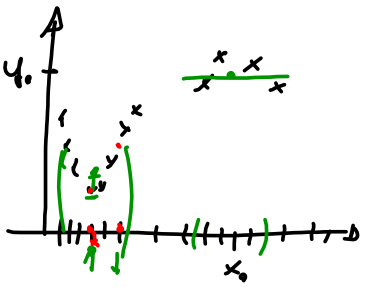


$$\hat{y}_0 = \frac{1}{k}\sum_{x \in N_k(x_0)} y_x$$

First improvement: weight depending on the distance

- our assumption is that the response y_0 we want to predict is similar to the responses y_i of the points x_i close to x_0
  - → in K-NN we average on the K closest points
  - → we expect closer points to be more similar to farther points
  - ⇒ we can give different weights based on the distance to x_0

$$w_i = \frac{1}{h} K\left(\frac{x_i - x_0}{h}\right) \quad \text{Kernel}$$

h is a tuning parameter (bandwidth or smoothing parameter)

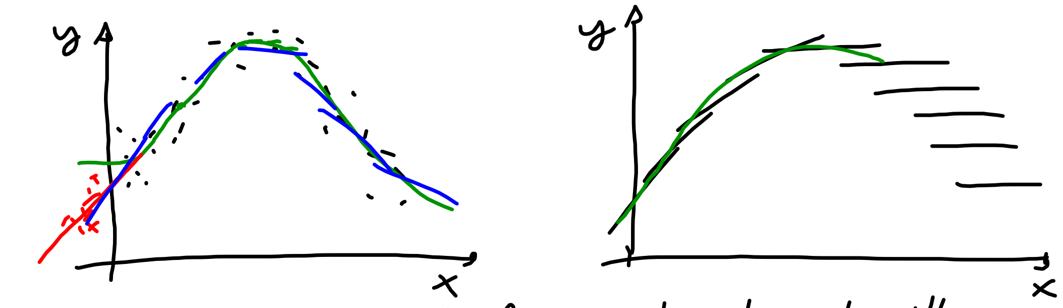
$$\hat{y}_0 = \sum_{i \in N_k(x_0)} w_i y_i$$

Typical Kernels are:

normal	$\frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{1}{2}\left(\frac{x_i - x_0}{h}\right)^2\right\}$	defined in $\mathbb{R}$
Epanechnikov	$\frac{3}{4}\left[1 - \left(\frac{x_i - x_0}{h}\right)^2\right]$	" " $(-1;1)$
biquadratic	$\frac{15}{16}\left(1 - \left(\frac{x_i - x_0}{h}\right)^2\right)^2$	" " $(-1;1)$
tricubic	$\frac{70}{81}\left(1 - \left\lvert\frac{x_i - x_0}{h}\right\rvert^3\right)^3$	" " $(-1;1)$
rectangular	$\frac{1}{2}$	" " $(-1;1)$

used (empirical evidence showed that it is not really important which kernel is implemented) → much more important the choice of h

Second improvement: from constant to linear approximation



Instead of approximating the function at each point with a constant, we use a line

$$y = f(x) + \varepsilon \qquad \hat{y} = \hat{\beta}_0$$

$\downarrow$ Taylor expansion around x_0

$$f(x) = \underbrace{f(x_0)}_{\beta_0} + \underbrace{f'(x_0)}_{\beta_1}(x - x_0) + \overset{\text{rest}}{\cancel{o(|x-x_0|)}}$$

To estimate β_0 and β_1, we can extend the concept of least squares estimator

$$\hat{\beta}_0, \hat{\beta}_1 = \underset{\beta_0, \beta_1}{\operatorname{argmin}} \sum_{i=1}^{n} \left(y_i - \beta_0 - \beta_1 (x_i - x_0)\right)^2 w_i$$

where w_i are weights that penalizes the contribution of observations far from x_0

The solution is

$$(\beta_0, \beta_1) = \hat{\beta} = (X^T W X)^{-1} X^T W y$$

$\uparrow$ weighted least squares

where $\underset{n\times 2}{X} = \begin{pmatrix} 1 & x_1 - x_0 \\ \vdots & \vdots \\ 1 & x_n - x_0 \end{pmatrix}$

$$\underset{n\times n}{W} = \begin{pmatrix} \frac{1}{h} K\left(\frac{x_1 - x_0}{h}\right) & & 0 \\ & \ddots & \\ 0 & & \frac{1}{h} K\left(\frac{x_n - x_0}{h}\right) \end{pmatrix}$$

Choice of the bandwidth

h is the tuning parameter, and control the model complexity

→ smaller h means a smaller window (in the picture, the light blue area), so we estimate $f(x)$ based on the local behaviour of the data

→ less bias, more variance → when h is too small, we have a bumpy curve that follow the randomness in the data → OVERFITTING

→ larger h means larger window, more data are used to estimate $f(x)$

→ lower variance, higher bias → when h is too large, our curve does not capture the systematic part of the data → UNDERFITTING

