UNIVERSITÀ DEGLI STUDI DI TRIESTE

Dipartimento di scienze economiche, aziendali, matematiche e statistiche "Bruno de Finetti"

5. Regressione semiparametrica

Francesco Pauli

DEAMS

Università di Trieste

A.A. 2020/2021

Modelli parametrici, semiparametrici, non parametrici

Si ha un modello parametrico quando la famiglia di distribuzioni all'interno della quale cerchiamo una distribuzione che descriva i dati è indicizzata da un parametro $\theta \in \mathbb{R}^d$, **con d non troppo grande e fisso**.

Muoversi verso i metodi semiparametrici o non parametrici significa

- ridurre le assunzioni
- aumentare il numero di parametri e, in qualche senso, stimarlo

(non c'è una separazione netta).

Esempio: stima della distribuzione

Esempio: stima della distribuzione di $X_1, \ldots, X_n \sim F()$

Stima parametrica di F: supponiamo

$$F \in \mathcal{F} = \{F_\theta() : \theta \in \mathbb{R}^d\}$$

sia ad esempio $X_i \sim \text{IID}(\mathcal{N}(\mu, \sigma^2))$, stimiamo θ (ad es. SMV) e $F_{\hat{\theta}}$ è la stima di F.

Stima non parametrica di F: si assume F sia una FdR, una stima non parametrica è la **FdR empirica**

$$\hat{F}_n(x) = \frac{1}{n} \sum_{i=1}^{n} |x_i \leq x|$$

Esempio: regressione

Esempio: stima della funzione di regressione $E(Y|X = x)$ da un campione (x_i, Y_i), $i = 1, \ldots, n$

Sono modelli parametrici

- il modello lineare: $(Y|X = x) \sim \mathcal{N}(\beta_1 + \beta_2 x, \sigma^2)$ indpendenti dove

$$E(Y|X = x) = \beta_1 + \beta_2 x$$

  e dove si stima $\theta = (\beta_1, \beta_2)$ ad esempio con la MV.
- anche i GLM, dove

$$g(E(Y|X = x)) = \beta_1 + \beta_2 x$$

  con g funzione legame nota.

In un modello non/semi parametrica/o: si assume $E(Y|X = x) = f(x)$ dove f appartiene a una classe di funzioni sufficientemente flessibile (non ci sono parametri di interesse diretto).

Regressione non parametrica

Si assume che le osservazioni siano indipendenti e

$$E(Y|X = x) = f(x); \quad V(Y|X = x) = \sigma^2$$

dove f è una funzione "regolare" (continua con qualche derivata continua).

Due approcci

- tecniche "locali"
  - se avessimo tante osservazioni per ciascun x_0 potremmo stimare $f(x_0)$ come una media campionaria.
  - in generale, avendo un'osservazione per ciascun x potremmo usare le osservazioni vicine.

Regressione non parametrica

Si assume che le osservazioni siano indipendenti e

$$E(Y|X = x) = f(x); \quad V(Y|X = x) = \sigma^2$$

dove f è una funzione "regolare" (continua con qualche derivata continua).

Due approcci

- tecniche "locali": stima $E(Y|X = x_0)$ usando punti vicini a x_0.
- tecniche "globali" (spline)
  - definiamo una classe di funzioni $f(x;\theta)$ abbastanza flessibile da poter approssimare qualunque funzione regolare $f(\cdot)$
  - stimiamo f scegliendo il miglio rappresentante in $f(x;\theta)$

Regressione non parametrica

Si assume che le osservazioni siano indipendenti e

$$E(Y|X = x) = f(x); \quad V(Y|X = x) = \sigma^2$$

dove f è una funzione "regolare" (continua con qualche derivata continua).

Due approcci

- tecniche "locali": stima $E(Y|X = x_0)$ usando punti vicini a x_0.
- tecniche "globali" (spline): definiamo un modello flessibile per $f(x; \boldsymbol{\theta})$

Regressione non parametrica

Si assume che le osservazioni siano indipendenti e

$$E(Y|X = x) = f(x); \quad V(Y|X = x) = \sigma^2$$

dove f è una funzione "regolare" (continua con qualche derivata continua).

Due approcci

- tecniche "locali": stima $E(Y|X = x_0)$ usando punti vicini a x_0.
- tecniche "globali" (spline): definiamo un modello flessibile per $f(x; \boldsymbol{\theta})$

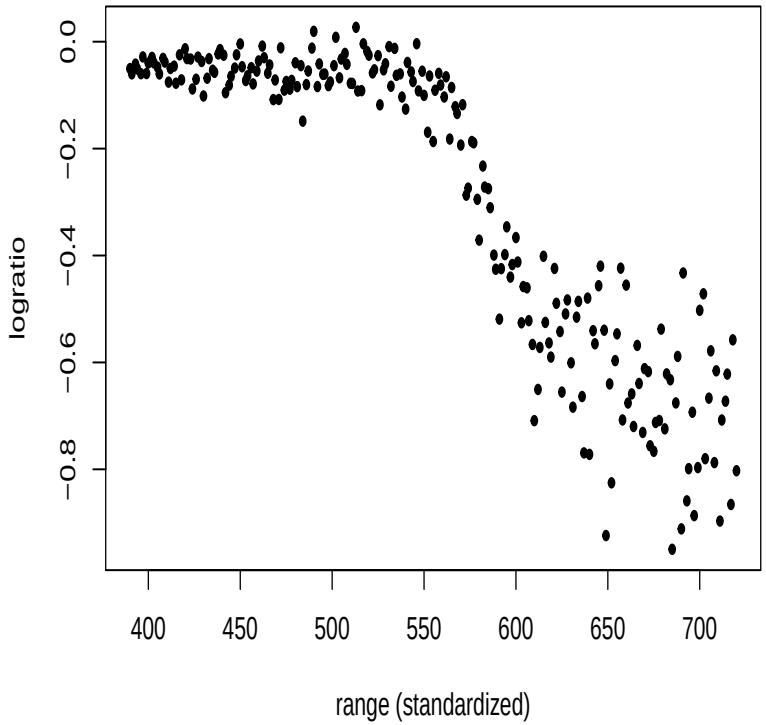
Un aspetto cruciale in entrambi i metodi è determinare quanto liscia debba essere $\hat{f}$ che si traduce in

- decidere cosa significa "vicino"
- decidere quanto flessibile dev'essere il modello $f(x; \boldsymbol{\theta})$

In entrambi i casi, serve un compromesso tra distorsione e varianza.

Indice

Esempio: dati "lidar"



LIDAR = LIght Detection And Ranging

- è una tecnica per individuare composti chimici nell'atmosfera
- x: distanza percorsa prima della riflessione
- y: logaritmo del rapporto tra luce ricevuta tra le due fonti laser

- L'obiettivo è stimare

$$f(x) = E(Y|X = x)$$

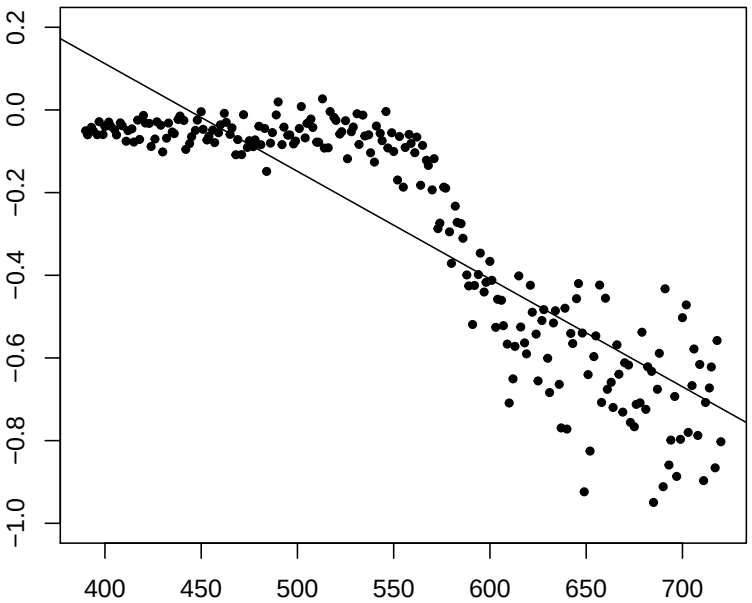
- (Esempio ben noto dove le tecniche banali, trasformazioni o regressione polinomiale, funzionano male.)

LIDAR: modello lineare

Assumiamo

$$y = X\beta + \varepsilon$$

dove $X \in \mathcal{M}_{n\times p}$, $\beta \in \mathbb{R}^p$, $\varepsilon \sim N(0, \sigma^2 I)$,



Non molto soddisfacente...

Usando la massima verosimiglianza

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

sicché

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

dove $H = X(X^T X)^{-1} X^T$ è la matrice di proiezione da $\mathbb{R}^n$ al sottospazio generato dalle colonne di X, si ricordi che

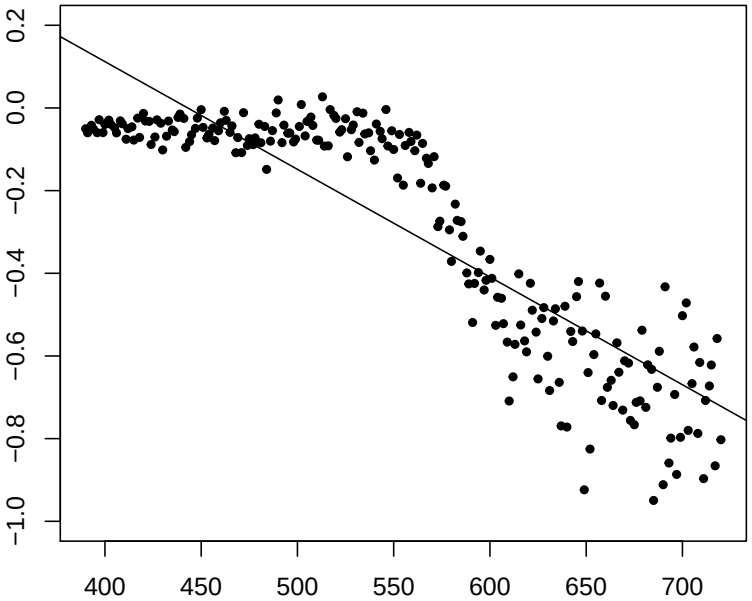
$$\text{trace} H = p$$

LIDAR: modello lineare

Assumiamo

$$\mathrm{y} = X\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

dove $X \in \mathcal{M}_{n\times p}$, $\boldsymbol{\beta} \in \mathbb{R}^p$, $\boldsymbol{\varepsilon} \sim N(0, \sigma^2 I)$,



Non molto soddisfacente...

