6. Multivariate Nonparametric Estimation

- Kernel density estimation
- Nonparametric regression

Presentation based on original notes by Prof. Antonio Cuevas.

In nonparametric statistics no stringent parametric assumptions are made on the underlying probability model that generated the data.

The appeal of nonparametric methods lies in their ability to reveal structure in data that might be missed by classical parametric methods.

Nonparametric kernel smoothing methods are often, however, much more computationally demanding than their parametric counterparts.

6.1. Density estimation

Suppose that we have a sample of data points from an unknown probability density function f. *Density estimation* is the construction of an estimate f_n of the density function from the observed data.

The classical approach to estimating the probability density that generated an observed sample is to assume a parametric model. But if the assumed parametric model is not correct, inferences derived from it can lead to misleading interpretations of the data.

In *nonparametric density estimation* less rigid assumptions are placed on the functional expression of the density. The resulting density estimator is more flexible and the data are "allowed to speak for themselves". This results in a powerful tool for exploratory data analysis: visualization of data sets, classification... and a natural framework to formally define (and handle) the intuitive basic notion of mode(s).

Drawbacks:

- The usual estimators depend on a *smoothing parameter*, typically hard to select: there is no unique obvious “optimal choice” for it.
- Need for large samples: the theoretical motivation for the estimators is usually asymptotic. The estimators usually have a poor performance if the sample size is “small”.
- The *curse of dimensionality*: difficulties in higher dimensions: as the data dimension grows the estimators require larger and larger sample sizes.

Since the multivariate nonparametric density estimation methods are generalizations of univariate ones, we will introduce first the univariate proposals.

6.1.1 Univariate nonparametric density estimation

A simple density estimator: the histogram

Given a sequence $\ldots < a_i^{(n)} < a_{i+1}^{(n)} < \ldots$, with $h_n = a_{i+1}^{(n)} - a_i^{(n)}$ and given a sample $x_1, \ldots, x_n$, we define ($\#C$ being the cardinality of the set C)

$$f_n(t; x_1, \ldots, x_n) \equiv \hat{f}_n(t) = \frac{\#\{i : x_i \in (a_j^{(n)}, a_{j+1}^{(n)}]\}}{nh_n},$$

for $t \in (a_j^{(n)}, a_{j+1}^{(n)}]$, $j = 0, \pm 1, \pm 2, \ldots$.

This can be expressed with indicator functions as:

$$f_n(t) = \frac{\sum_{i=1}^n \mathbb{1}_{(a_j^{(n)}, a_{j+1}^{(n)}]}(x_i)}{nh_n},$$

Kernel estimators: definition and motivation

Another simple estimator (based on a similar idea to that of the histogram) is the *moving window estimator*

$$f_n(t) = \frac{1}{n2h_n} \sum_{i=1}^{n} \mathbb{1}_{[t-h_n,t+h_n]}(x_i) =$$

$$= \frac{1}{n2h_n} \sum_{i=1}^{n} \mathbb{1}_{[-h_n,h_n]}(t-x_i) = \frac{1}{n2h_n} \sum_{i=1}^{n} \mathbb{1}_{[-1,1]} \left(\frac{t-x_i}{h_n} \right).$$

This can be generalized by replacing the uniform density $\mathbb{1}_{[-1,1]}/2$ with another density K (called *kernel*):

$$f_n(t) = \frac{1}{nh_n} \sum_{i=1}^{n} K \left(\frac{t-x_i}{h_n} \right). \qquad (1)$$

The *kernel estimator* (1) can be seen as a smoothed version of the histogram. If K has d derivatives, so does f_n.

Possible kernel choices:

Kernel	Form
Epanechnikov	$K(x) = \frac{3}{4}(1 - x^2)\mathbb{1}_{[-1,1]}(x)$
Biweight	$K(x) = \frac{15}{16}(1 - x^2)^2\mathbb{1}_{[-1,1]}(x)$
Triweight	$K(x) = \frac{35}{32}(1 - x^2)^3\mathbb{1}_{[-1,1]}(x)$
Gaussian	$K(x) = \frac{1}{\sqrt{2\pi}}e^{-x^2/2}$
Uniform	$K(x) = \frac{1}{2}\mathbb{1}_{[-1,1]}(x)$

The kernel estimator f_n in (1) can be seen as *the convolution of the re-scaled kernel* $K_h(z) = \frac{1}{h}K\left(\frac{z}{h}\right)$ *with the empirical probability* F_n:

$$\begin{aligned} f_n(t) &= \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{t-X_i}{h}\right) = \frac{1}{n}\sum_{i=1}^{n} K_h\left(t-X_i\right) \\ &= \int_{\mathbb{R}} K_h(t-x)dF_n(x), \end{aligned}$$

Intuitively this means that the density estimator f_n is a "smoothed version" of the empirical distribution. To understand this better, assume for example that K is the Gaussian kernel. Then we can sample an observation X_i^0 from the density f_n as follows:

$$X_i^0 = x_i^* + hZ_i,\ \ i = 1, \ldots, k$$

where x_i^* is an observation randomly chosen with probability $1/n$ among the original data $x_1, \ldots, x_n$ and Z_i is a random observation drawn from $N(0,1)$ (it appears multiplied by h so hZ_i is a random observation from $N(0,h)$).

Example 6.1. The body mass index (BMI), or Quetelet index, is a measure for human body shape based on an individual's mass and height. It is defined as the individual's body mass divided by the square of the height, with the BMI value universally being given in units of kg/m^2.

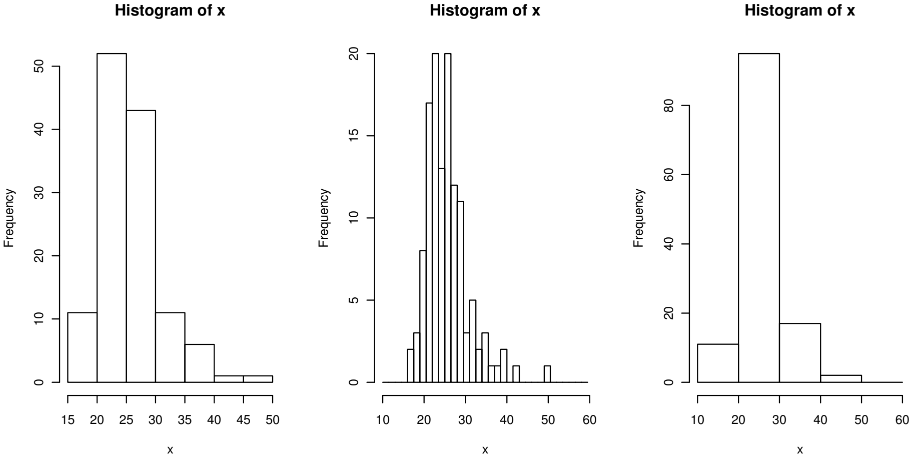
```
x = scan('Datos-IMC.txt')
hist(x)
```

The default values of the breaks a_i are obtained with

```
hist(x)$breaks
[1] 15 20 25 30 35 40 45 50
```

The histogram depends on the choice of these values:

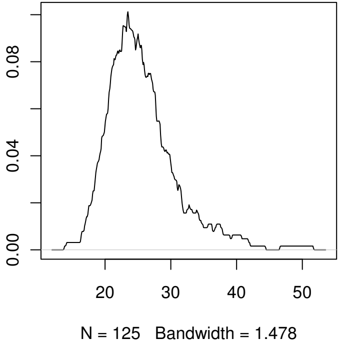
```
# First open a graph with three ‘‘pannels’’
layout(matrix(1:3,1,3)); layout.show(3)
# Now we plot three histograms
# With default bin-width:
hist(x)
# Undersmoothing (small bin-width):
hist(x,breaks=seq(10,60,1.5))
# Oversmoothing (large bin-width):
hist(x, breaks=seq(10,60,10))
```

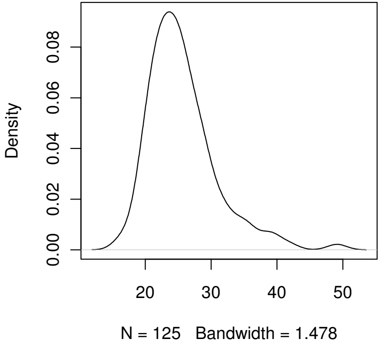

Histogram of x
Frequency
0
10
20
30
40
50
15
20
25
30
35
40
45
50
x
Histogram of x
Frequency
0
5
10
15
20
10
20
30
40
50
60
x
Histogram of x
Frequency
0
20
40
60
80
10
20
30
40
50
60
x

The Body Mass Index data (IMC) with kernel estimators...