The optimal choice of h is related to the bias-variance trade-off.
In practice h is chosen by cross-validation or AIC_c

Theoretical aspects

Under specific conditions ($Var(\varepsilon_i) = \sigma^2 \ \forall i$, $Cov(\varepsilon_i, \varepsilon_j) = 0 \ \forall i \neq j$ and regularity condition)

h sufficiently small
n " large

$$E[\hat{f}(x)] \approx f(x) + \underbrace{\frac{h^2}{2}\sigma_w^2 f''(x)}_{\text{bias} = b(x)}$$

$$Var[\hat{f}(x)] \approx \frac{\sigma^2}{nh} \frac{\alpha(w)}{g(x)}$$

with $\sigma_w^2 = \int z^2 w(z)\, dz$, $\alpha(w) = \int w(z)^2 dz$, $g(x)$ is the density from which x_i have been sampled

$h \to 0 \qquad \underline{E[\hat{f}(x)] - f(x)} \to 0 \qquad Var(\hat{f}(x)) \to \infty$

$h \to \infty \qquad Var(\hat{f}(x)) \to 0 \qquad E[\hat{f}(x)] - f(x) \to \infty$

Having expected value and variance, we can, in theory, use the asymptotic distribution

$$\frac{\hat{f}(x) - f(x) - b(x)}{\sqrt{Var(\hat{f}(x))}} \sim N(0;1)$$

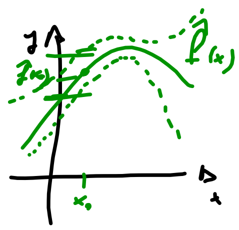
to construct <u>confidence bands</u> around $\hat{f}(x)$

↳ same concept of confidence intervals, provide a ~~measure~~ indication of the uncertainty around our estimate

Problem: the bias contains a term, $f''(x)$ that is unknown and we cannot estimate, not even approximatively

Solution: use variability bands on the form

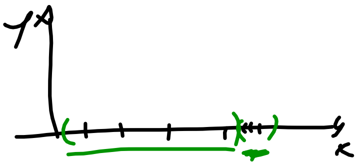
$$\hat{f}(x) - z_{1-\frac{\alpha}{2}}\sqrt{\hat{Var}(\hat{f}(x))} \quad , \quad \hat{f}(x) + z_{1-\frac{\alpha}{2}}\sqrt{\hat{Var}(\hat{f}(x))}$$



where $z_{1-\frac{\alpha}{2}}$ is the $1-\frac{\alpha}{2}$ quantile of the standard normal distribution

Note:
- the variability bands are computed pointwise
- for each points, they are not confidence intervals
- the confidence level for a fixed x is anyway $1-\alpha$, but it does not work for the entire curve

loess
- combine the local linear regression we saw today with the idea of a fixed amount of points used for the local estimation (like KNN)
  - → instead of computing the smoothing window based on the distance from x_0, the window is constructed in order to include a specific amount (or proportion) of data
    - → smoothing parameter
  - → idea behind: it more reasonable to use larger smoothing windows where the data are more sparse



Extension to several dimensions

In theory, our non-parametric estimation of $f(x)$ can be extend to more dimensions

E.g. $p=2$ $\qquad y = f(x_1, x_2) + \varepsilon \qquad \mathbb{R}^2 \to \mathbb{R}$

Let $y_i \in \mathbb{R}$, $x_i = (x_{i1}, x_{i2}) \in \mathbb{R}^2 \qquad i=1,\dots,n$

$$\min_{\beta_0,\beta_1,\beta_2} \sum_{i=1}^{n} \left(y_i - \beta_0 - \beta_1(x_{i1} - x_{01}) - \beta_2(x_{i2} - x_{02})\right)^2 w_i \qquad (1)$$

where now w_i has form

$$w_i = \frac{1}{h_1 h_2} K\left(\frac{x_{i1} - x_{01}}{h_1}\right) K\left(\frac{x_{i2} - x_{02}}{h_2}\right)$$

Now we have 2 tuning (smoothing) parameters, one for each dimensions (we need to take into account the different variability of x_1 and x_2)

Again, the solution of the minimization problem (1) is based on weighted least squares

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

where: $y = (y_1, \dots, y_n)^T$

$W = \mathrm{diag}(w_1, \dots, w_n)$

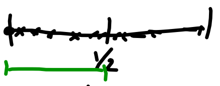
$$\underset{n\times 3}{X} = \begin{pmatrix} 1 & x_{11} - x_{01} & x_{12} - x_{02} \\ \vdots & \vdots & \vdots \\ 1 & x_{n1} - x_{01} & x_{n2} - x_{02} \end{pmatrix}$$

In theory it works for any p (not only $p=2$ like here above).