Si noti che

$$\hat{\mathrm{y}} = X\hat{\boldsymbol{\beta}}\mathrm{y} = X(X^TX)^{-1}X^T\mathrm{y}$$

significa che la stima del valore atteso condizionato è

$$\widehat{E(Y|X=x)} = \hat{f}(x) = \sum_{i=1}^{n} h_i(x)\,Y_i$$

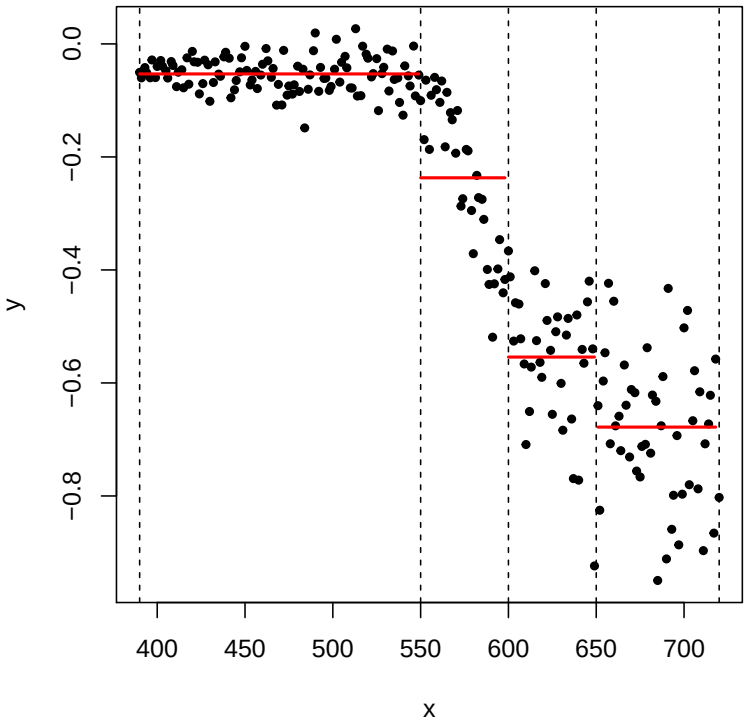
dove

$$h(\mathrm{x})^T = \mathrm{x}^T(X^TX)^{-1}X^T$$

LIDAR: costante a tratti

Consideriamo una partizione dello spazio della variabile esplicativa, indichiamo gli estremi degli intervalli con

$$-\infty = c_0 < c_1 < \ldots < c_{K-1} < c_K = +\infty$$



e stimiamo $E(Y|X = x)$ assumendo sia costante negli intervalli

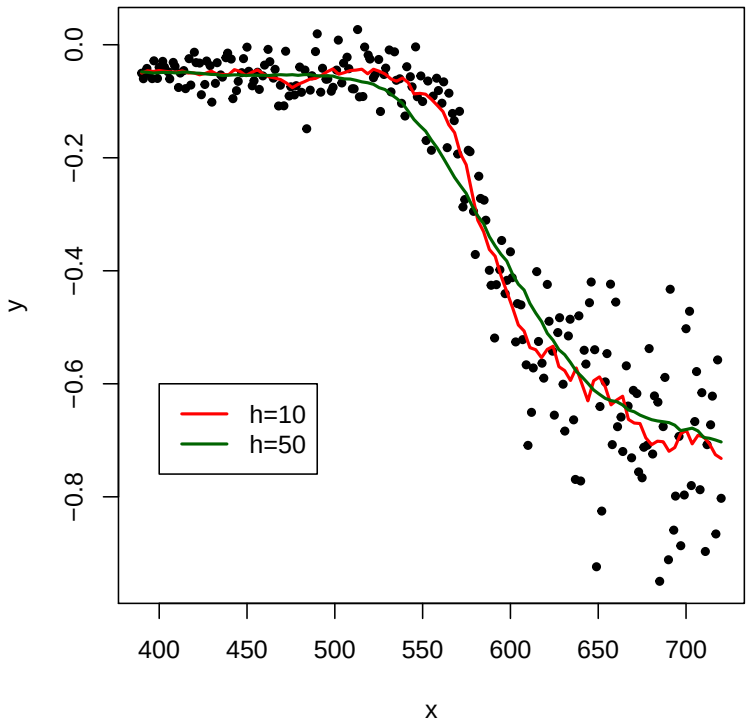
$$\hat{f}(x) = \frac{\sum_{k=0}^{K-1} \sum_{i=1}^{n} y_i I_{[c_k, c_{k+1}]}(x_i)}{\sum_{k=0}^{K-1} \sum_{i=1}^{n} I_{[c_k, c_{k+1}]}(x_i)}$$

Il risultato

- non è liscio (addirittura discontinuo), e
- dipende dalla scelta degli intervalli.

LIDAR: media mobile

Se assumiamo che $f(x)$ sia continua, allora è ragionevole stimare $f(x)$ come la media di valori di Y_i che corrispondono a x_i vicini a x.

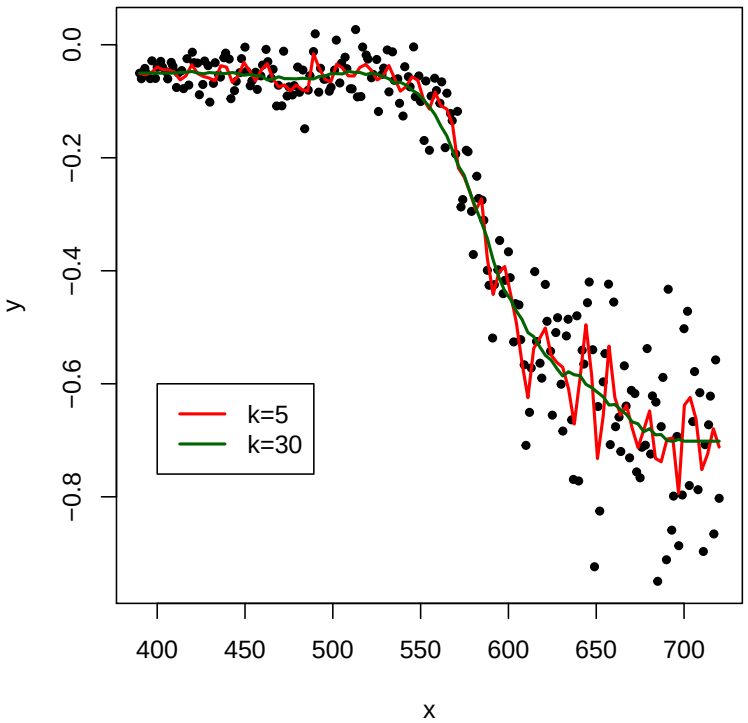


In particulare potremmo usare la media di quegli x_i che giacciono in un intorno di x di raggio h

$$\hat{f}(x) = \frac{\sum_{i=1}^{n} y_i I_h(|x - x_i|)}{\sum_{i=1}^{n} I_h(|x - x_i|)}$$

LIDAR: media mobile (vicini più vicini)

Se assumiamo che $f(x)$ sia continua, allora è ragionevole stimare $f(x)$ come la media di valori di Y_i che corrispondono a x_i vicini a x.



Alternativamente potremmo usare la media dei k vicini più vicini ad x,

$$N_k(x) = \{x_i : |x - x_i| \leq d_{(k)}\}$$

dove $d_i = |x - x_i|$ e $d_{(1)} \leq \ldots \leq d_{(n)}$ sono le distanze ordinate, allora

$$\hat{f}(x) = \frac{1}{k} \sum_{i=1}^{n} y_i I_{N_k(x)}(x_i)$$

Errore di stima: distorsione e varianza

La stima

$$\hat{f}(x) = \frac{1}{k}\sum_{i=1}^{n} y_i I_{N_k(x)}(x_i) = \frac{1}{k}\sum_{y_i \in N_k(x)} y_i$$

dove $N_k(x) = \{x_i : |x - x_i| \leq d_{(k)}\}$ è basata su k osservazioni: **più grande è** k,

- più osservazioni sono usate e quindi minore è la variabilità:
- d'altra parte, s'impiegano osservazioni più distanti, a seconda della forma di $f()$ in un intorno di x, la media delle osservazioni può differire più o meno marcatamente da $E(Y|X = x) = f(x)$:

Il compromesso tra distorsione e varianza è una caratteristica distintiva dei lisciatori.

Errore di stima: distorsione e varianza

La stima

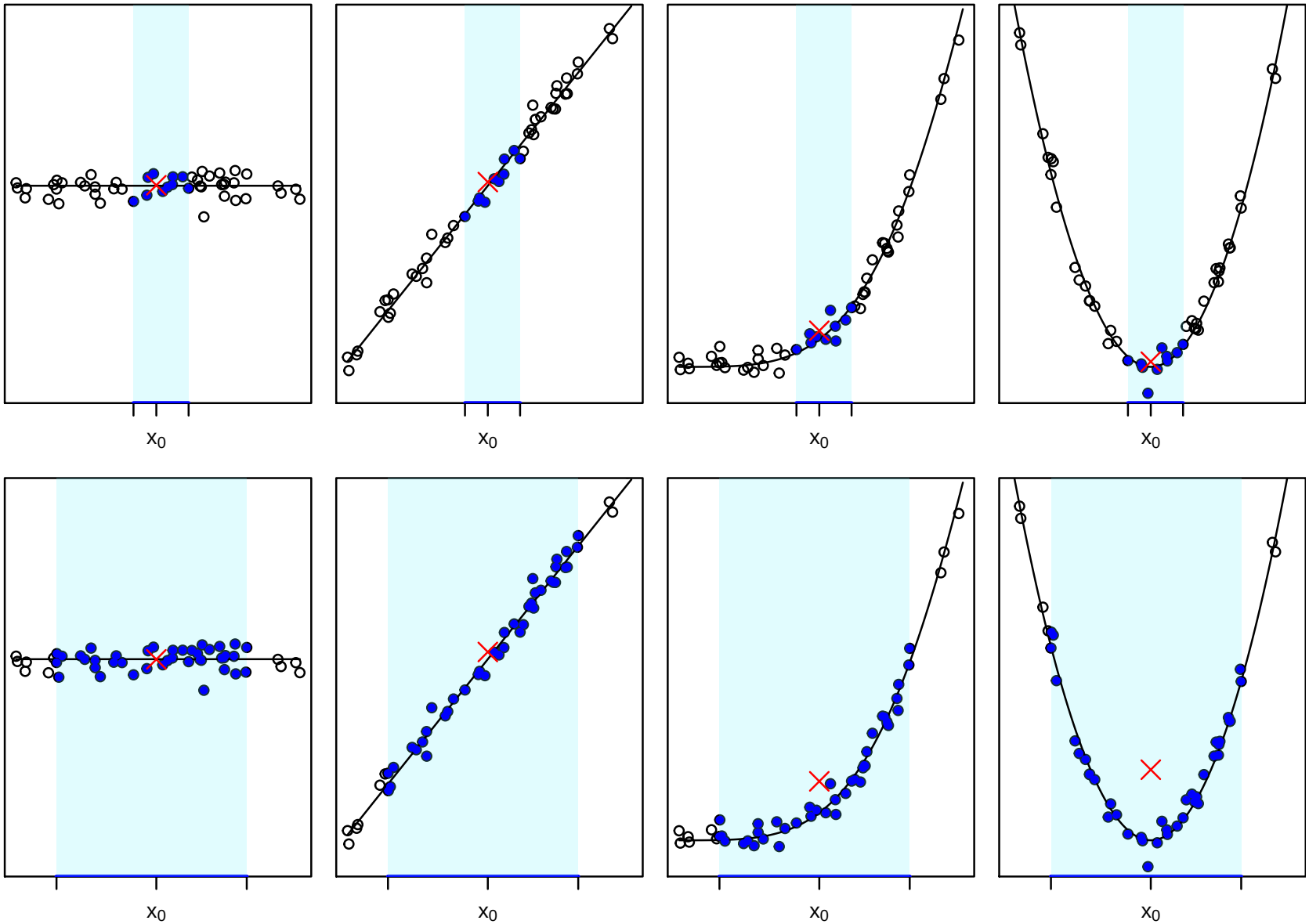
$$\hat{f}(x) = \frac{1}{k}\sum_{i=1}^{n} y_i I_{N_k(x)}(x_i) = \frac{1}{k}\sum_{y_i \in N_k(x)} y_i$$

dove $N_k(x) = \{x_i : |x - x_i| \leq d_{(k)}\}$ è basata su k osservazioni: **più grande è** k,

- più osservazioni sono usate e quindi minore è la variabilità:
  - **più piccola è la varianza**
- d'altra parte, s'impiegano osservazioni più distanti, a seconda della forma di $f()$ in un intorno di x, la media delle osservazioni può differire più o meno marcatamente da $E(Y|X = x) = f(x)$:
  - **più grande è la distorsione**

Il compromesso tra distorsione e varianza è una caratteristica distintiva dei lisciatori.