To see the influence of changing K:

```
plot(density(x,kernel="rectangular"))
plot(density(x,kernel="gaussian"))
```

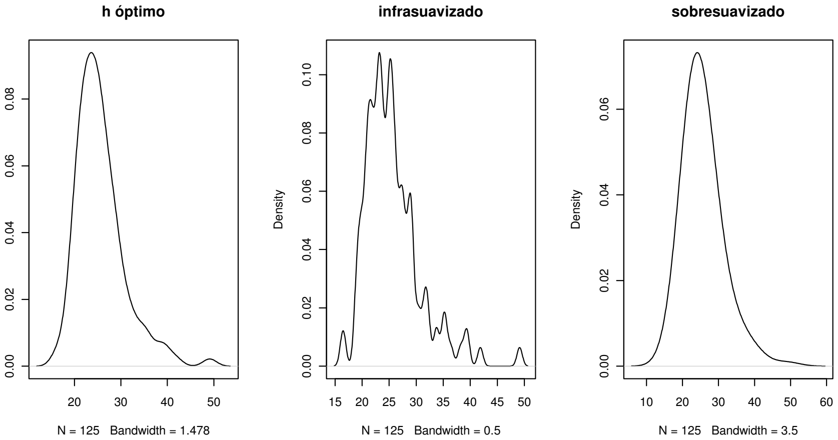




The Body Mass Index data (IMC) with kernel estimators...

To see the effect of changing *h*:

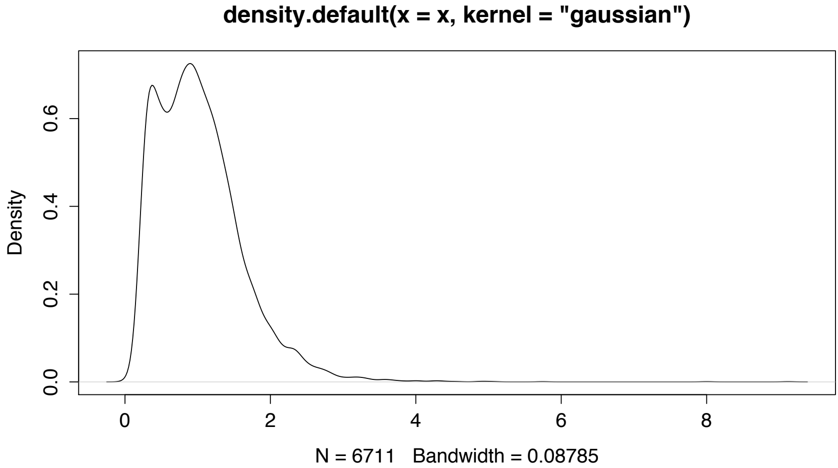
```
layout(matrix(1:3,1,3));layout.show(3)
plot(density(x,kernel="gaussian"),main="h óptimo")
plot(density(x,kernel="gaussian", bw=0.5),main="infrasuavizado")
plot(density(x,kernel="gaussian", bw=3.5),main="sobresuavizado")
```



Example 6.2. These data have been analyzed many times, from the point of view of both Economic Theory and Statistics. They correspond to the family annual income of 6711 British families in 1975. The data were rescaled dividing them by the sample mean.

```
x = scan('Datos-ingresos.txt')
plot(density(x))
```

The bandwidth h is chosen in an (approximately) "optimal way" (to be discussed further on), though the meaning of "optimal" is a rather controversial issue in this setup. In any case, this graphic suggests the **bimodal nature** of the population under study, a relevant feature which is necessarily hidden when a parametric model is used. It can be seen that this conclusion is quite robust against perturbations in the value of h.


density.default(x = x, kernel = "gaussian")
Density
0.0
0.2
0.4
0.6
0
2
4
6
8
N = 6711 Bandwidth = 0.08785

Different criteria for the estimation error

- L_2-error:

$$\begin{aligned}\text{MISE}(h) &= \mathbb{E}\|f_n - f\|_2^2 = \mathbb{E}\int (f_n(t) - f(t))^2 dt \\ &= \int (\mathbb{E}(f_n(t)) - f(t))^2\, dt + \int \mathbb{V}(f_n(t))dt \\ &= \text{Bias (square) term} + \text{Variance term.}\end{aligned}$$

  The decomposition bias+variance is common in nonparametrics. Usually there is a sort of “trade-off” (e.g., in the choice of h): when the variance increases the bias decreases and vice versa.

- L_1-error: $J_n = J_n(h) = \mathbb{E}\|f_n - f\|_1 = \mathbb{E}\int |f_n(t) - f(t)|dt$
- L_∞-error: $\mathbb{E}\|f_n - f\|_\infty$

  The most popular criterion is, by far, the L_2-error. This is mostly due to its mathematical tractability. There are also strong reasons in favor of using the other two criteria (see, e.g. Devroye and Gÿorfi 1985, Nadaraya 1989).

On the L_2-error

Theorem 6.1: *Let f_n be a sequence of kernel density estimators, based on a sample $X_1, \ldots, X_n$ of iid observations drawn from f:*

$$f_n(t) = \frac{1}{n} \sum_{i=1}^{n} K_h(t - X_i),$$

where $K_h(z) := h^{-1} K(z/h)$, the smoothing parameter satisfies

$$h := h_n \to 0, \tag{2}$$

and the kernel satisfies

$$K \in L_2 \text{ is a symmetric density with } m_4(K) = \int u^4 K(u) du < \infty. \tag{3}$$

Assume further that

$$f \text{ has four continuous derivatives and } f,\ f^{''},\ f^{(4)} \in L_2(\mathbb{R}). \tag{4}$$

Then,

$$MISE(h) = \frac{h^4}{4} m_2(K)^2 R(f^{''}) + o(h^4) + \frac{R(K)}{nh} - \frac{1}{n} R(f) + O(n^{-1} h^4), \tag{5}$$

where $R(g) = \int g^2$ and $m_2(K) = \int u^2 K(u) du$.

Remarks and consequences from the expression (5) of the L_2 error:

- ⋆ The optimal asymptotic error: The first term in (5) comes from the expansion of the bias term, the second one comes from the variance. The remaining terms are asymptotically negligible. So, the *asymptotic mean square error* is given by

$$\mathrm{AMISE}(h) = \frac{h^4}{4} m_2(K)^2 R(f^{''}) + \frac{R(K)}{nh}. \tag{6}$$

  Under the usual assumption $nh_n \to \infty$, we get L_2-consistency: $MISE(h) \to 0$.

  Formula (6) can be found in many places. However, the precise statement of the result (with ALL the required assumptions) is not easy to find. It can be found, e.g., in Chacón (2004).

- ⋆ Bias and variance: The variance term in AMISE does not depend on f. The bias term depends on f through the curvature $R(f^{''})$. The larger $R(f^{''})$, the harder is the estimation task.

⋆ The optimal choice of h: Minimization of (6) yields this expression for h:

$$h_0 = \frac{R(K)^{1/5}}{m_2(K)^{2/5} R(f'')^{1/5}} n^{-1/5} \tag{7}$$

Thus, every possible estimate of the curvature $R(f'')$ provides a selection for h. This is the plug-in methodology for "automatic" selection of the smoothing parameter. It can be dealt with in many different ways:

- The "rule of thumb": to estimate $R(f'')$ as if f were a normal distribution with the variance estimated from the data.
- Nonparametric plug-in: to estimate $R(f'')$ by $R(\hat{f}_n'')$ where $\hat{f}_n''$ ia another density estimator based on a new "pilot" bandwidth. A popular choice is the so-called Sheather and Jones (1991) bandwidth selector (given by `bw.SJ(x)` in R).

- The optimal order of convergence:

$$AMISE_0 = AMISE(h_0) = \frac{5}{4} m_2(K)^{2/5} R(K)^{4/5} R(f'')^{1/5} n^{-4/5} \quad (8)$$

- The optimal kernel: The minimization in K of AMISE_0 (subject to $m_2(K) = 1$) leads to the so-called Epanechnikov kernel.

6.1.2 Multivariate nonparametric density estimation

In the multivariate framework, where $\mathbf{X}$ is a random vector in $\mathbb{R}^p$ with density f, the general form of the kernel density estimator based on a sample $\mathbf{X}_1, \ldots, \mathbf{X}_n$ of $\mathbf{X}$ is (Wand and Jones 1995)

$$f_n(\mathbf{x}; \mathbf{H}) = \frac{1}{n} \sum_{i=1}^{n} K_{\mathbf{H}}(\mathbf{x} - \mathbf{X}_i), \tag{9}$$

where $K_{\mathbf{H}}(\mathbf{x}) := |\mathbf{H}|^{-1/2} K(\mathbf{H}^{-1/2}\mathbf{x})$, the $p \times p$ matrix of smoothing parameters $\mathbf{H}$ is symmetric and positive definite, and $K : \mathbb{R}^p \to [0, \infty)$ is a probability density.

The choice of the bandwidth matrix $\mathbf{H}$ is the most important factor affecting the accuracy of the estimator (9), since it controls the orientation and amount of smoothing induced.