In practice, these techniques are never used for $p > 2$

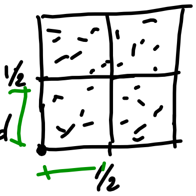
- difficulties to plot / visualize the results
- hard to interpret the results
- suffer from the curse of dimensionality: increasing the number of dimensions, the number of observed points close to the point of interest decreases really quickly

E.g.

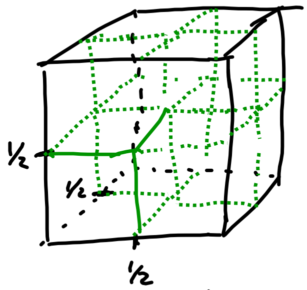


if the observations are uniformly distributed

1/2



1/4



1/8

In order to compensate for the increased dimension of the space, in order to base our estimate on the same amount of data, we need n^p observations

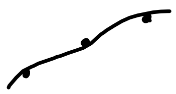
(e.g., if in 1 dimension, we want $\hat{f}(x)$ based on 100 observations,
" 2 " 100^2, with 5 variables 100^5 (10 billions)
" 10 " 100^{10})

Basically, when the problem is multidimensional (with a large number of dimensions) we cannot use our non-parametric techniques

- issues with the number of observation;
- computational issues

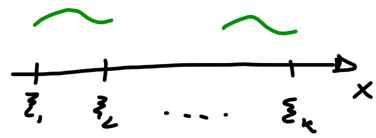
Possible way to proceed: construct principal components, use let us say the first two.
↳ maintain as much variability as possible in as few dimensions as possible

SPLINES



piecewise polynomial function
- split the support in several pieces ($\text{fix } \xi_i$, $\xi_1 < \dots < \xi_K$)
- fit a polynomial in each piece

→ piecewise constant
→ " linear
→ continuous piecewise linear
→ discontinuous cubic
→ continuous cubic
→ cubic continuous in first derivative
→ " " " 2nd "



can be any, the preferred is 3

$$\begin{cases} f(\xi_i^-) = f(\xi_i^+) \\ f'(\xi_i^-) = f'(\xi_i^+) \\ f''(\xi_i^-) = f''(\xi_i^+) \end{cases}$$

cubic splines

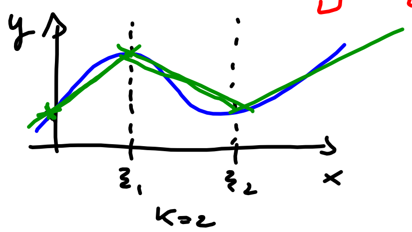
How can we use splines to evaluate the relationship between y and x (regression splines)?

Simplest case, $\boxed{K=2}$, $d=1$ (2 knots, straight lines)

→ parametric case $f(x;\beta)$

$$f(x;\beta) = \beta_0 + \beta_1 x + \beta_2 (x-\xi_1)_+ + \beta_3 (x-\xi_2)_+$$

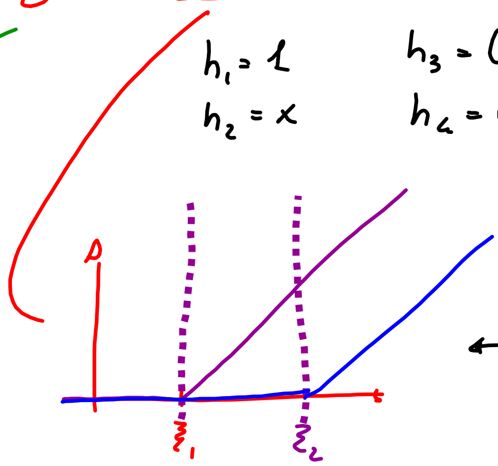
basis



$h_1 = 1$ $\quad h_3 = (x-\xi_1)_+$

$h_2 = x$ $\quad h_4 = (x-\xi_2)_+$

$$\hat{f}(x;\beta) = \sum_{j=1}^{4} \hat{\beta}_j h_j(x)$$



← form of $(x-\xi_2)_+$

In the case of cubic splines with a generic number of knots K,

$$f(x;\beta) = \sum_{j=1}^{K+4} \beta_j h_j(x)$$

where

$$h_j(x) = x^{j-1} \quad \text{for } j = 1, \dots, 4$$

$$\begin{cases} h_1(x) = x^0 = 1 \\ h_2(x) = x^1 = x \\ h_3(x) = x^2 \\ h_4(x) = x^3 \end{cases}$$

$$h_{j+4}(x) = (x-\xi_j)_+^3 \quad \text{for } j = 1, \dots, K$$

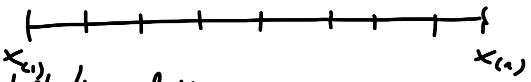
We need to decide K, the number of knots, and their position

K is the complexity parameter: higher values, more complex functions

↳ find by <u>cross-validation</u>

Once K has been selected, we need to place $\xi_1, \dots, \xi_K$ in the support of x

- uniformly among the range of x



- use the quantiles of the empirical distribution of x