Distorsione, forma di f e k



Derivazione teorica di distorsione e varianza

Sia $N_k(x) = \{x_i : |x - x_i| \leq d_{(k)}\}$, e lo stimatore

$$\hat{f}(x) = \frac{1}{k} \sum_{i=1}^{n} y_i I_{N_k(x)}(x_i) = \frac{1}{k} \sum_{y_i \in N_k(x)} y_i$$

La varianza è (assumendo $V(Y_i) = \sigma^2$ per ogni i)

$$V(\hat{f}(x)) = \frac{1}{k} \sum_{y_i \in N_k(x)} V(Y_i) = \frac{\sigma^2}{k}$$

Derivazione teorica di distorsione e varianza

Sia $N_k(x) = \{x_i : |x - x_i| \leq d_{(k)}\}$, e lo stimatore

$$\hat{f}(x) = \frac{1}{k} \sum_{i=1}^{n} y_i I_{N_k(x)}(x_i) = \frac{1}{k} \sum_{y_i \in N_k(x)} y_i$$

La distorsione è

$$\begin{aligned} E(\hat{f}(x)) - f(x) &= \frac{1}{k} \sum_{N_k(x)} (f(x_i) - f(x)) \\ &\approx \frac{1}{k} \sum_{N_k(x)} \left(f'(x)(x_i - x) + \frac{1}{2} f''(x)(x_i - x)^2 \right) \end{aligned}$$

assumendo le covariate equidistanziate: $x_{i+1} - x_i = \Delta$

$$\approx \frac{2k(k+2)(k+1)}{6k} f''(x)\Delta^2$$

Derivazione teorica di distorsione e varianza

Sia $N_k(x) = \{x_i : |x - x_i| \leq d_{(k)}\}$, e lo stimatore

$$\hat{f}(x) = \frac{1}{k} \sum_{i=1}^{n} y_i I_{N_k(x)}(x_i) = \frac{1}{k} \sum_{y_i \in N_k(x)} y_i$$

Quindi l'MSE è

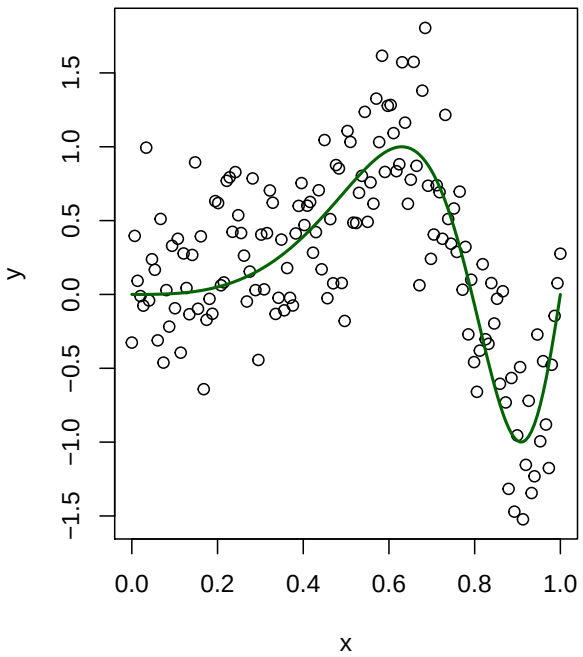
$$E((\hat{f}(x) - f(x))^2) \approx \left(\underbrace{\frac{2k(k+2)(k+1)}{6k} f''(x)\Delta^2}_{distorsione} \right)^2 + \underbrace{\frac{\sigma^2}{k}}_{varianza}$$

e quindi

- la distorsione cresce con k e con $|f''|$
- la varianza decresce con k

Distorsione e varianza, esempio

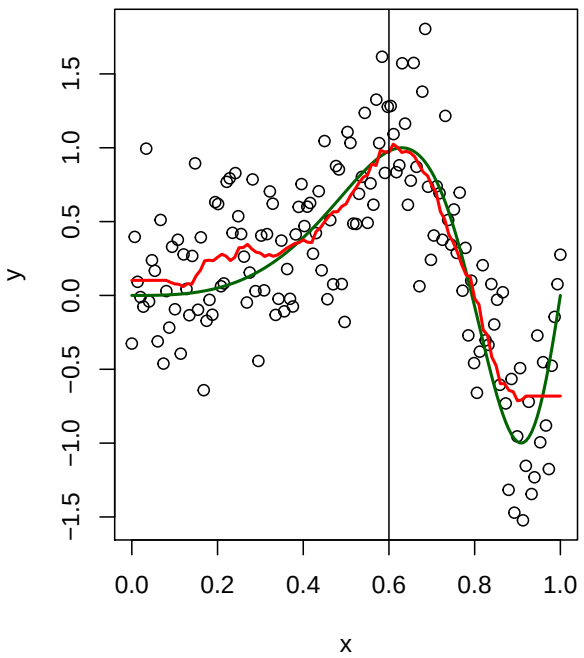
Consideriamo un campione, la vera $f(\cdot)$ è in verde,

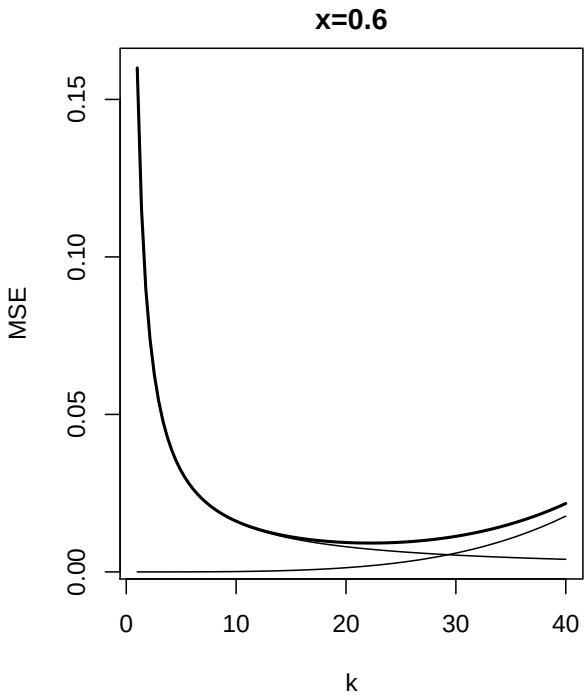


Distorsione e varianza, esempio

Consideriamo un campione, la vera $f(\cdot)$ è in verde,

- calcoliamo distorsione, varianza e quindi MSE per $\hat{f}_k(0.6)$ in funzione di k, individuiamo un valore ottimale di k.

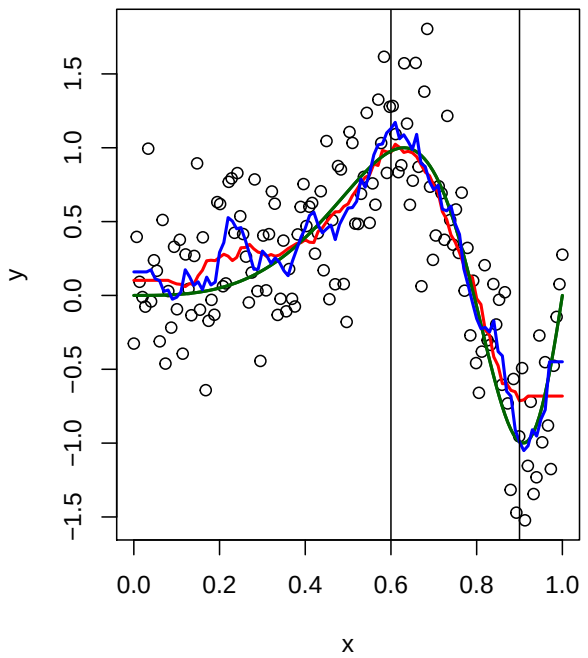


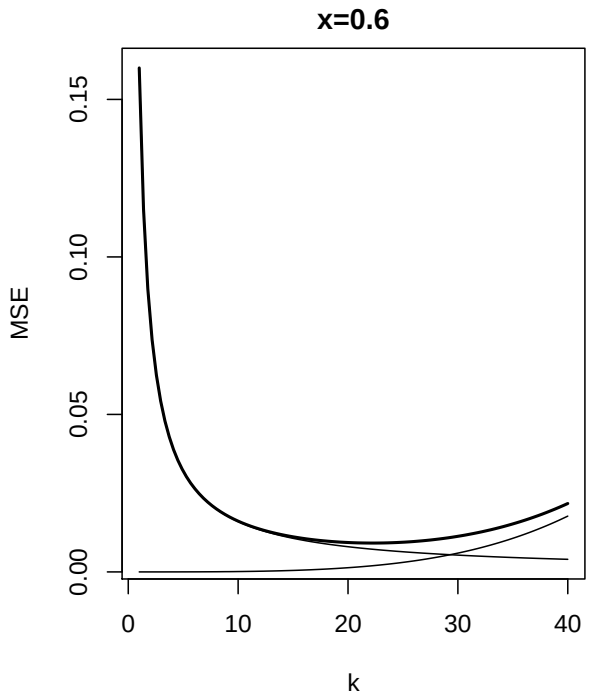


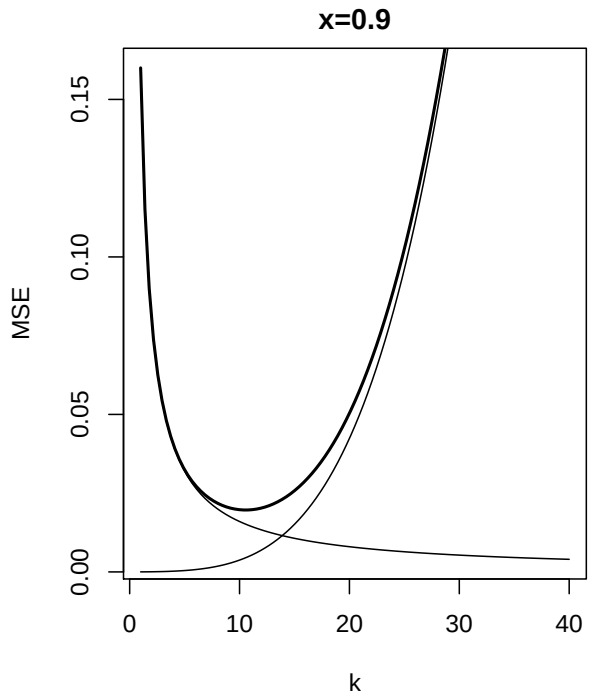
Distorsione e varianza, esempio

Consideriamo un campione, la vera $f(\cdot)$ è in verde,

- calcoliamo distorsione, varianza e quindi MSE per $\hat{f}_k(0.6)$ in funzione di k, individuiamo un valore ottimale di k.
- facciamo lo stesso per $\hat{f}_k(0.9)$, otteniamo un **diverso** valore ottimale di k.