The simplest choice is $\mathbf{H} = h^2 \mathbf{I}_p$, where h is a unidimensional smoothing parameter and $\mathbf{I}_p$ is the $p \times p$ identity matrix. Then we have the same amount of smoothing applied in all coordinate directions and the kernel estimator (9) has the form

$$f_n(\mathbf{x}) = \frac{1}{nh^d} \sum_{i=1}^{n} K\left(\frac{\mathbf{x} - \mathbf{X}_i}{h}\right) = \frac{1}{n} \sum_{i=1}^{n} K_h\left(\mathbf{x} - \mathbf{X}_i\right)$$

where $K_h(\mathbf{x}) := h^{-p} K(\mathbf{x}/h)$.

Another, relatively easy-to-manage, choice is to take the bandwidth matrix equal to a diagonal matrix, which allows for different amounts of smoothing in each of the coordinates.

Two particular choices of the kernel function K:

- Product kernel: $K(\mathbf{x}) = K(x_1, \ldots, x_p) = \prod_{j=1}^{p} k(x_j)$
- Symmetric kernel: $K(\mathbf{x}) = c_{k,p}\, k(\|\mathbf{x}\|_2^{1/2})$. For example, the Gaussian kernel $K(\mathbf{x}) = (2\pi)^{-p/2} \exp(-\mathbf{x}'\mathbf{x}/2)$.

The most commonly used optimality criterion for selecting the bandwidth matrix $\mathbf{H}$ is the mean integrated squared error

$$\mathrm{MISE}(\mathbf{H}) = \mathbb{E}\|f_n - f\|_2^2 = \mathbb{E}\int (f_n(\mathbf{x}) - f(\mathbf{x}))^2 d\mathbf{x}.$$

Its asymptotic approximation is

$$\mathrm{AMISE}(\mathbf{H}) = n^{-1}|\mathbf{H}|^{-1/2}R(K) + \frac{1}{4}m_2(K)^2(\mathrm{vec}\ \mathbf{H})'\mathbf{\Psi}_4(\mathrm{vec}\ \mathbf{H}),$$

where $R(K) := \int K^2(\mathbf{x})d\mathbf{x}$, $\int \mathbf{x}\mathbf{x}'K^2(\mathbf{x})d\mathbf{x} = m_2(K)\,\mathbf{I}_p$, $D^2 f$ is the $p \times p$ Hessian matrix of second order partial derivatives of f, $\mathbf{\Psi}_4 := \int (\mathrm{vec}\ D^2 f(\mathbf{x}))(\mathrm{vec}\ D^2 f(\mathbf{x}))d\mathbf{x}$ is a $p^2 \times p^2$ matrix of integrated fourth-order partial derivatives of f and vec is the vector operator converting a matrix into a column vector.

See Wand and Jones (1994) and Duong and Hazelton (2005).

For the particular choice $\mathbf{H} = h^2 \mathbf{I}_p$ we obtain

$$\text{AMISE}(h) = \frac{R(K)}{nh^p} + \frac{h^4}{4} \int (\text{tr} D^2 f(\mathbf{x}))^2 d\mathbf{x}.$$

Then the optimal h is $O(n^{-1/(p+4)})$ yielding an AMISE $O(n^{-4/(p+4)})$. Thus, the convergence to 0 of the error is MUCH SLOWER than in the one-dimensional case. This is the so-called "curse of dimensionality". Therefore, multidimensional density estimation is usually not applied for large dimensions ($p \geq 5$).

If $\mathbf{H} = \text{diag}(h_1, \ldots, h_p)$ and $\mathbf{\Sigma} = \text{diag}(\sigma_1^2, \ldots, \sigma_p^2)$, then the bandwidth minimizing AMISE is given by

$$\boxed{h_j = \left(\frac{4}{p+2}\right)^{1/(p+4)} n^{-1/(p+4)} \sigma_j.}$$

Replacing the σ_j's with estimates, $\hat{\sigma}_j$, and noting that the first factor is always near 1, we arrive at Scott's (1992) rule: $h_j = n^{-1/(p+4)} \hat{\sigma}_j$.

On the curse of dimensionality

In $B(\mathbf{0},1)$ the outer crown $B(\mathbf{0},1) \setminus B(\mathbf{0},0.99)$ is almost 19 % of the volume in dimension 20 but only 2 % in dimension 2. On the contrary, $B(0,1/2)$ is 25 % of the volume in dimension 2 but just 10^{-6} times the total volume of the ball in dimension 20.

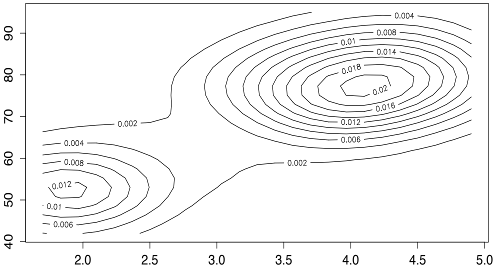
In probability terms: in a standard one-dimensional normal 90 % of the data are in the interval $[-1.6, 1.6]$; in dimension 10, 99 % of the data correspond to points whose distance from the origin is larger than 1.6. This is the **phenomenon of the empty spaces**.

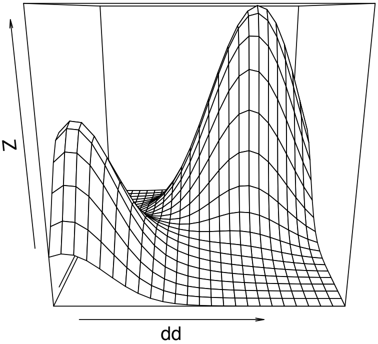
Thus, in order to estimate $f(0)$ (where f is the standard normal) with a kernel estimator with Gaussian kernel, the values of the smallest value n ensuring $\text{ínf}_h \, E(f_n(0) - f(0))^2/f(0)^2 \leq 0.1$ in dimensions 1,2,3,4,5,...10 are, respectively, 4, 19, 67, 223, 768, 842000.

There exist a number of packages that can perform multivariate kernel density estimation in R, for example, `KernSmooth` of Wand and Ripley, `sm` of Bowman and Azzalini, and `feature` of Duong and Matt.

Example: Times between Old Faithful eruptions (Y) and duration of eruptions (X).

```
Datos = read.table('Datos-geyser.txt',header=TRUE)
library(MASS)
dd =  kde2d(Datos$X,Datos$Y) # estimate the bivariate
  kernel estimator
contour(dd) # plot level sets of the density
 estimator
persp(dd) # plot density estimator in 3d
```


0.002
0.004
0.006
0.008
0.01
0.012
0.014
0.016
0.018
0.02
0.002
0.004
0.006
0.008
0.01
0.012
2.0
2.5
3.0
3.5
4.0
4.5
5.0
40
50
60
70
80
90


Z
dd

6.1.3 More on the choice of the smoothing parameter

Given a sample $X_1, \ldots, X_n$, an alternative way to select h, would be to define an appropriate estimator of MISE(h) (depending on h and the sample) and to select the value of h minimizing this value. A popular procedure (often applied in slightly modified versions) is the so-called *least squares cross-validation*. Note that

$$\mathrm{MISE}(h) = \mathbb{E} \int (f_n(t;h) - f(t))^2 \, dt = \mathbb{E} \left(\int f_n^2 - 2 \int f_n f + \int f^2 \right).$$

As the last term does not depend on h, a natural idea would be to minimize (in h)

$$Q(h) = \int f_n^2 - 2 \int f_n f.$$

But $Q(h)$ still depends on f. So we replace $\int f_n f$ in $Q(h)$ by the leave-one-out approximation $\frac{1}{n} \sum_{i=1}^n f_{ni}$, where f_{ni} is the estimator $f_{n-1}(X_i; h)$ based on the sample $X_1, \ldots, X_{i-1}, X_{i+1}, \ldots, X_n$ and evaluated at $t = X_i$.

This is a reasonable estimator, as it can be easily seen that

$$\mathbb{E}\left(\frac{1}{n}\sum_{i=1}^{n} f_{ni}\right) = \mathbb{E}\left(\int f_n f\right).$$

Thus, we are approximating $Q(h)$ by

$$Q_0(h) = \int f_n^2 - \frac{2}{n}\sum_{i=1}^{n} f_{ni}.$$

Now, the *least squares cross-validation bandwidth* (see Rudemo 1982, Bowman 1984, Stone 1984) is defined by

$$h_{\mathrm{LS}} = \arg\min Q_0(h).$$

It can be proved (see Silverman 1986, p. 50) that

$$Q_0(h) = \frac{1}{n^2 h}\sum_i\sum_j K^{(2)}\left(\frac{X_i - X_j}{h}\right) - \frac{2}{n(n-1)}\sum_i\sum_j K_h(X_i - X_j) + \frac{2}{(n-1)h}K(0)$$

where $K^{(2)} := K * K$. Thus if, for example K is the standard Gaussian kernel, $K^{(2)}$ would be the density $N(0, \sqrt{2})$.

It was proved by Stone (1984) that, if f is bounded, then $f_n(\cdot; h_{\mathrm{LS}})$ is consistent in L^2.

