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

We can now integrate with respect to x, to get

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

Thus, for example for kernel regression, given a certain ε, we can choose a certain bandwidth h, and then a certain n so that we get the squared error equal to a constant times ε. Similar arguments can be made for partitioning estimates and k-nearest-neighbors. Thus, if the extra condition in Eq. (1) is satisfied, these three methods are universally consistent.

We can now look at the three cases:

- Partitioning: We have then $c = 2$, and we get universal consistency. Indeed, we have:

$$\begin{aligned}
\sum_{i=1}^{n} \mathbb{E}\big[\hat{w}_i(x) f(x_i)\big] &= \sum_{j=1}^{J} \sum_{i=1}^{n} \mathbb{E}\big[\hat{w}_i(x) 1_{x \in A_j} f(x_i)\big] \\
&= \sum_{j=1}^{J} \mathbb{E}\Big(1_{x \in A_j}\big[1_{n_{A_j} > 0} \frac{1}{n_{A_j}} \sum_{i \in B_j} f(x_i) \; + 1_{n_{A_j} = 0} \frac{1}{n} \sum_{i=1}^{n} f(x_i)\Big) \\
&\leqslant \sum_{j=1}^{J} \mathbb{E}\Big(1_{x \in A_j}\big[\mathbb{E}[f(z) | z \in A_j] \; + 1_{x \in A_j} \frac{1}{n} \sum_{i=1}^{n} f(x_i)\Big) \\
&\leqslant 2\mathbb{E}[f(x)].
\end{aligned}$$

- 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. (1) is not easy to show, and is often referred to as Stone's lemma. See [2, Lemma 10.7].

5 Adaptivity (♦♦)

As shown above, all local averaging techniques achieve the same performance on Lipschitz-continuous functions, which is a bad unavoidable performance when d grows (curse of dimensionality). Moreover, higher smoothness of the target function does not seem to be easy to leverage.

Positive definite kernel methods will provide simple ways in Lecture 6. Among local averaging techniques, there are ways to do it. For example, using locally linear regression, where one solves for any test point x,

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

(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 Lecture 2), but the performance (both empirical and theoretical [8]) improves. See an example with the regressogram weights below.

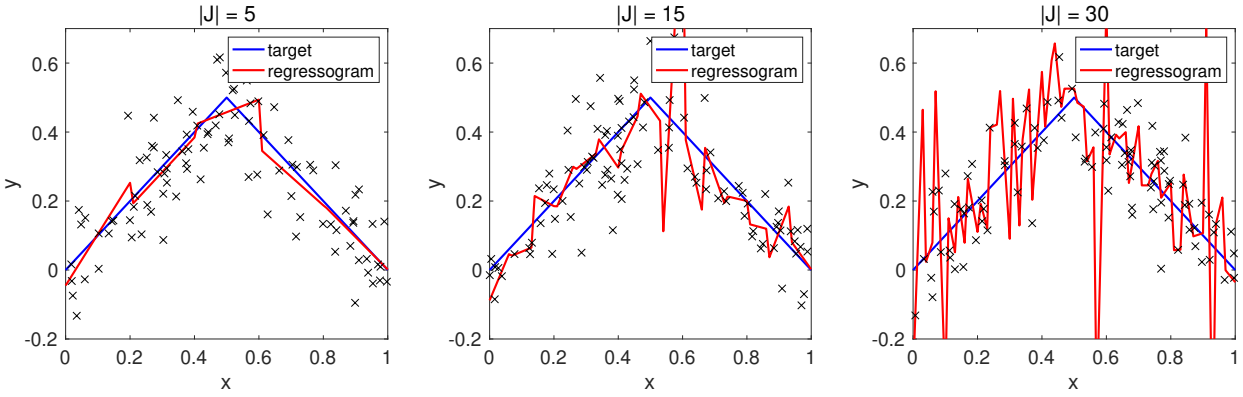


Figure 4: **FB:**

Acknowledgements

These class notes have been adapted from the notes of many colleagues I have the pleasure to work with, in particular Lénaïc Chizat, Pierre Gaillard, Alessandro Rudi and Simon Lacoste-Julien. Special thanks to Vivien Cabannes for the help on consistency proofs.

References

[1] George H. Chen and Devavrat Shah. *Explaining the Success of Nearest Neighbor Methods in Prediction.* Now Publishers, 2018.

[2] Gérard Biau and Luc Devroye. *Lectures on the Nearest Neighbor Method*, volume 246. Springer, 2015.

[3] Jean-Yves Audibert, Alexandre B Tsybakov, et al. Fast learning rates for plug-in classifiers. *The Annals of statistics*, 35(2):608–633, 2007.

[4] Kamalika Chaudhuri and Sanjoy Dasgupta. Rates of convergence for nearest neighbor classification. In *Advances in Neural Information Processing Systems*, pages 3437–3445, 2014.

[5] László Györfi, Michael Kohler, Adam Krzyzak, and Harro Walk. *A Distribution-free Theory of Nonparametric Regression*. Springer Science & Business Media, 2006.

[6] Charles J. Stone. Consistent nonparametric regression. *The Annals of Statistics*, pages 595–620, 1977.

[7] Luigi Ambrosio, Nicola Gigli, and Giuseppe Savaré. Density of Lipschitz functions and equivalence of weak gradients in metric measure spaces. *Revista Matemática Iberoamericana*, 29(3):969–996, 2013.

[8] Alexandre B. Tsybakov. *Introduction to Nonparametric Estimation*. Springer Science & Business Media, 2008.