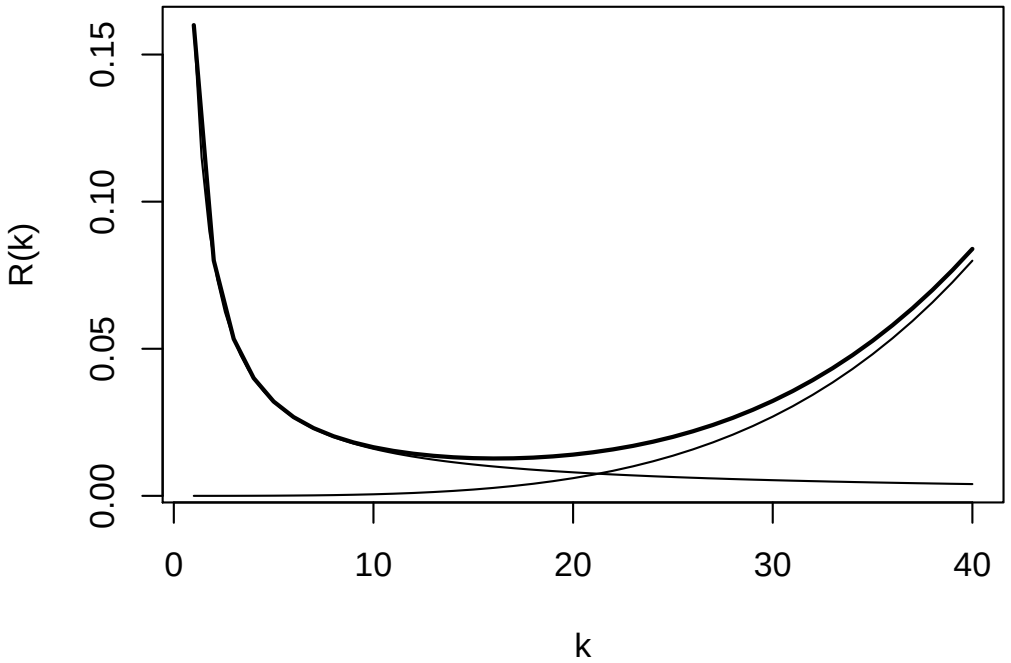
Da $MSE(x)$ all'errore complessivo

Disponiamo dell'MSE per $\hat{f}_k(x)$:

$$MSE(\hat{f}_k(x)) = E((f(x) - \hat{f}_k(x))^2) = (f(x) - E(\hat{f}_k(x)))^2 + V(\hat{f}_k(x))$$

mettiamoli insieme per ottenere un errore complessivo

$$R(k) = E\left(\frac{1}{n}\sum_{i=1}^{n}(\hat{f}_k(x_i) - f(x_i))^2\right)$$



La scelta di k potrebbe basarsi su $R(k)$, ha senso scegliere $k = \text{argmin}_k R(k)$.

Stimatore di $R()$

L'obiettivo è stimare

$$R(k) = E\left(\frac{1}{n}\sum_{i=1}^{n}(\hat{f}_k(x_i) - f(x_i))^2\right)$$

(principalmente per individuare il valore ottimale di k.)

Uno stimatore naïf sarebbe

$$\frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{f}_k(x_i))^2$$

ma questo è ovviamente una sottostima in quanto ...

Stimatore di $R()$: validazione incrociata uno a uno

Uno stimatore migliore per $R(k)$ è

$$CV(k) = \hat{R}(k) = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{f}_{k,-i}(x_i))^2$$

dove $\hat{f}_{k,-i}(x_i)$ è il lisciatore stimato **senza** l'i-esima osservazione.
Si noti che

$$\begin{aligned} E(Y_i - \hat{f}_{k,-i}(x_i))^2 &= E(Y_i - f(x_i) + f(x_i) - \hat{f}_{k,-i}(x_i))^2 \\ &= \sigma^2 + E(f(x_i) - \hat{f}_{h,-i}(x_i))^2 \\ &\approx \sigma^2 + E(f(x_i) - \hat{f}_k(x_i))^2 \end{aligned}$$

Cioè, $\hat{R}$ è, approssimativamente, uno stimatore non distorto dell'errore di previsione

$$E(\hat{R}) \approx R + \sigma^2$$

Lisciatori lineari

Discutiamo della stima dell'errore per una classe di lisciatori che comprende quelli visti sopra e molti altri: i **lisciatori lineari**, ovvero dei lisciatori per i quali esiste, per ogni x, un vettore $\ell(x) = (\ell_1(x), \ldots, \ell_n(x))^T$ tale che

$$\hat{f}(x) = \sum_{i=1}^{n} \ell_i(x) Y_i$$

il che significa che

$$\hat{\mathsf{f}} = \begin{bmatrix} \hat{f}(x_1) \\ \vdots \\ \hat{f}(x_n) \end{bmatrix} = \begin{bmatrix} \ell_1(x_1) & \cdots & \ell_n(x_1) \\ \vdots & & \\ \ell_1(x_n) & \cdots & \ell_n(x_n) \end{bmatrix} \mathsf{Y} = L\mathsf{Y}$$

La matrice L è la **matrice di lisciamento**, si definiscono anche i gradi di libertà del lisciatore come

$$\nu = tr(L)$$

Lisciatori lineari

I precedenti lisciatori sono tutti lisciatori lineari, ricaviamo le matrici L ad essi associati. (Senza perdita di generalità, si assume che le x_i siano ordinate).

- regressogramma: L è diagonale a blocchi e assume valore pari al reciproco del numero di osservazioni in ciascun blocco.
- medie mobili
  - vicini più vicini: L è 0 ovunque tranne che su una striscia intorno alla diagonale dove vale $1/k$.
  - raggio: L è analoga al caso precedente se le x_i sono equidistanziate, altrimenti...

Validazione incrociata uno a uno per lisciatori lineari

Per un lisciatore lineare definito dalla matrice L

$$CV = \hat{R}(k) = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{f}_{k,-i}(x_i))^2 = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{Y_i - \hat{f}_k(x_i)}{1 - L_{ii}}\right)^2$$

sicché non necessita di ricalcolare il lisciatore ma solo di conoscere L_{ii}.

Un'ulteriore semplificazione è costituita dal **criterio di validazione incrociata generalizzata** che prevede di sostituire L_{ii} con il suo valore medio

$$GCV = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{Y_i - \hat{f}(x_i)}{1 - \nu/n}\right)^2$$

Derivazione delle formule per CV e GCV

Definiamo $\hat{f}_{-i}(x_i)$. Essendo

$$\hat{f}(x_i) = \sum_{j=1}^{n} \ell_j(x_i) y_j$$

e assumendo $\sum_{j=1}^{n} \ell_j(x_i) = 1$ (le costanti sono mantenute), definiamo

$$\hat{f}_{-i}(x_i) = \frac{\sum_{j \neq i} \ell_j(x_i) y_j}{\sum_{j \neq i} \ell_j(x_i)} = \frac{\sum_{j \neq i} \ell_j(x_i) y_j}{1 - \ell_i(x_i)} = \frac{\sum_{j \neq i} \ell_j(x_i) y_j}{1 - L_{ii}}$$

Si noti che potremmo definire $\hat{f}_{-i}()$ come il lisciatore ri-stimato senza (x_i, y_i), la definizione è equivalente per lo stimatore a raggio, non per i k più vicini.

Derivazione delle formule per CV e GCV

Con la formula sopra si ottiene

$$
\begin{aligned}
y_i - \hat{y}_{-i} &= y_i - \frac{1}{1 - L_{ii}} \sum_{j \neq i} \ell_j(x_i) y_j \\
&= y_i - \frac{1}{1 - L_{ii}} \left(\sum_{j=1}^{n} \ell_j(x_i) y_j - L_{ii} y_i \right) \\
&= y_i - \frac{1}{1 - L_{ii}} \left(\hat{y}_i - L_{ii} y_i \right) \\
&= \frac{1}{1 - Lii} \left((1 - L_{ii}) y_i - \hat{y}_i + L_{ii} y_i \right) = \frac{1}{1 - Lii} (y_i - \hat{y}_i)
\end{aligned}
$$

e quindi la formula del GCV.

Altri criteri

Si noti che, essendo $(1-x)^{-2} \approx 1+2x$ in un intorno di 0, il GCV è approssimativamente uguale al C_p di Mallow.

$$GCV = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{Y_i - \hat{f}(x_i)}{1-\nu/n}\right)^2 \approx \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{f}(x_i)\right)^2 + \frac{2\nu\hat{\sigma}^2}{n} = C_p$$

Più in generale, molti criteri usati per la scelta del grado di lisciamento (k) hanno la forma

$$B(k) = \Lambda(n,k)\frac{1}{n} + \sum_{i=1}^{n}\left(Y_i - \hat{f}(x_i)\right)^2$$

per qualche funzione $\Lambda(\cdot,\cdot)$

Indice

Kernel regression

Con i metodi descritti sin qui, man mano che ci si muove lungo l'asse x, si calcola $\hat{f}(x)$ come media di differenti gruppi di osservazioni y_i.

Questo porta a una stima finale poco "liscia".

Un modo di lisciare maggiormente è di usare una media pesata dove il peso delle osservazioni decresce man mano che ci si allontana da x.

Stimatore di Nadaraya-Watson

Lo stimatore di Nadaraya-Watson è un lisciatore lineare

$$\hat{f}(x) = \sum_{i=1}^{n} \ell_i(x) Y_i$$

in cui

$$\ell_i(x) = \frac{K\left(\frac{x-x_i}{h}\right)}{\sum_{j=1}^{n} K\left(\frac{x-x_j}{h}\right)}$$

dove $K()$ è un nucleo.

Nuclei

Si ha

$$\hat{f}_n(x) = \frac{\sum_{i=1}^n K\left(\frac{x-X_i}{h}\right) Y_i}{\sum_{i=1}^n K\left(\frac{x-X_i}{h}\right)}$$

dove K è tale che

- $K(x) \geq 0$
- $\int K(x)dx = 1$
- $\int xK(x)dx = 0$
- $\int x^2K(x)dx > 0$

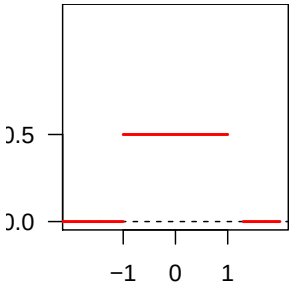
Esempi di nuclei

	$K(u)$
Uniform	$\frac{1}{2} I_{[-1,1]}(u)$
Triangle	$(1-\lvert u\rvert) I_{[-1,1]}(u)$
Triweight	$\frac{35}{32}(1-u^2)^3 I_{[-1,1]}(u)$
Quartic	$\frac{15}{16}(1-u^2)^2 I_{[-1,1]}(u)$
Gaussian	$\frac{1}{\sqrt{2\pi}} e^{-u^2/2}$
Epanechnikov	$\frac{3}{4}(1-u^2) I_{[-1,1]}(u)$
Cosine	$\frac{\pi}{4}\cos\left(\frac{\pi}{2}u\right) I_{[-1,1]}(u)$