The h_{LS} bandwidth (as well as several modifications of it) is still very popular in spite of several shortcomings:

- It is very sensitive to repeated observations.
- The convergence to the optimum is very slow. Typically, if h_0 denotes the optimal h,

$$\frac{h_{LS} - h_0}{h_0} = O_P(n^{-1/10}). \tag{10}$$

The cross-validation approach has a straightforward generalization to the multivariate setting (see Scott 1992, Duong and Hazelton 2005). The difficulty comes from the fact that the bandwidth is now a $p \times p$ matrix $\mathbf{H}$. In the most general case this means to minimize over $p(p+1)/2$ parameters. If we assume $\mathbf{H}$ to be a diagonal matrix, this still remains a p-dimensional optimization problem.

Typing ?bw.nrd in R we obtain several choices for the bandwidth in the 1-dimensional setting:

Example 6.2:

```
x = scan(’Datos-ingresos.txt’)
bw.ucv(x) # cross-validation bandwidth
[1] 0.02982046
bw.SJ(x) # Sheather and Jones bandwidth
[1] 0.04097176
```

The R package ks by Duong computes some versions of the cross-validation and plug-in multivariate bandwidth matrices for dimension $p \leq 6$.

Example: Times between Old Faithful eruptions (Y) and duration of eruptions (X).

```
Hlscv(XY) # cross-validation bandwidth matrix
```

```
             [,1]         [,2]
[1,] 6.415263e-10 7.685498e-06
[2,] 7.685498e-06 1.059208e+02

Warning message:
In Hlscv(XY) :
  Data contain duplicated values: LSCV is not well-behaved
   in this case
```

```
Hpi(XY) # Plug-in bandwidth matrix
```

```
           [,1]       [,2]
[1,] 0.09416555  0.8731318
[2,] 0.87313184 18.4161607
```

6.2. Nonparametric regression ("learning a function")

6.2.1 The regression problem

Broadly, the regression problem consists in modeling a *response* random variable $Y \in \mathbb{R}$ as a function m of a *predictor* random vector (or *covariate* or *feature*) $\mathbf{X} \in \mathbb{R}^p$.

The aim is to find the "best" approximation of Y in terms of $\mathbf{X}$. Here "best" refers to optimizing with respect to the L_2 norm. Assume $\mathbb{E}(Y^2) < \infty$. The *regression function* m is defined as

$$m := \arg\min_{g \in \mathcal{M}} \mathbb{E}(Y - g(\mathbf{X}))^2, \qquad (11)$$

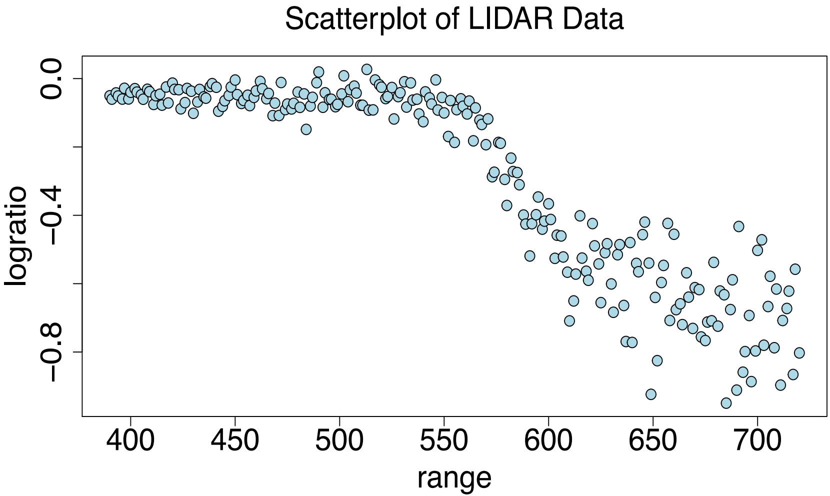
where $\mathcal{M}$ denotes the class of real measurable functions g such that $\mathbb{E}(g^2(X)) < \infty$.
It can be easily shown that the solution to (11) is given by the *conditional expectation*,

$$m(\mathbf{x}) = \mathbb{E}(Y | \mathbf{X} = \mathbf{x}).$$

Example 6.3. Ruppert *et al.* (2003) describe data from a LIDAR experiment in Sigrist (1994). LIDAR is a remote sensing technology that measures distance by illuminating a target with a laser and analyzing the reflected light.

The file `lidar.dat` contains 221 observations of the following variables: the predictor, `range`, is the distance travelled before the light is reflected back to its source; the response, `logratio`, is the logarithm of the ratio of light received from two laser sources. The frequency of one laser is the resonance frequency of mercury while the second has a different frequency.

```
Datos = read.table('lidar.dat',header=TRUE)
par(mar = c(5, 4.5, 4, 2) + 0.1)
plot(Datos, cex = 1.5, pch = 21,col='black', bg='lightblue',
     xlab='range',ylab='logratio',cex.lab=1.5,cex.axis=1.5,
     main='Scatterplot of LIDAR Data',font.main=1,cex.main=1.5)
```


Scatterplot of LIDAR Data
logratio
0.0
−0.4
−0.8
range
400
450
500
550
600
650
700

6.2.2 Linear regression

The classical approach in regression is the *linear model*, where the regression function m is assumed to be a linear function of the regression coefficients. Specifically, the linear regression model is

$$m(\mathbf{x}) = \mathbf{\Phi}(\mathbf{x})'\boldsymbol{\beta} = \beta_1\phi_1(\mathbf{x}) + \beta_2\phi_2(\mathbf{x}) + \ldots + \beta_q\phi_q(\mathbf{x}),$$

where $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_q)'$ and $\mathbf{\Phi} = (\phi_1, \ldots \phi_q)' : \mathbb{R}^p \to \mathbb{R}^q$ is a known (possibly non-linear) transformation of the predictor $\mathbf{x}$.

In particular, the simplest linear regression model assumes m to be an affine function of $\mathbf{x}$:

$$m(\mathbf{x}) = \beta_1 x_1 + \ldots + \beta_p x_p + \beta_{p+1} = (\mathbf{x}', 1)\,\boldsymbol{\beta}, \qquad (12)$$

where $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p, \beta_{p+1})'$.

Let $(\mathbf{x}_1', y_1), \ldots, (\mathbf{x}_n', y_n)$ be a sample from $(\mathbf{X}', Y)$, with $\mathbf{x}_i' = (x_{i1}, \ldots, x_{in})$, $i = 1, \ldots, n$.

In the linear regression model (12) the least squares estimator of $\boldsymbol{\beta}$ is given by $\hat{\boldsymbol{\beta}} = (\mathbb{X}'\mathbb{X})^{-1}\mathbb{X}'\mathbb{Y}$, where

$$\mathbb{X} = \begin{pmatrix} x_{11} & \ldots & x_{1p} & 1 \\ \vdots & & \vdots & \vdots \\ x_{n1} & \ldots & x_{np} & 1 \end{pmatrix}$$

is the *design matrix* and $\mathbb{Y} = (y_1, \ldots, y_n)'$.

The resulting estimator of the linear regression function is then

$$m_n(\mathbf{x}) = (\mathbf{x}', 1)\,\hat{\boldsymbol{\beta}} = (\mathbf{x}', 1)(\mathbb{X}'\mathbb{X})^{-1}\mathbb{X}'\mathbb{Y} = \sum_{i=1}^{n} W_{in}(\mathbf{x})Y_i.$$

6.2.3 Nonparametric regression

In the nonparametric framework we remove any parametric restrictions on m. This helps bring to light the underlying structure of the regression data. Our aim is to estimate or "learn" the function m under weak assumptions. Nonparametric estimators of m are sometimes called *smoothers*.

An estimator m_n of m is a *linear smoother* if, for each $\mathbf{x}$, there exists a vector of weights $w_n(\mathbf{x}) = (W_{1n}(\mathbf{x}), \ldots, W_{nn}(\mathbf{x}))'$ such that

$$m_n(\mathbf{x}) = w_n(\mathbf{x})'\mathbb{Y} = \sum_{i=1}^{n} W_{in}(\mathbf{x}) Y_i. \tag{13}$$

Observe that a linear smoother is not necessarily a linear regression function. In fact, most usual nonparametric estimators are linear smoothers, with weights $W_{in}(\mathbf{x})$ depending on the proportion of sample points falling in a close neighbourhood of $\mathbf{x}$. Then they are called *local averaging regression estimators*.

Usually the weights $W_{in}(\mathbf{x})$ are nonnegative and $W_{in}(\mathbf{x})$ is "small" if $\mathbf{X}_i$ is "far" from $\mathbf{x}$. The weights in the smoothers we consider fulfill these two requirements and also $\sum_{i=1}^n W_{in}(\mathbf{x}) = 1$ for all $\mathbf{x}$.

The next result, known as Stone's Theorem, states conditions on the weights which guarantee the weak universal consistency of the local averaging estimates.

Theorem 6.2 (Stone 1977): *Assume that the following conditions are satisfied for any distribution of* $\mathbf{X}$*:*

(i) *There is a constant c such that for every nonnegative measurable function f satisfying $\mathbb{E}f(\mathbf{X}) < \infty$ and any n,*

$$\mathbb{E}\left(\sum_{i=1}^n |W_{in}(\mathbf{X})|\, f(\mathbf{X}_i)\right) \leq c\, \mathbb{E}f(\mathbf{X}).$$

(ii) *There is a $D \geq 1$ such that, for all n,*

$$\mathbb{P}\left\{\sum_{i=1}^n |W_{in}(\mathbf{X})| \leq D\right\} = 1.$$

(iii) *For all $a > 0$*

$$\lim_{n\to\infty} \mathbb{E}\left(\sum_{i=1}^{n} |W_{in}(\mathbf{X})| \, \mathbb{1}_{\{\|\mathbf{X}_i - \mathbf{X}\| > a\}}\right) = 0.$$

(iv)

$$\sum_{i=1}^{n} |W_{in}(\mathbf{X})| \xrightarrow[n\to\infty]{\mathbb{P}} 1.$$

(v)

$$\lim_{n\to\infty} \mathbb{E}\left(\sum_{i=1}^{n} W_{in}^2(\mathbf{X})\right) = 0.$$

Then the regression estimator given by (13) is weakly universally consistent, that is, for all distributions of $(\mathbf{X}, Y)$ with $\mathbb{E}Y^2 < \infty$,

$$\lim_{n\to\infty} \mathbb{E}\left(\int (m_n(\mathbf{x}) - m(\mathbf{x}))^2 d\mathbf{x}\right) = 0.$$

Kernel regression estimator

The multivariate kernel density estimation techniques provide a nonparametric estimator of m. Let $f_{\mathbf{X}}(\mathbf{x})$, $f(\mathbf{x}, y)$ and $f(y|\mathbf{x})$ be the marginal density of $\mathbf{X}$, the joint density of $\mathbf{X}$ and Y, and the conditional density of Y given $\mathbf{X}$ respectively. Then

$$m(\mathbf{x}) = \mathbb{E}(Y|\mathbf{X} = \mathbf{x}) = \int y\, f(y|\mathbf{x})\, dy = \int y\, \frac{f(\mathbf{x}, y)}{f_{\mathbf{X}}(\mathbf{x})}\, dy, \qquad (14)$$

A product kernel density estimate of $f(\mathbf{x}, y)$ is

$$\begin{aligned} \hat{f}(\mathbf{x}, y) &:= \frac{1}{nh_x^p h_y} \sum_{i=1}^{n} K_x\left(\frac{\mathbf{x} - \mathbf{X}_i}{h_x}\right) K_y\left(\frac{y - Y_i}{h_y}\right) \\ &= \frac{1}{n} \sum_{i=1}^{n} K_{x;h_x}(\mathbf{x} - \mathbf{X}_i) K_{y;h_y}(y - Y_i), \end{aligned}$$

where K_x and K_y are probability densities on $\mathbb{R}^p$ and $\mathbb{R}$ respectively with $\int yK_y(y)dy = 0$, $K_{x;h} := h^{-p}K(\cdot/h)$ and $K_{y;h} := h^{-1}K(\cdot/h)$.

A kernel estimate of $f_{\mathbf{X}}(\mathbf{x})$ is

$$\hat{f}_{\mathbf{X}}(\mathbf{x}) := \frac{1}{nh_x^p} \sum_{i=1}^{n} K_x \left(\frac{\mathbf{x} - \mathbf{X}_i}{h_x} \right) = \frac{1}{n} \sum_{i=1}^{n} K_{x;h_x}(\mathbf{x} - \mathbf{X}_i).$$

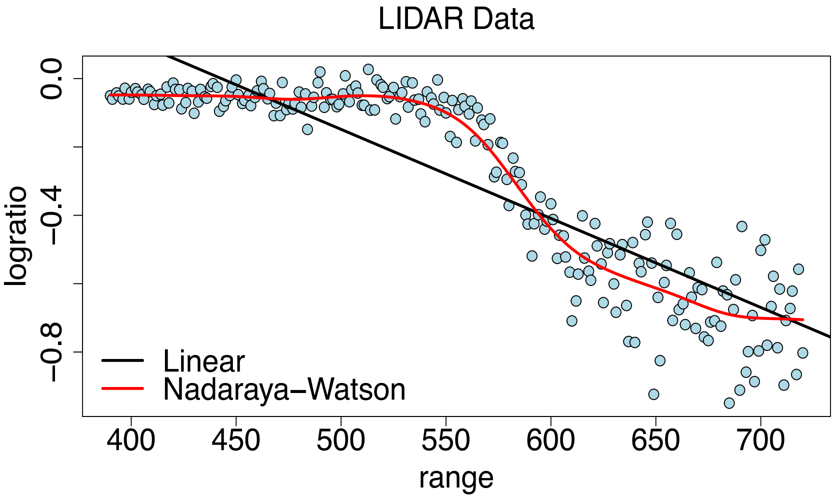
Substituting the unknown densities in (14) by these estimates yields the *Nadaraya-Watson kernel estimator* of the regression function m (Nadaraya 1964 and Watson 1964):

$$m_{\mathrm{NW}}(\mathbf{x}) = \frac{\sum_{i=1}^{n} Y_i \, K_{x;h_x}(\mathbf{x} - \mathbf{X}_i)}{\sum_{i=1}^{n} K_{x;h_x}(\mathbf{x} - \mathbf{X}_i)}. \tag{15}$$

Observe that m_{NW} is a linear smoother.

Example 6.3 (LIDAR): In R the function `npreg` in the `np` package computes the NadarayaWatson kernel regression estimate.

```
Datos = read.table(’lidar.dat’,header=TRUE)
Y = Datos$logratio
X = Datos$range

par(mar = c(5, 4.5, 4, 2) + 0.1)
plot(Datos, cex = 1.5, pch = 21,col=’black’, bg=’lightblue’,
     xlab=’range’,ylab=’logratio’,cex.lab=1.5,cex.axis=1.5,
     main=’LIDAR Data’,font.main=1,cex.main=1.5)

lr  = lm (Y ˜ X) # Linear regression
abline(lr, lwd=3) # Plot the linear regression on the existing
   scatterplot

#library(np)
nw  = npreg(Y ˜ X) # Nadaraya-Watson (NW) regression
lines(X, fitted(nw), lwd=3, col="red") # Plot the NW regression on the
   existing scatterplot

legend("bottomleft",c(’Linear’,’Nadaraya-Watson’),col=c(’black’,’red’),
       cex=1.5, bty="n",lty=c(1,1),lwd=c(3,3),text.font=1)
```


LIDAR Data
logratio
0.0
−0.4
−0.8
400
450
500
550
600
650
700
range
Linear
Nadaraya−Watson

Theorem 6.3 (Universal consistency of kernel smoother): *Assume that the kernel K is bounded and bounded away from zero in a neighborhood of 0. If $\mathbb{E}|Y|^2 < \infty$ and $h_n \to 0$ with $nh_n^p \to \infty$, then the kernel estimator fulfills*

$$\mathbb{E}\|m_{NW} - m\|_2^2 \to 0 \quad \textit{as } n \to \infty.$$

The proof is based on Stone's Theorem (Theorem 6.2) (see Györfi *et al.* 2002, p. 71).

As in kernel density estimation, kernel regression involves choosing the kernel function and the bandwidth parameter. Here one observes the same phenomenon as in density estimation: the difference between two kernel functions K is almost negligible. Consequently, the important task is bandwidth selection. The choice of the smoothing parameter h is made so that some error criterion is minimized.

There are several criteria measuring in one way or another how close the estimate m_n is to the true curve. For the Nadaraya Watson estimator (15) it has been shown (Marron and Härdle 1986) that ASE (see (17)), ISE and MISE (see (16)) lead asymptotically to the same level of smoothing. Hence, we can use the easiest criterion to calculate and manipulate.

A natural choice would be the *mean integrated squared error*,

$$\mathrm{MISE} = \mathbb{E} \int_{-\infty}^{\infty} (m_n(\mathbf{x}) - m(\mathbf{x}))^2 f_{\mathbf{X}}(\mathbf{x})\, d\mathbf{x}. \tag{16}$$

The weighting by $f_{\mathbf{X}}$ puts more emphasis in regions with more data and assigns less weight to observations in regions of sparse data (to reduce the variance in this region) or at the tail of the distribution of $\mathbf{X}$ (to trim away boundary effects).

We could think of optimizing its asymptotic version, AMISE, as in density estimation. However, the AMISE in regression involves more unknown quantities than the AMISE for density estimation.