Kernel functions

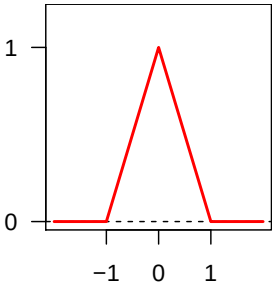
Uniform

$\frac{1}{2} I_{[-1,1]}(u)$



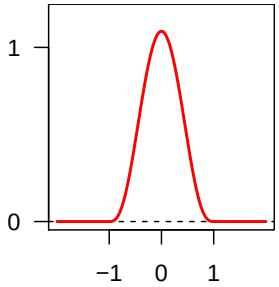
Triangle

$(1 - |u|) I_{[-1,1]}(u)$



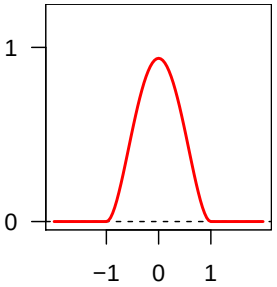
Triweight

$\frac{35}{32}(1 - u^2)^3 I_{[-1,1]}(u)$



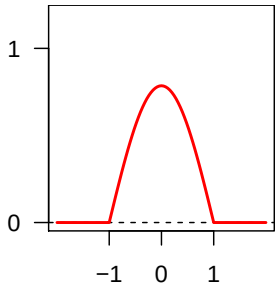
Quartic

$\frac{15}{16}(1 - u^2)^2 I_{[-1,1]}(u)$



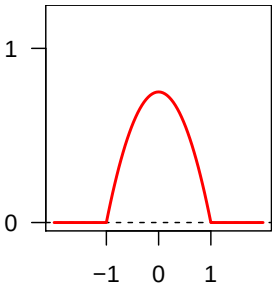
Cosine

$\frac{\pi}{4} \cos\left(\frac{\pi}{2} u\right) I_{[-1,1]}(u)$



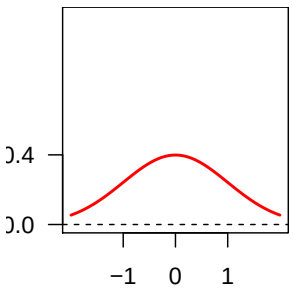
Epanechnikov

$\frac{3}{4}(1 - u^2) I_{[-1,1]}(u)$



Gaussian

$\frac{1}{\sqrt{2\pi}} e^{-u^2/2}$



Nadaraya-Watson estimator: risk

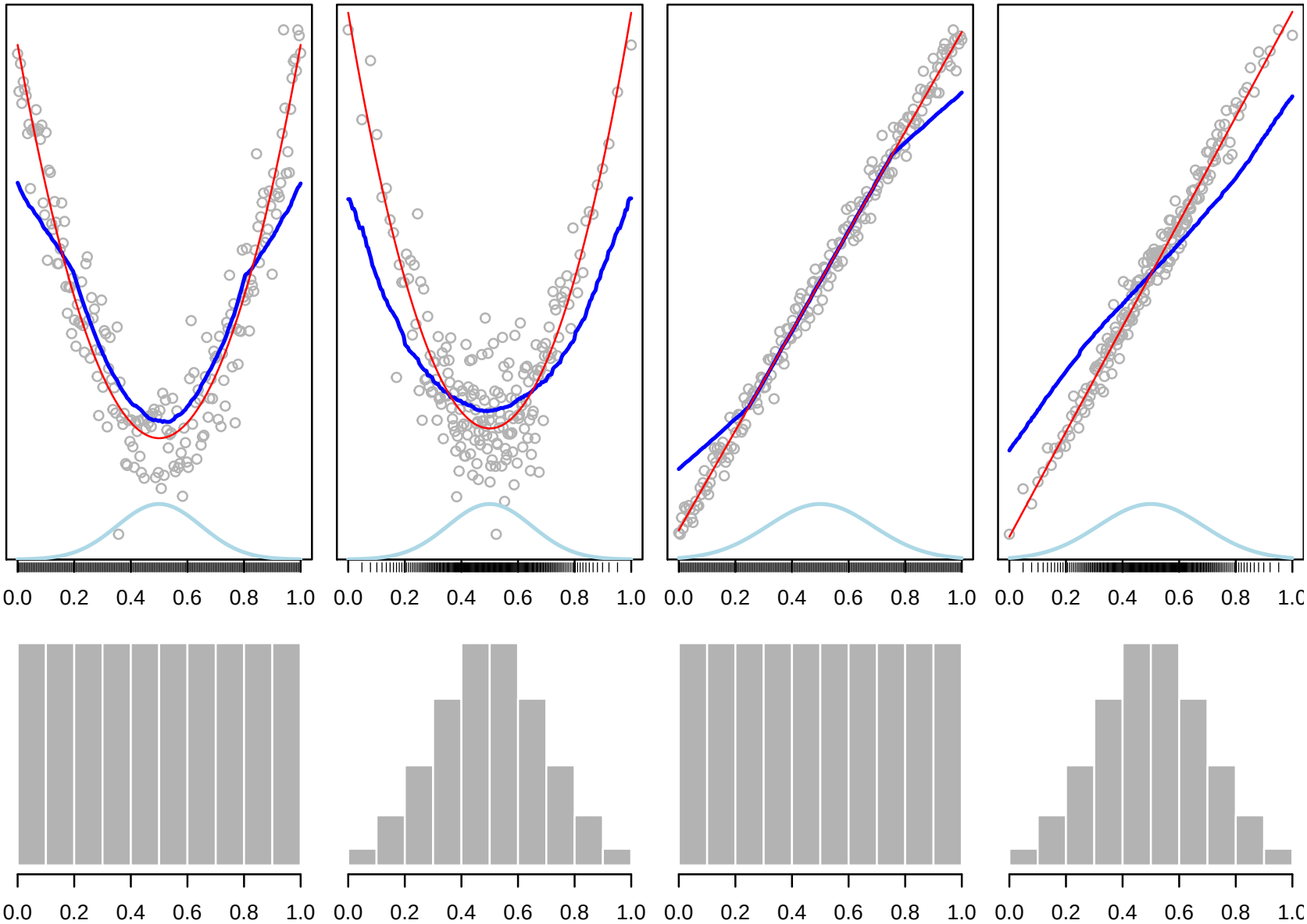
Si mostra che, se x_i proviene dalla densità $g()$, per $h_n \to 0$ e $nh_n \to \infty$

$$R = \frac{h_n^4}{4}\left(\int u^2 K(u)du\right)^2 \int \left(f''(x) + 2f'(x)\frac{g'(x)}{g(x)}\right)^2 dx + \frac{\sigma^2 \int K^2(u)du}{nh_n}\int \frac{1}{g(x)}dx + o(nh_n^{-1}) + o(h_n^4)$$

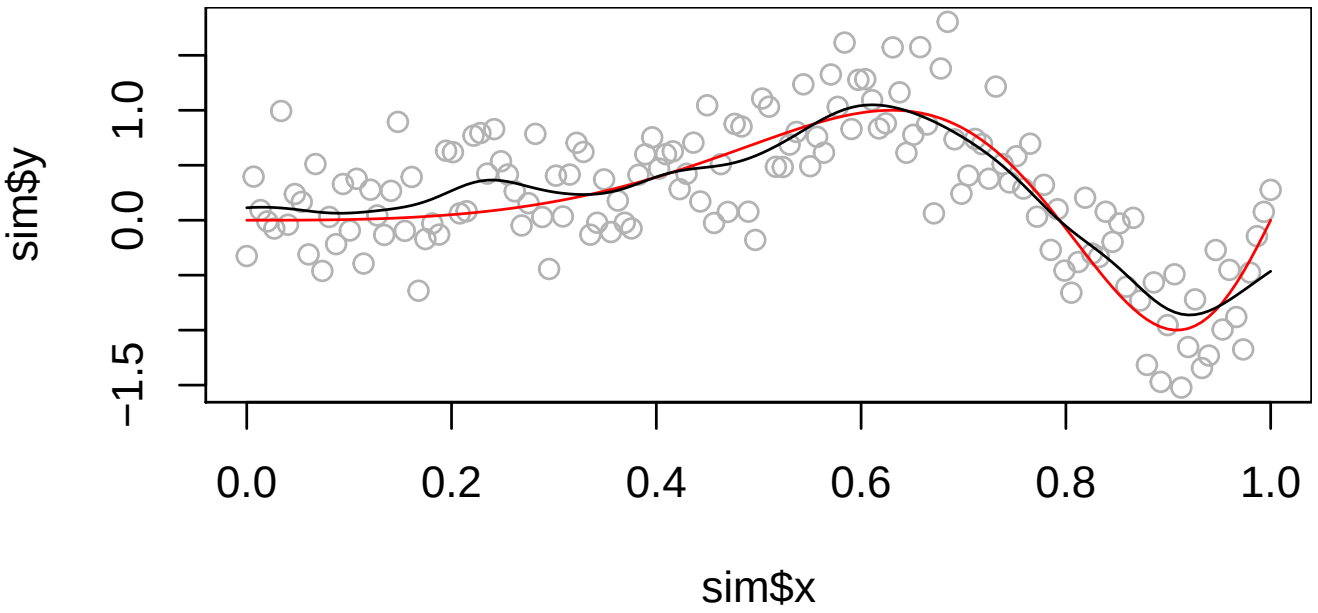
Dove si nota che

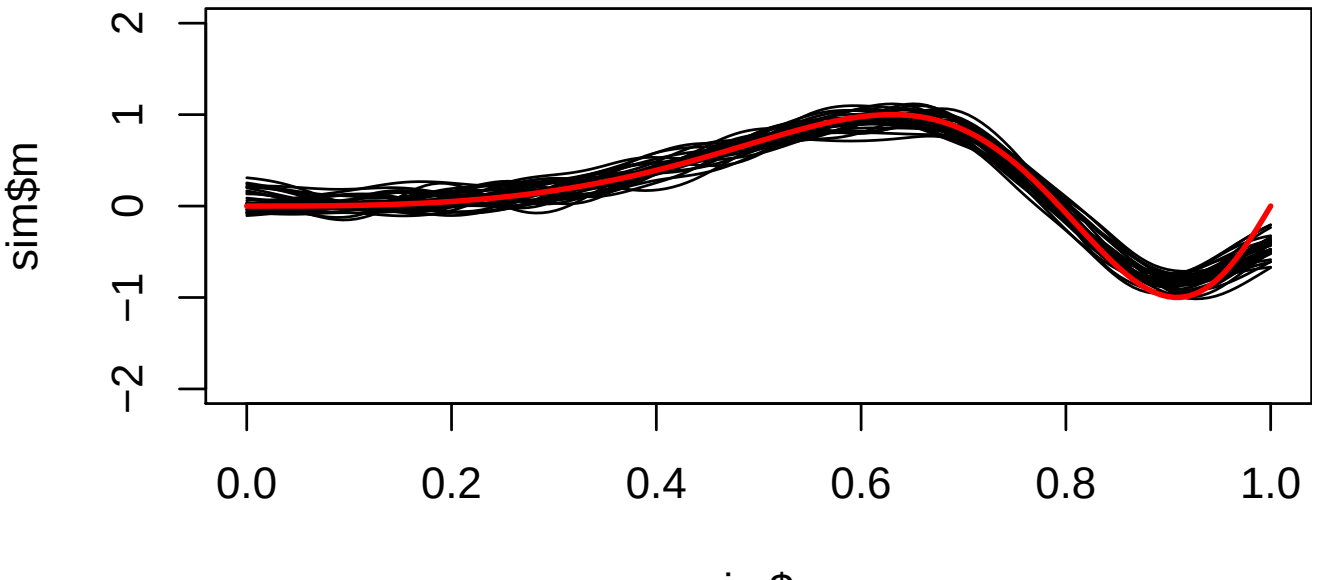
- la varianza decresce con h
- la distorsione cresce con h^4
- la distorsione cresce con f''
- la distorsione cresce con $f'(x)\frac{g'(x)}{g(x)}$: *design bias*

Design bias e boundary bias



Boundary bias





N-W come mimimo

Notiamo che lo stimatore di N-W in x, $\hat{f}(x)$, è la soluzione di

$$\underset{a}{\operatorname{argmin}} \sum_{i=1}^{n} K_i \left(\frac{x_i - x}{h} \right) (Y_i - a)^2$$

cioè, lo stimatore di N-W è, localmente, uno stimatore dei minimi quadrati pesati.