For example, for $p = 1$, the *mean squared error* at x has the following asymptotic expansion:

$$\begin{aligned} \text{MSE}(x;h) &= \mathbb{E}(m_n(x) - m(x))^2 \\ &= \frac{1}{nh}\|K\|_2^2 \frac{\sigma^2(x)}{f_X(x)} \\ &\quad + \frac{1}{4}h^4 m_2^2(K)\left(m''(x) + \overbrace{2m'(x)\frac{f_X'(x)}{f_X(x)}}^{\text{Design bias}}\right)^2 \end{aligned}$$

Thus, the plug-in procedure in kernel regression has disadvantages with respect to other methods, such as cross-validation or penalization (see, e.g., Härdle 1990, Härdle *et al.* 2004). The cross-validation approach aims at minimizing the *averaged squared error*

$$\text{ASE}(h) = \frac{1}{n}\sum_{i=1}^{n}(m_n(X_i) - m(X_i))^2 w(X_j), \tag{17}$$

where w is a weight function.

k-NN (nearest neighbours) regression estimator

$$m_{\text{NN}}(x) = \frac{\sum_{i=1}^{n} Y_i \mathbb{1}_{\{\mathbf{X}_i \in k(\mathbf{x})\}}}{k_n},$$

where "$\mathbf{X}_i \in k(\mathbf{x})$" means that $\mathbf{X}_i$ is one of the k nearest neighbours of $\mathbf{x}$ among the sample points $\mathbf{X}_1, \ldots, \mathbf{X}_n$.

Theorem 6.3 (Universal consistency): *If $\mathbb{E}|Y|^p < \infty$ and $k_n \to \infty$ with $k_n/n \to 0$, then the k_n-NN estimator fulfills*

$$\mathbb{E}\|m_{NN} - m\|_p^p \to 0.$$

For the proof see, e.g., Györfi *et al.* (2002), p. 88.

Local polynomial regression

Kernel regression estimators suffer from boundary bias and design bias. These problems can be alleviated by using a generalization of kernel regression called local polynomial regression.

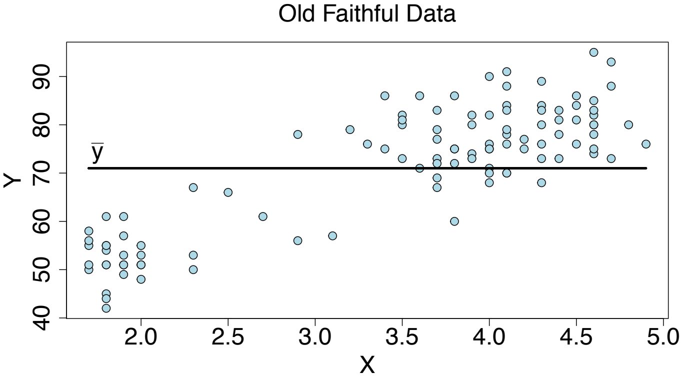
To motivate the local polynomial estimator, let us begin with the one-dimensional case $p = 1$. Observe that, since the unknown regression function m minimizes $\mathbb{E}(Y - m(X))^2$, it is natural to look for a regression estimator m_n minimizing a sample version of the L^2 error. First consider choosing a regression estimator $m_n(x) = a$ minimizing the ordinary sum of squares

$$\mathrm{OSS} = \sum_{i=1}^{n} (Y_i - a)^2.$$

The solution is a constant function $m_n(x) = \bar{Y} = \sum_{i=1}^{n} Y_i/n$ for all x, which, in general, is not a good estimator of $m(x)$.

Example (Old Faithful Data):

```
Datos = read.table('Datos-geyser.txt',header=TRUE)
XY = cbind(Datos$X,Datos$Y)
par(mar = c(5, 4.5, 4, 2) + 0.1)
plot(XY,cex=1.5,pch=21,col='black',bg='lightblue',xlab='X',ylab='Y',cex.lab=1.5,
    cex.axis=1.5,main='Old Faithful Data',font.main=1,cex.main=1.5)
X = Datos$X
Y = Datos$Y
n = length(Y)
mY = mean(Y)
lines(X, mY*rep(1,n), lwd=3, col="black")
text(1.75,74,expression(bar(y)),cex=1.5)
```



Now choose $m_n(x) = a$ to minimize the *weighted sum of squares*

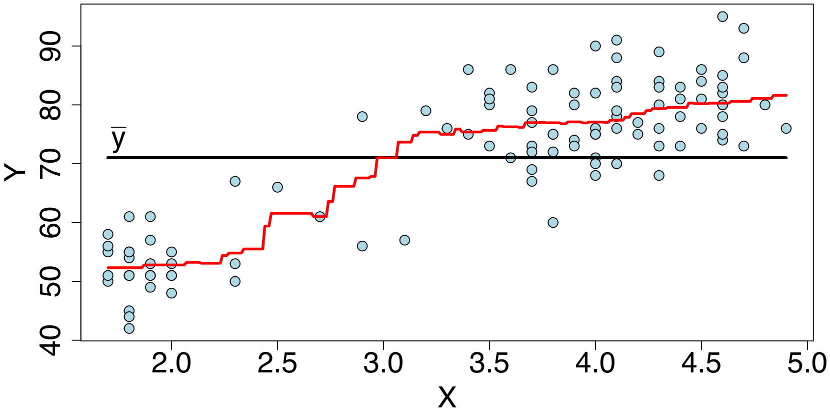
$$\mathrm{WSS} = \sum_{i=1}^{n} (Y_i - a)^2 K\left(\frac{X_i - x}{h}\right).$$

The solution is the Nadaraya-Watson estimator (15). Thus, this kernel regression estimator is a locally constant regression estimator, weighting more heavily values of Y corresponding to X_i's closer to x.

Example (Old Faithful Data): Kernel regression with uniform kernel

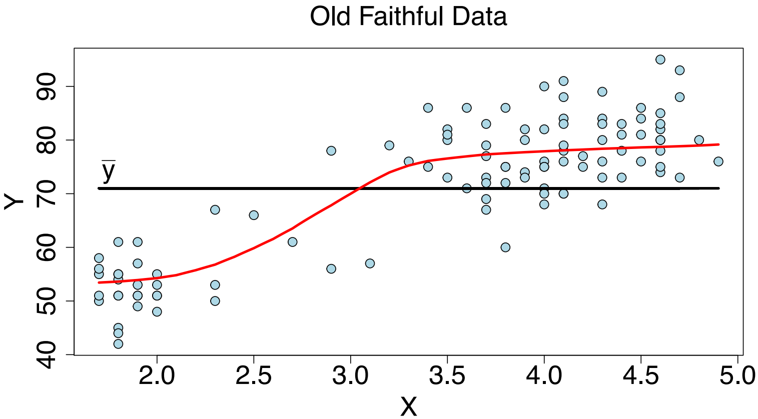
```
# library(np)
nwU  = npreg(Y ~ X,ckertype="uniform") # Nadaraya-Watson regression with uniform kernel
newX = data.frame(seq(min(X),max(X),0.01))
predYnewX = predict(nwU,exdat=newX)
lines(t(newX), predYnewX, lwd=3, col="red") # Plot the NW regression on the existing scatterplot
```

Old Faithful Data



Example (Old Faithful): Kernel regression with Epanechnikov kernel

```
Datos = read.table(’Datos-geyser.txt’,header=TRUE)
XY = cbind(Datos$X,Datos$Y); X = Datos$X; Y = Datos$Y; n = length(Y); mY = mean(Y)
par(mar = c(5, 4.5, 4, 2) + 0.1)
plot(XY, cex = 1.5, pch = 21,col=’black’, bg=’lightblue’,xlab=’X’,ylab=’Y’,
     cex.lab=1.5,cex.axis=1.5,main=’Old Faithful Data’,font.main=1,cex.main=1.5)

lines(X, mY*rep(1,n), lwd=3, col="black"); text(1.75,74,expression(bar(y)),cex=1.5)

#library(np)
nwU  = npreg(Y ˜ X,ckertype="epanechnikov") # NW regression with Epanechnikov kernel
newX = data.frame(seq(min(X),max(X),0.01)) ; predYnewX = predict(nwU,exdat=newX)
lines(t(newX), predYnewX, lwd=3, col="red") # Plot NW regression on the scatterplot
```



This suggests that we might improve the estimator by using a local polynomial of degree d instead of a local constant. Assume that we want to estimate the regression function m at some point x. Then we will be approximating $m(u)$ for values u in a neighbourhood of x by the polynomial

$$\begin{aligned} m_n(u) &= m_n(x) + m_n'(x)(u-x) + \ldots + \frac{1}{n!} f^{(d)}(x)(u-x)^d \\ &= \beta_0 + \beta_1(u-x) + \beta_2(u-x)^2 + \ldots + \beta_d(u-x)^d \end{aligned}$$

The d-th order *local polynomial regression estimator* is obtained by minimizing

$$\begin{aligned} \text{WSS} &= \sum_{i=1}^{n} (Y_i - m_n(X_i))^2 K\left(\frac{X_i - x}{h}\right) \\ &= \sum_{i=1}^{n} (Y_i - \beta_0 - \ldots - \beta_d(x - X_i)^d)^2 K\left(\frac{X_i - x}{h}\right). \end{aligned}$$

and setting $m_n(x) = \hat{\beta}_0$.

As a default choice it is usually recommended to take $d = 1$, thus obtaining the *local linear regression estimator*. This means that we approximate $m(u)$ for values u in a neighbourhood of x by the line $m_n(u) = m_n(x) + m_n'(x)(u - x) = a + b(u - x)$ and minimize

$$\text{WSS} \;=\; \sum_{i=1}^{n} (Y_i - a - b(x - X_i))^2 K\left(\frac{X_i - x}{h}\right).$$

From a direct calculation we get

$$m_n(x) = \hat{a} = \sum_{i=1}^{n} W_{in}(x) Y_i,$$

where $W_{in}(x) = w_i(x) / \sum_{i=1}^{n} w_i(x)$,

$$w_i(x) = K\left(\frac{x - X_i}{h}\right)(s_{n,2} - (x - X_i)s_{n,1}),$$

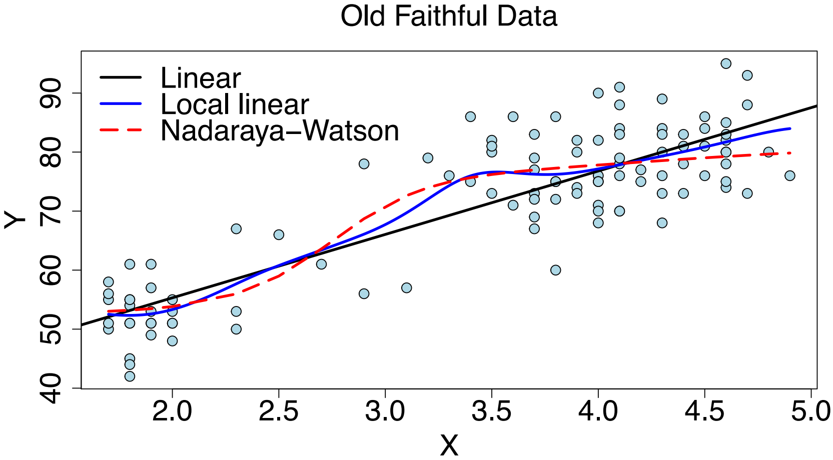
and

$$s_{n,j} = \sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right)(x - X_i)^j,\; j = 1, 2.$$

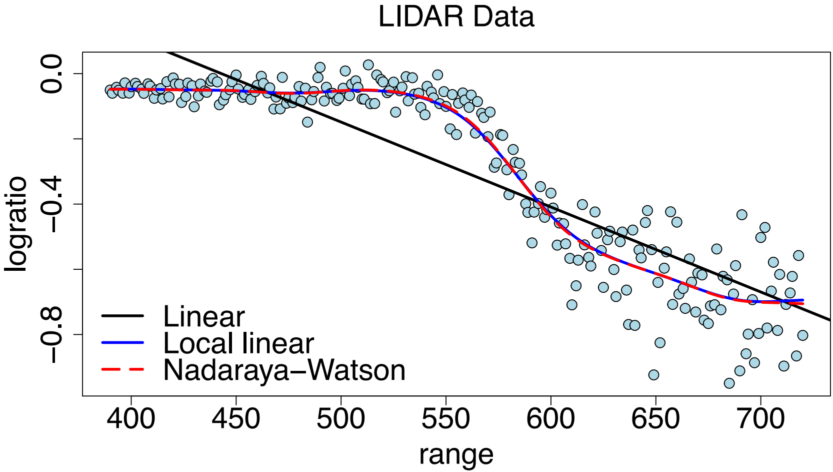
Example (Old Faithful): Linear, local linear and kernel regression with Gaussian kernel

```
Datos = read.table(’Datos-geyser.txt’,header=TRUE)
XY = cbind(Datos$X,Datos$Y); X = Datos$X; Y = Datos$Y;

par(mar = c(5, 4.5, 4, 2) + 0.1)
plot(XY, cex = 1.5, pch = 21,col=’black’, bg=’lightblue’,
     xlab=’X’,ylab=’Y’,cex.lab=1.5,cex.axis=1.5,
     main=’Old Faithful Data’,font.main=1,cex.main=1.5)

lr  = lm (Y ˜ X) # Linear regression
abline(lr, lwd=3) # Plot the linear regression on the existing scatterplot

#library(locpoly)
bw = dpill(X,Y) # bandwidth in local linear Gaussian regression estimate
LocLin = locpoly(X,Y,degree=1,kernel=’normal’,bandwidth=bw)
lines(LocLin,lwd=3,col="blue",lty=1)

#library(np)
nwG  = npreg(Y ˜ X,kernel=’gaussian’) # Nadaraya-Watson (NW) regression
Xord = sort(X, index.return = TRUE)
lines(Xord$x,nwG$mean[Xord$ix], lwd=3, col="red",lty=5) # Plot the NW regression on the existing
    scatterplot

legend("topleft",c(’Linear’,’Local linear’,’Nadaraya-Watson’),col=c(’black’,’blue’,’red’),
       cex=1.5, bty="n",lty=c(1,1,5),lwd=c(3,3,3),text.font=1)
```

`locfit` is another `R` package for performing local linear regression.


Old Faithful Data
Linear
Local linear
Nadaraya−Watson
Y
40
50
60
70
80
90
X
2.0
2.5
3.0
3.5
4.0
4.5
5.0

Example 6.3 (LIDAR):



The following result gives the large sample behaviour of the pointwise mean square error of the local linear estimator and shows why local linear regression is better than kernel regression. A proof can be found in Fan (1992) and Fan and Gijbels (1996).

Theorem 6.4: *Let $Y_i = m(X_i) + \sigma(X_i)\epsilon_i$, for $i = 1, \ldots, n$ and $a \leq X_i \leq b$. Let $x \in (a, b)$. Assume that $X_1, \ldots, X_n$ are a sample from a distribution with density f_X and that*

- **(i)** $f_X(x) > 0$,
- **(ii)** f_X, r'' *and* σ^2 *are continuous in a neighbourhood of* x,
- **(iii)** $h_n \to 0$ *and* $nh_n \to \infty$.

Then the local linear estimator and the kernel estimator both have variance

$$\mathbb{V}(m_n(x)) = \frac{1}{nh_n} \|K\|_2^2 \frac{\sigma^2(x)}{f_X(x)} + o_P\left(\frac{1}{nh_n}\right).$$

The Nadaraya-Watson kernel estimator has bias

$$\mathbb{E}(m_n(x)) - m(x) = h_n^2 m_2(K) \left(\frac{1}{2} m''(x) + \frac{m'(x) f_X'(x)}{f_X(x)} \right) + o_P(h^2),$$

whereas the local linear estimator has bias

$$\mathbb{E}(m_n(x)) - m(x) = h_n^2 m_2(K) \frac{1}{2} m''(x) + o_P(h^2).$$

Thus the local linear estimator is free from design bias. At the boundary points a and b, the Nadaraya-Watson kernel estimator has asymptotic bias of order h_n while the local linear estimator has bias of order h_n^2. In this sense, local linear estimation eliminates boundary bias.

The plug-in bandwidth:

Integrating the pointwise MSE gives an expression for the MISE (16), which can be minimized to obtain a global optimal bandwidth

$$h_n = \left(\frac{\|K\|_2^2 \int_a^b \sigma^2(x)dx}{n\, m_2^2(K) \int (m''(x))^2 f_X(x)dx} \right)^{1/5} = O(n^{-1/5}). \qquad (18)$$

Here we have assumed that the kernel K is a probability density.

To get an approximation to (18) fit a global quartic

$$\tilde{m}(x) = \hat{\beta}_0 + \hat{\beta}_1 x + \hat{\beta}_2 x^2 + \hat{\beta}_3 x^3 + \hat{\beta}_4 x^4 \quad \text{(pilot estimate of } m(x))$$

to the data (X_i, Y_i), $i = 1, \ldots, n$, using least squares and estimate $\sigma^2(x)$ by $\tilde{\sigma}^2 = n^{-1} \sum_{i=1}^n (Y_i - \tilde{m}(X_i))^2$.

Also,

$$
\begin{aligned}
n\int (m''(x))^2 f_X(x)dx &\simeq n\int_a^b (\tilde{m}''(x))^2\, f_X(x)\, dx \\
&\simeq \sum_{i=1}^n (\tilde{m}''(X_i))^2 .
\end{aligned}
$$

Thus, we approximate the optimal bandwidth in (18) by

$$
h_{PI} = \left(\frac{\|K\|_2^2\, \tilde{\sigma}^2\, (b-a)}{\sum_{i=1}^n (\tilde{m}''(X_i))^2} \right)^{1/5} .
$$

A slightly more refined plug-in bandwidth is implemented in the `R` function `dpill` of the `KernSmooth` package.