Si potrebbe allora impiegare i minimi quadrati pesati ma con un polinomio anzichè una costante, per ogni valore di x si approssima $f()$ in un intorno di x con il polinomio

$$p_x(u; \mathrm{a}) = a_0 + a_1(u - x) + \frac{a_2}{2!}(u - x)^2 + \ldots + \frac{a_p}{p!}(u - x)^p$$

N-W come mimimo $\rightarrow$ polinomi locali

Si potrebbe allora impiegare i minimi quadrati pesati ma con un polinomio anzichè una costante, per ogni valore di x si approssima $f()$ in un intorno di x con il polinomio

$$p_x(u; \mathrm{a}) = a_0 + a_1(u - x) + \frac{a_2}{2!}(u - x)^2 + \ldots + \frac{a_p}{p!}(u - x)^p$$

e si stima $\mathrm{a}(x)$ (rendiamo esplicita la dipendenza da x) minimizzando

$$\hat{\mathrm{a}}(x) = \underset{a}{\operatorname{argmin}} \sum_{i=1}^{n} K_i \left(\frac{x_i - x}{h} \right) (Y_i - p_x(X_i; \mathrm{a}))^2$$

e definiamo il seguente stimatore di $f(x)$

$$\hat{f}(x) = p_x(x, \hat{\mathrm{a}}) = \hat{a}_0(x)$$

Polinomi locali, notazione matriciale

Sia

$$X_x = \begin{bmatrix} 1 & x_1 - x & \cdots & \frac{1}{p!}(x_1 - x)^p \\ \vdots & \vdots & & \vdots \\ 1 & x_n - x & \cdots & \frac{1}{p!}(x_n - x)^p \end{bmatrix}$$

$$W_x = \text{diag}\left\{ K_i\left(\frac{x_i - x}{h}\right), i = 1, \ldots, n \right\}$$

allora la somma dei quadrati pesata è

$$(\mathrm{Y} - X_x \mathrm{a})^T W_x (\mathrm{Y} - X_x \mathrm{a})$$

e

$$\hat{\mathrm{a}} = (X_x^T W_x X_x)^T X_x^T W_x \mathrm{Y}$$

Polinomi locali, notazione matriciale

$$\hat{a} = (X_x^T W_x X_x)^T X_x^T W_x Y$$

Lo stimatore $\hat{f}(x) = \hat{a}_0(x)$ è dunque

$$\hat{f}(x) = e_1^T (X_x^T W_x X_x)^T X_x^T W_x Y$$

dove $e_1^T = (1, 0, \ldots, 0)$.
Quindi $\hat{f}(x)$ è un lisciatore lineare

$$\hat{f}(x) = \sum_{i=1}^{n} \ell_i(x) Y_i$$

dove

$$\ell(x)^T = (\ell_1(x), \ldots, \ell_n(x))^T = e_1^T (X_x^T W_x X_x)^T X_x^T W_x$$

Lisciatore lineare locale

Posto $p = 1$, si ottiene lo stimatore lineare locale

$$\ell_i(x) = \frac{b_i(x)}{\sum_{j=1}^{n} b_j(x)}$$

dove

$$b_i(x) = K\left(\frac{x_i - x}{h}\right)(S_{n,2}(x) - (x_i - x)S_{n,1}(x))$$

$$S_{n,j}(x) = \sum_{i=1}^{n} K\left(\frac{x_i - x}{h}\right)(x_i - x)^j, \quad j = 1, 2$$

Lisciatore lineare locale: distorsione e varianza

Si mostra che il rischio in x è

$$R_x = \frac{h_n^4}{4}\left(\int u^2 K(u)du\right)^2 f''(x)^2 + \frac{\sigma^2 \int K^2(u)du}{g(x)nh_n} + o(nh_n^{-1}) + o(h_n^4)$$

Lisciatore lineare locale: distorsione e varianza

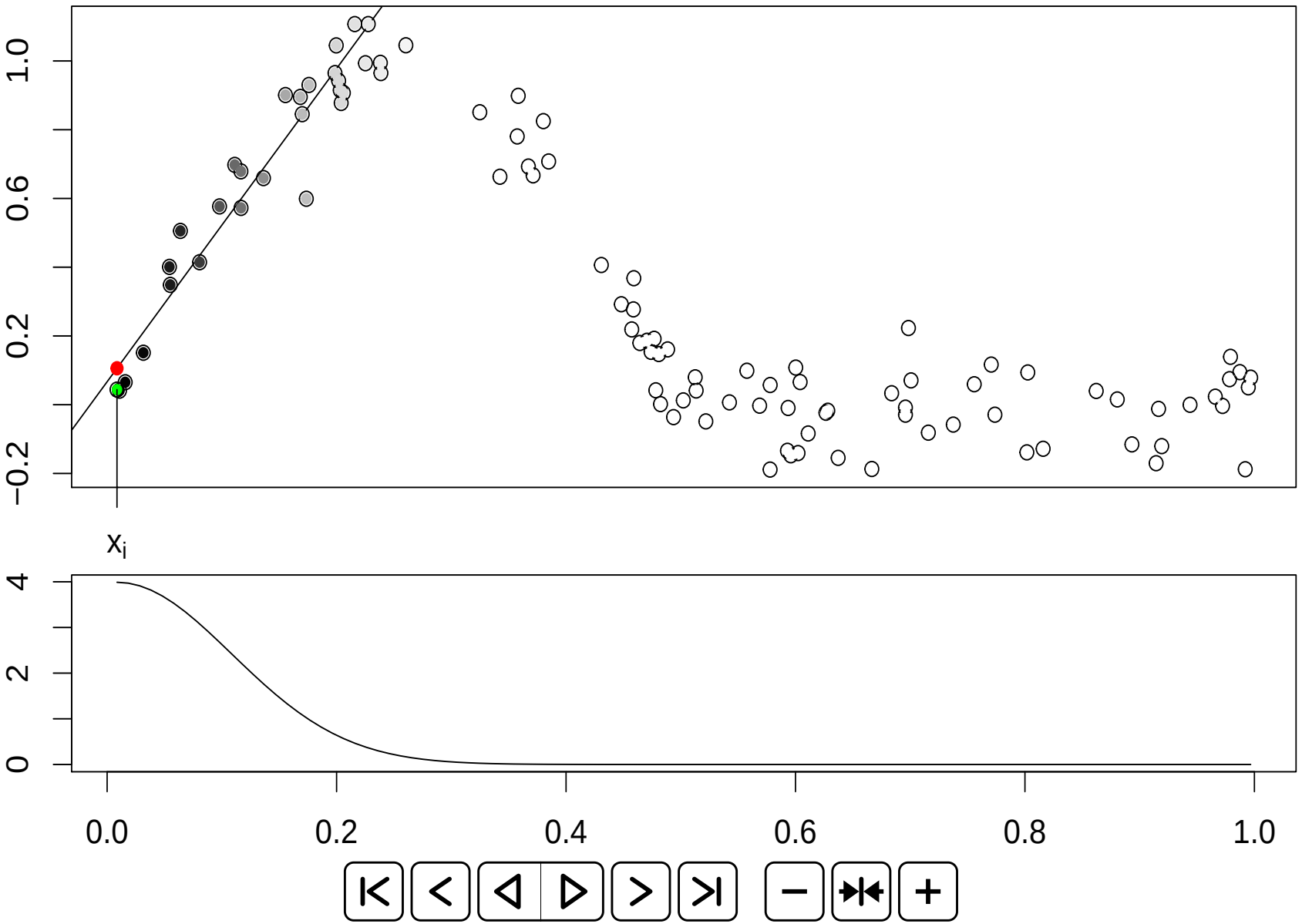
Si mostra che il rischio in x è

$$R_x = \frac{h_n^4}{4}\left(\int u^2 K(u)du\right)^2 f''(x)^2 + \frac{\sigma^2 \int K^2(u)du}{g(x)nh_n} + o(nh_n^{-1}) + o(h_n^4)$$

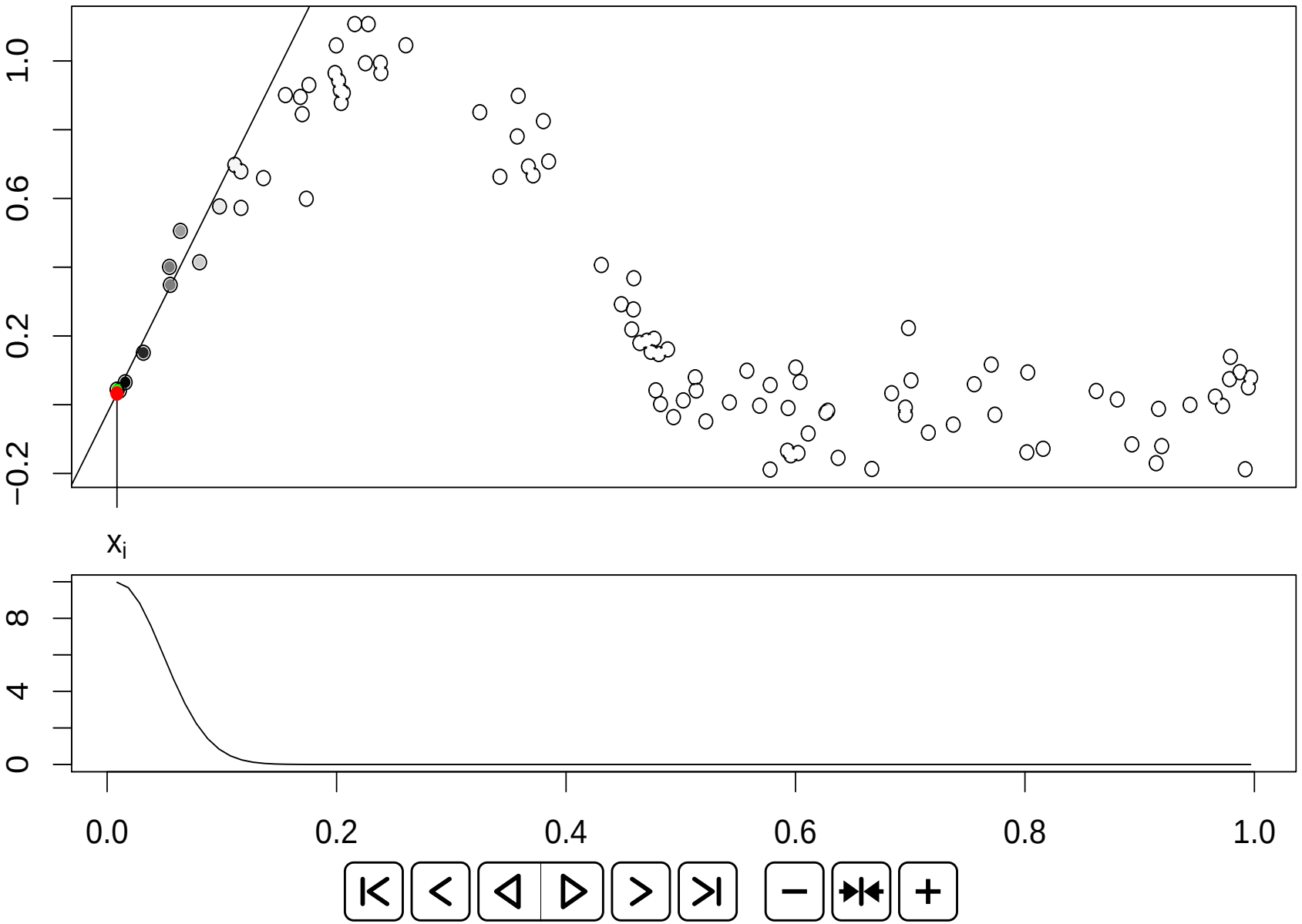
Se lo confrontiamo con quello dello stimatore di N-W notiamo che è scomparso il *design bias*.

$$\begin{aligned} R =& \frac{h_n^4}{4}\left(\int u^2 K(u)du\right)^2 \int \left(f''(x) + 2f'(x)\frac{g'(x)}{g(x)}\right)^2 dx \\ &+ \frac{\sigma^2 \int K^2(u)du}{nh_n}\int \frac{1}{g(x)}dx + o(nh_n^{-1}) + o(h_n^4) \end{aligned}$$