The cross-validation bandwidth:

The MISE (16) can be approximated by

$$\mathbb{E}\left(\frac{1}{n}\sum_{i=1}^{n}(m_n(X_i) - m(X_i)^2\right) = \mathbb{E}(\mathrm{ASE}).$$

In turn, this error can be approximated using the cross-validation (leave-one-out) procedure by

$$\frac{1}{n}\sum_{i=1}^{n}(m_{n,i}(X_i) - Y_i)^2, \tag{19}$$

where $m_{n,i}$ is the regression estimator based on the sample $X_1, \ldots, X_{i-1}, X_{i+1}, \ldots, X_n$.

The cross-validation bandwidth h_{CV} is the smoothing parameter minimizing the cross-validation score (19).

The generalization of the local linear estimator to any dimension p of the regressor $\mathbf{X}$ is as follows. We seek to minimize the weighted sum of squares

$$\begin{aligned} \text{WSS} &= \sum_{i=1}^{n} (Y_i - m_n(\mathbf{X}_i))^2 K_{\mathbf{H}}(\mathbf{X}_i - \mathbf{x}) \\ &= \frac{1}{h^d} \sum_{i=1}^{n} \left(Y_i - a_0 - \sum_{j=1}^{p} a_j (X_{ij} - x_j) \right)^2 K\left(\frac{\|\mathbf{X}_i - \mathbf{x}\|}{h} \right), \end{aligned}$$

where we have taken $K_{\mathbf{H}}(\mathbf{x}) := h^{-d} K(\|\mathbf{x}\|/h)$ and we have used the linear expansion

$$m_n(\mathbf{u}) = m_n(\mathbf{x}) + (\nabla m_n(\mathbf{x}))'(\mathbf{u} - \mathbf{x}) = a_0 + (a_1, \ldots, a_p)(\mathbf{u} - \mathbf{x})$$

for the $\mathbf{u}$'s in a neighbourhood of $\mathbf{x}$.

The local linear regression estimator at $\mathbf{x}$ is $m_n(\mathbf{x}) = \hat{a}_0$, where $\hat{\mathbf{a}} = (\hat{a}_0, \ldots, \hat{a}_p)'$ are the values of $\mathbf{a} = (a_0, \ldots, a_p)'$ minimizing the WSS. The solution is

$$\hat{\mathbf{a}} = (\mathbb{X}_{\mathbf{x}}' \mathbf{W}_{\mathbf{x}} \mathbb{X}_{\mathbf{x}})^{-1} \mathbb{X}_{\mathbf{x}}' \mathbf{W}_{\mathbf{x}} \mathbb{Y},$$

where $\mathbf{W}_{\mathbf{x}}$ is the diagonal matrix whose i-th diagonal element is $K(\|\mathbf{X}_i - \mathbf{x}\|/h)$, $\mathbb{Y} = (y_1, \ldots, y_n)'$ and

$$\mathbb{X}_{\mathbf{x}} = \begin{pmatrix} 1 & X_{11} - x_1 & \ldots & X_{1p} - x_p \\ \vdots & & \vdots & \vdots \\ 1 & X_{n1} - x_1 & \ldots & X_{np} - x_p \end{pmatrix}.$$

In dimension $p > 1$ the local linear estimator still reduces the boundary bias and the design bias with respect to the kernel regression estimator.

The following theorem is a special case of a result in Ruppert and Wand (1994).

Theorem 6.5: *Let the regularity conditions given in Ruppert and Wand (1994) hold. Assume* $\mathbf{x}$ *is a point from the interior of the support of* $\mathbf{X}$*. Then, conditional on* $\mathbf{X}_1, \ldots, \mathbf{X}_n$*, the bias of the local linear regression estimator* $m_n(\mathbf{x})$ *is*

$$\frac{1}{2}\, h^2\, m_2(K)\, trace(\mathcal{H}) + o_P(h^2),$$

where $m_2(K)\mathbf{I} := \int \mathbf{t}\mathbf{t}' K(\mathbf{t}) d\mathbf{t}$ *and* $\mathcal{H}$ *is the matrix of second partial derivatives of* m *evaluated at* $\mathbf{x}$*. The variance of* $m_n(\mathbf{x})$ *is*

$$\frac{\sigma^2(\mathbf{x}) \|K\|_2^2}{nh^p f_{\mathbf{X}}(\mathbf{x})}(1 + o_P(1)).$$

Also the bias at the boundary is of the same order as in the interior, namely, h^2*.*

There are not many `R` functions performing multivariate local linear regression. The function `locfit` from the package `locfit` is obscure.

A new book by Klemelä (2014), *Multivariate Nonparametric Regression and Visualization: With R and Applications to Finance*, is coming out precisely on this subject. Klemelä has written an R package, `regpro`, performing nonparametric regression techniques even in the multivariate setting. The R package and a tutorial is accesible at the web page

`http://cc.oulu.fi/~jklemela/regstruct/`

but I have not had time to try it yet. You are invited to do it!

References

Bowman, A.W. (1984). An alternative method of cross-validation for the smoothing of density estimates. *Biometrika*, 71, 353–360.

Chacón, J.E. (2004). *Estimación de densidades: algunos resultados exactos y asintóticos*. Ph.D. Thesis. Univ. Extremadura.

Devroye, L. and Györfi, L. (1985). *Nonparametric density estimation. The L_1 view*. Wiley. Downloadable at `www.szit.bme.hu/~gyorfi/L1bookBW.pdf`.

Duong, T. and Hazelton, M.L. (2005). Cross validation bandwidth matrices for multivariate kernel density estimation. *Scandinavian Journal of Statistics*, 32, 485–506.

Fan, J. (1992). Design-adaptive nonparametric regression. *Journal of the American Statistical Association*, 87, 998–1004.

Fan, J. and Gijbels, I. (1996). *Local Polynomial Modelling and Its Applications*. Chapman and Hall.

Györfi, L., Kohler, M., Krzyżak, A. and Walk, H. (2002). *A Distribution-Free Theory of Nonparametric Regression*. Springer.

Härdle, W. (1990). *Applied Nonparametric Regression*. Cambridge University Press. E-book at `http://sfb649.wiwi.hu-berlin.de/fedc_homepage/xplore/ebooks/html/anr/`.

Härdle, W., Müller, M., Sperlich, S. and Werwatz, A. (2004). *Nonparametric and Semiparametric Models*. Springer. E-book at `http://sfb649.wiwi.hu-berlin.de/fedc_homepage/xplore/ebooks/html/spm/`.

Klemelä, J. (2009). *Smoothing of Multivariate Data: Density Estimation and Visualization*. Wiley.

Klemelä,J. (2014). *Multivariate Nonparametric Regression and Visualization: With R and Applications to Finance*. Wiley.

Marron, J.S. and Härdle, W. (1986). Random approximations to some measures of accuracy in nonparametric curve estimation. *Journal of Multivariate Analysis*, 20, 91–113.

Nadaraya, E.A. (1964). On estimating regression. *Theory of Probability and its Applications*, 9, 141–142.

Nadaraya, E.A. (1989). *Nonparametric estimation of probability densities and regression curves*. Kluwer Academic Publishers.

Rudemo, M. (1982). Empirical choice of histograms and kernel density estimators. *Scandinavian Journal of Statistics*, 9, 65–78.

Ruppert, D. and Wand, M.P. (1994). Multivariate locally weighted least squares regression. *The Annals of Statistics*, 22, 1346–1370.

Ruppert, D., Wand, M. and Carroll, R. (2003). *Semiparametric Regression*. Cambridge University Press.

Scott, D.W. (1992). *Multivariate Density Estimation: Theory, Practice, and Visualization*. Wiley.

Sheather, S.J. (2004). Density estimation. *Statistical Science*, 19, 588–597.

Sheather, S.J. and Jones, M.C. (1991). A reliable data-based bandwidth selection method for kernel density estimation. *Journal of the Royal Statistical Society, Series B*, 53, 683–690.

Sigrist, M.E. (1994). *Air Monitoring by Spectroscopic Techniques*. Wiley.

Silverman, B.W. (1986). *Density Estimation for Statistics and Data Analysis*. Chapman & Hall.

Simonoff, J.S. (1996). *Smoothing Methods in Statistics*. Springer.

Stone, C.J. (1977). Consistent nonparametric regression. *The Annals of Statistics*, 5, 595-645.

Stone, C.J. (1984). An asymptotically optimal window selection rule for kernel density estimates. *The Annals of Statistics*, 12, 1285–1297.

Tsybakov, A.B. (2009). *Introduction to Nonparametric Estimation*. Springer.

Wand, M.P. and Jones, M.C. (1994). Multivariate plug-in bandwidth selection. *Computational Statistics*, 9, 97–177.

Wand, M.P. and Jones, M.C. (1995). *Kernel Smoothing*. Chapman & Hall.

Wasserman, L. (2006). *All of nonparametric statistics*. Springer.

Watson, G.S. (1964). Smooth regression analysis. *Sankhyā: The Indian Journal of Statistics, Series A*, 26, 359–372.