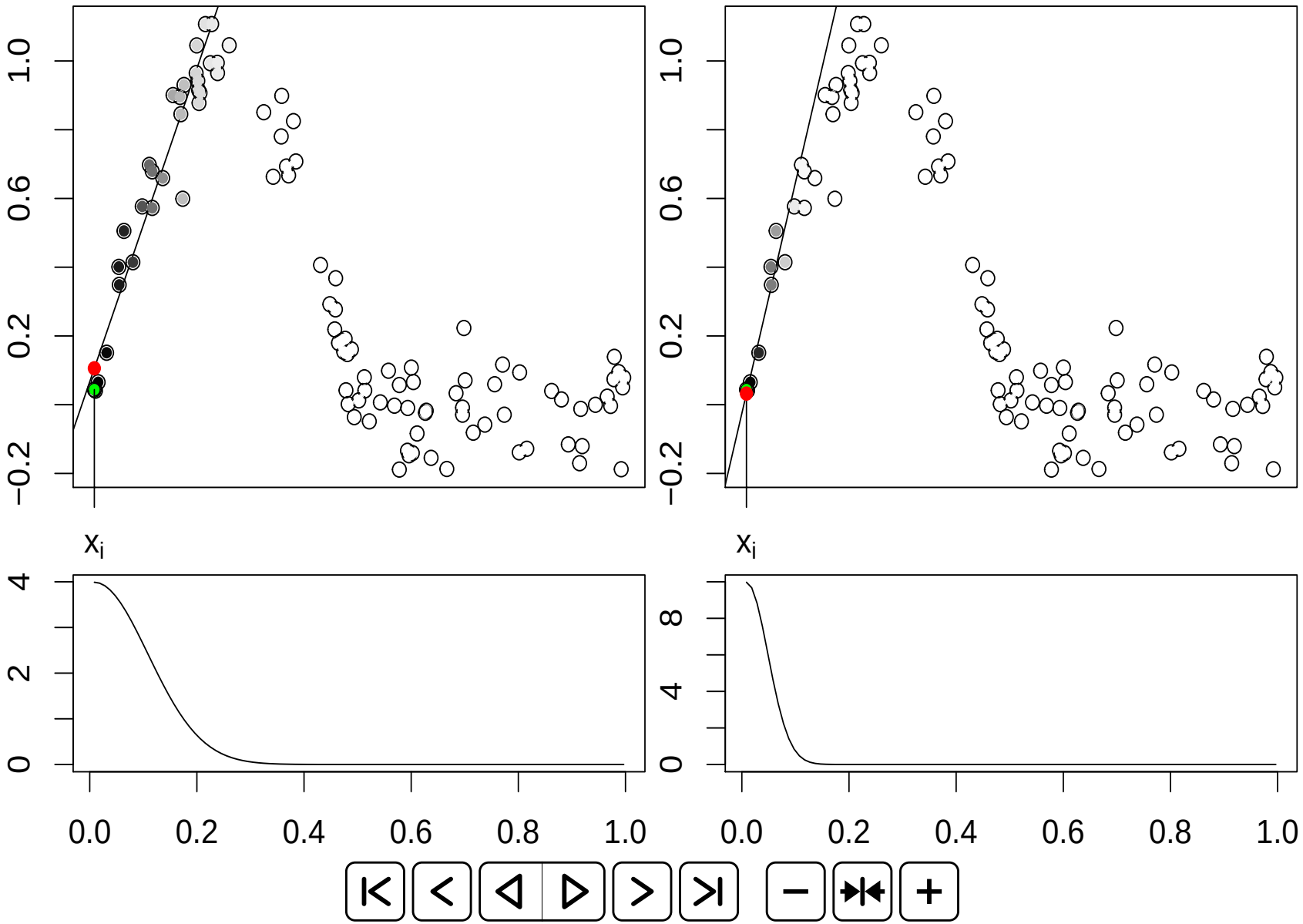
Graphical representation



Graphical representation

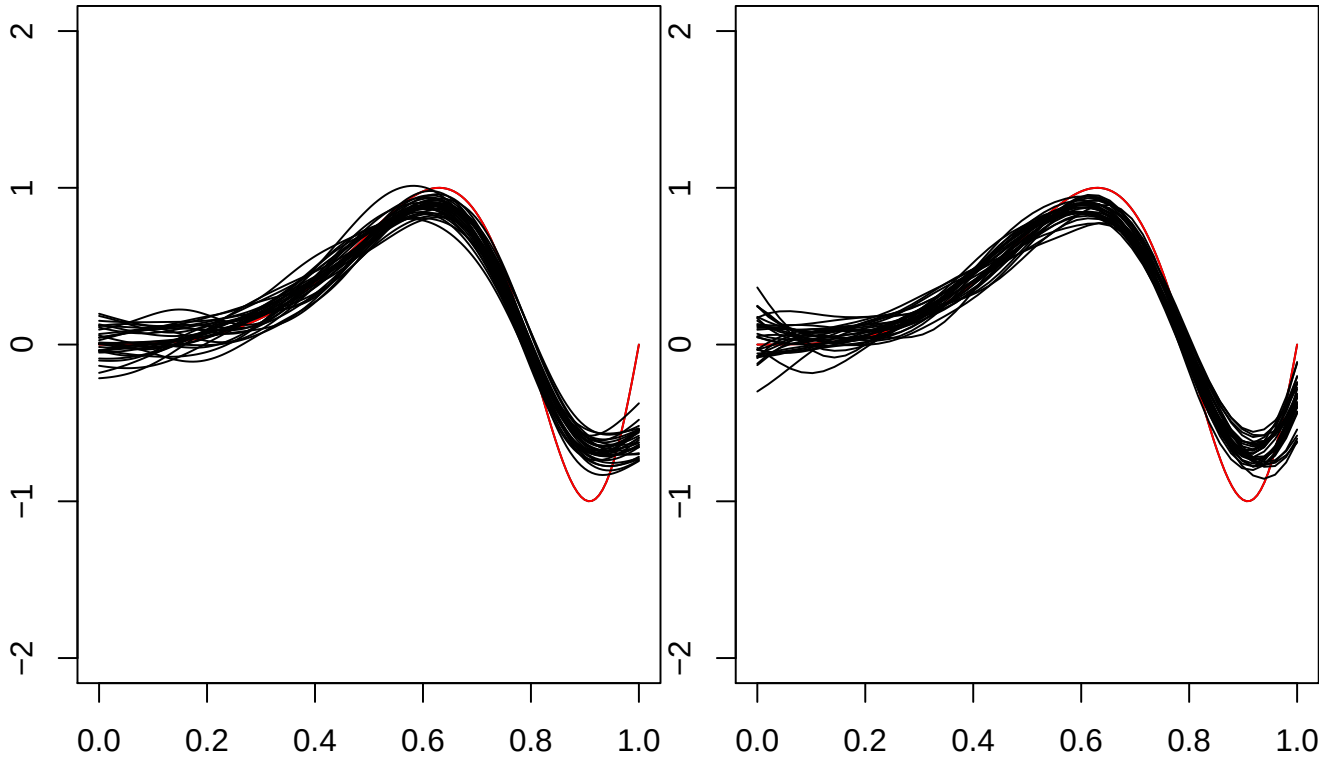


Graphical representation: comparison

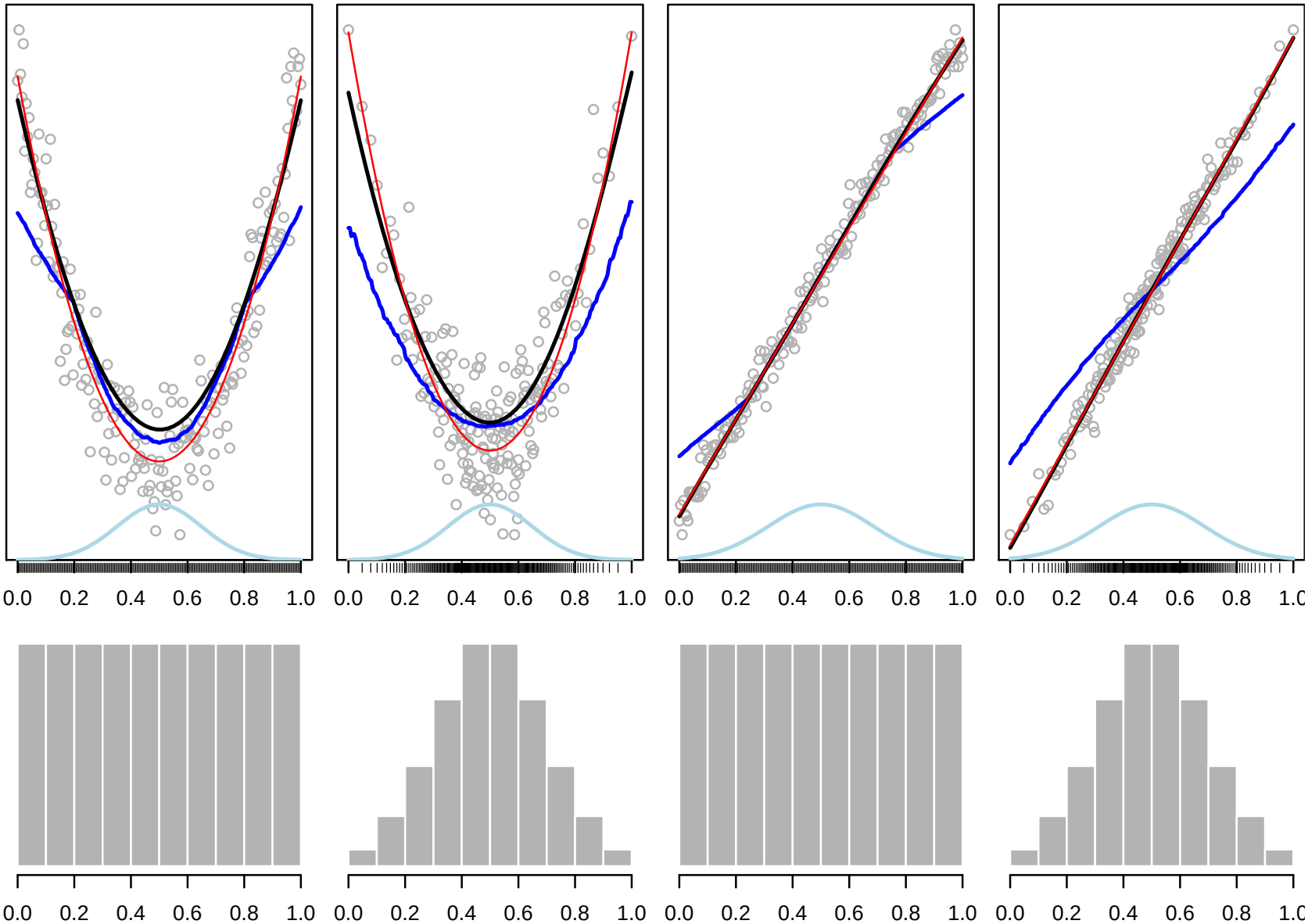


Boundary bias

N-W versus local linear, same bandwith



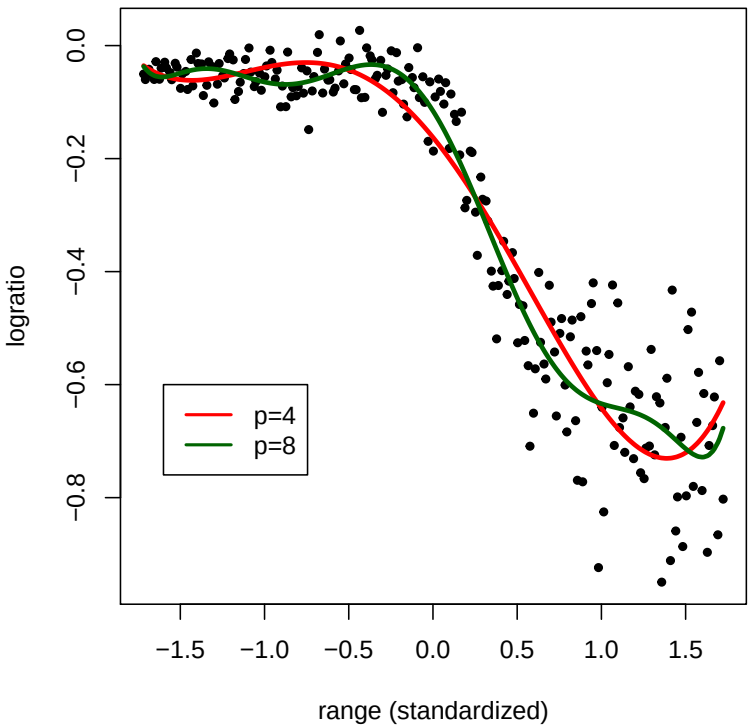
Design bias and boundary bias



LIDAR: modello polinomiale

Assumiamo che $f()$ sia (approssimabile da) un polinomio

$$f(x;\beta) = \beta_0 + \beta_1 x + \beta_2 x^2 + \ldots + \beta_p x^p$$



Il ruolo di k è svolto da p, al crescere di p

- aumenta la varianza
- diminuisce la distorsione

Problema: elevata correlazione dei $\hat{\beta}_j$

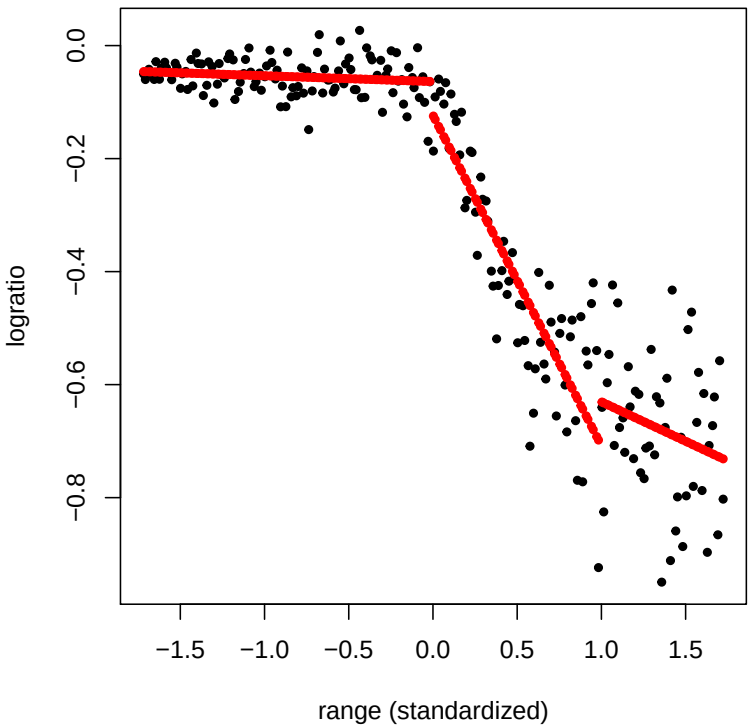
	x	I(x^2)	I(x^3)	I(x^4)
x	1.00	-0.00	-0.92	0.00
I(x^2)	-0.00	1.00	0.00	-0.96
I(x^3)	-0.92	0.00	1.00	-0.00
I(x^4)	0.00	-0.96	-0.00	1.00

LIDAR: modello lineare a tratti

In alternativa, si potrebbe usare un modello lineare a tratti

$$f(x;\boldsymbol{\beta}) = \beta_{0,j} + \beta_{1,j}x \text{ se } c_{j-1} \leq x < c_j,\ j = 1,\ldots,J$$

avendo suddiviso il supporto di x in intervalli (cfr costante a tratti).



Il ruolo di k è svolto da J, al crescere di J

- aumenta la varianza
- diminuisce la distorsione

Problema: la funzione che si ottiene non è continua.

Indice

Spline lineare: esempio con due nodi

Una soluzione più sofisticata si ottiene con

$$y_i = \beta_1 + \beta_2 x_i + \beta_3 (x_i - \nu_1)_+ + \beta_4 (x_i - \nu_2)_+ + \varepsilon_i$$

dove

$$(x)_+ = \begin{cases} x & \text{se } x > 0 \\ 0 & \text{altrimenti} \end{cases}; \quad (x - \nu)_+ = \begin{cases} x - \nu & \text{se } x > \nu \\ 0 & \text{altrimenti} \end{cases}$$

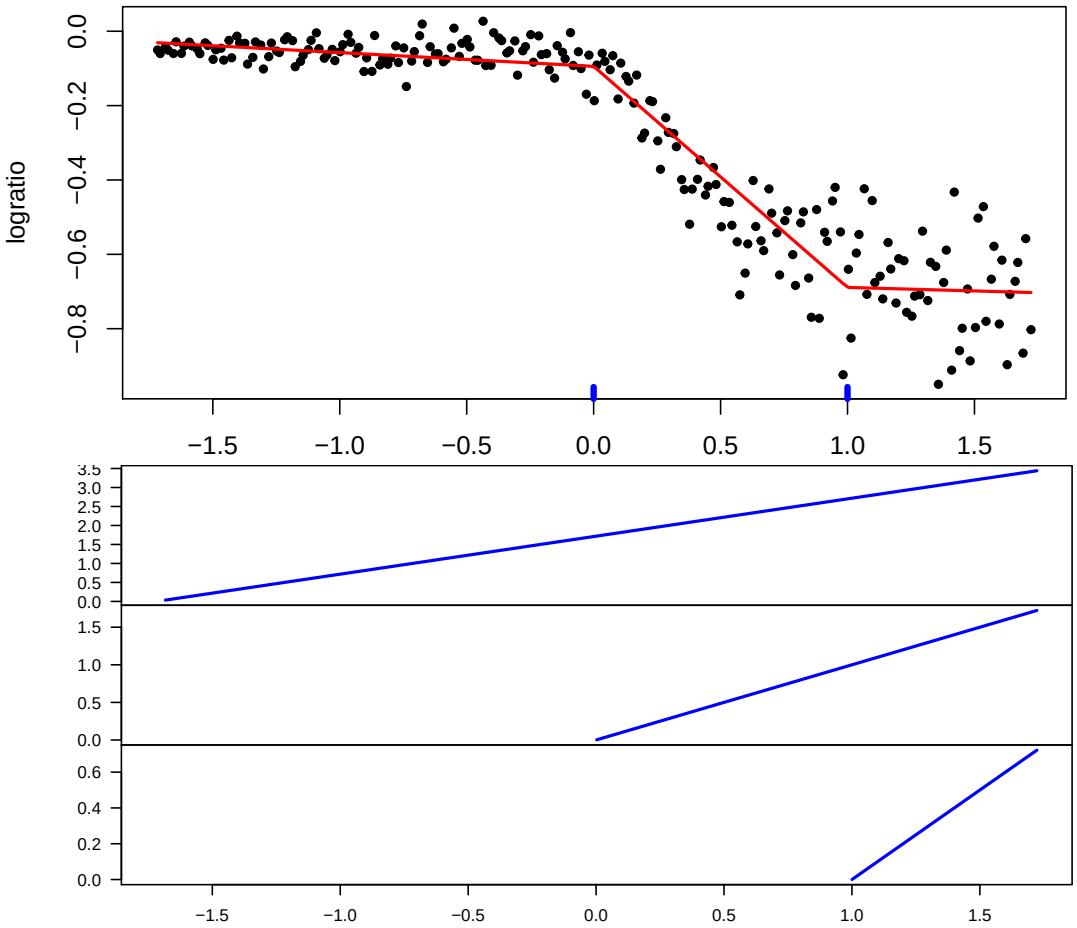
- $\varepsilon_i \sim IID(\mathcal{N}(0, \sigma^2))$,
- ν_1 e ν_2, detti nodi, sono valori fissati nel supporto di x
- β_i sono stimati come al solito:

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\text{argmin}} \sum_{i=1}^{n} (y_i - f(x_i; \boldsymbol{\beta}))^2$$

dove

$$f(x_i; \boldsymbol{\beta}) = \beta_1 + \beta_2 x_i + \beta_3 (x_i - \nu_1)_+ + \beta_4 (x_i - \nu_2)_+$$

Spline lineare: esempio con due nodi



È un modello lineare con esplicative

$$x_i,\ (x_i - \nu_1)_+,\ (x_i - \nu_2)_+$$

La funzione $\hat{f}(x)$ è una combinazione lineare delle funzioni

$$B_0(x) = x$$
$$B_1(x) = (x - \nu_1)_+$$
$$B_2(x) = (x - \nu_2)_+$$

dette funzioni base (in blu nel grafico).

Spline lineare: K nodi

Più in generale, si fissano K nodi

$$\nu_1, \ldots, \nu_K$$

e si stima il modello lineare (attenzione alla notazione!)

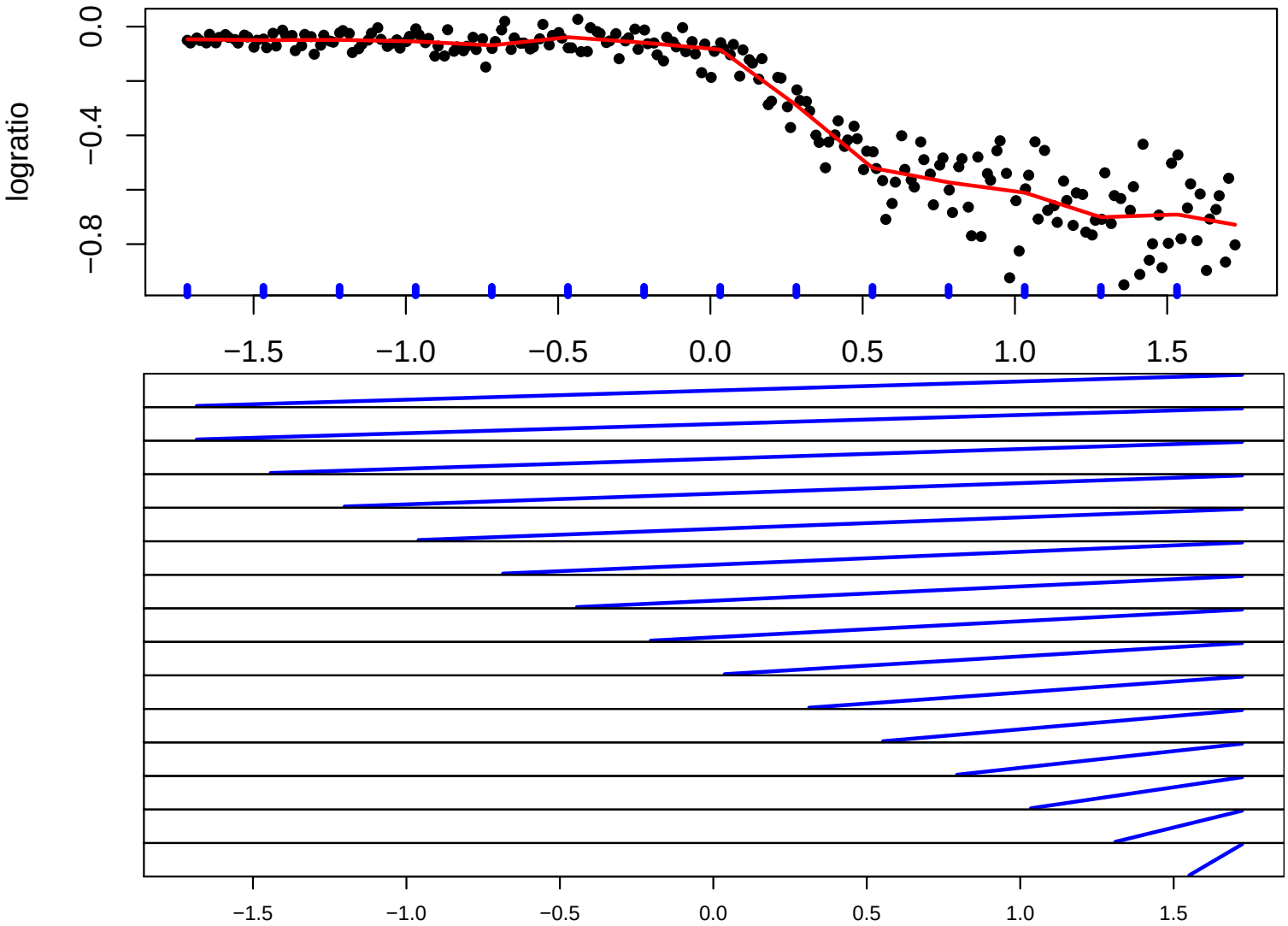
$$y_i = \beta_1 + \beta_2 x_i + \sum_{k=1}^{K} b_k (x_i - \nu_k)_+ + \varepsilon_i$$

La funzione spline è rappresentata da

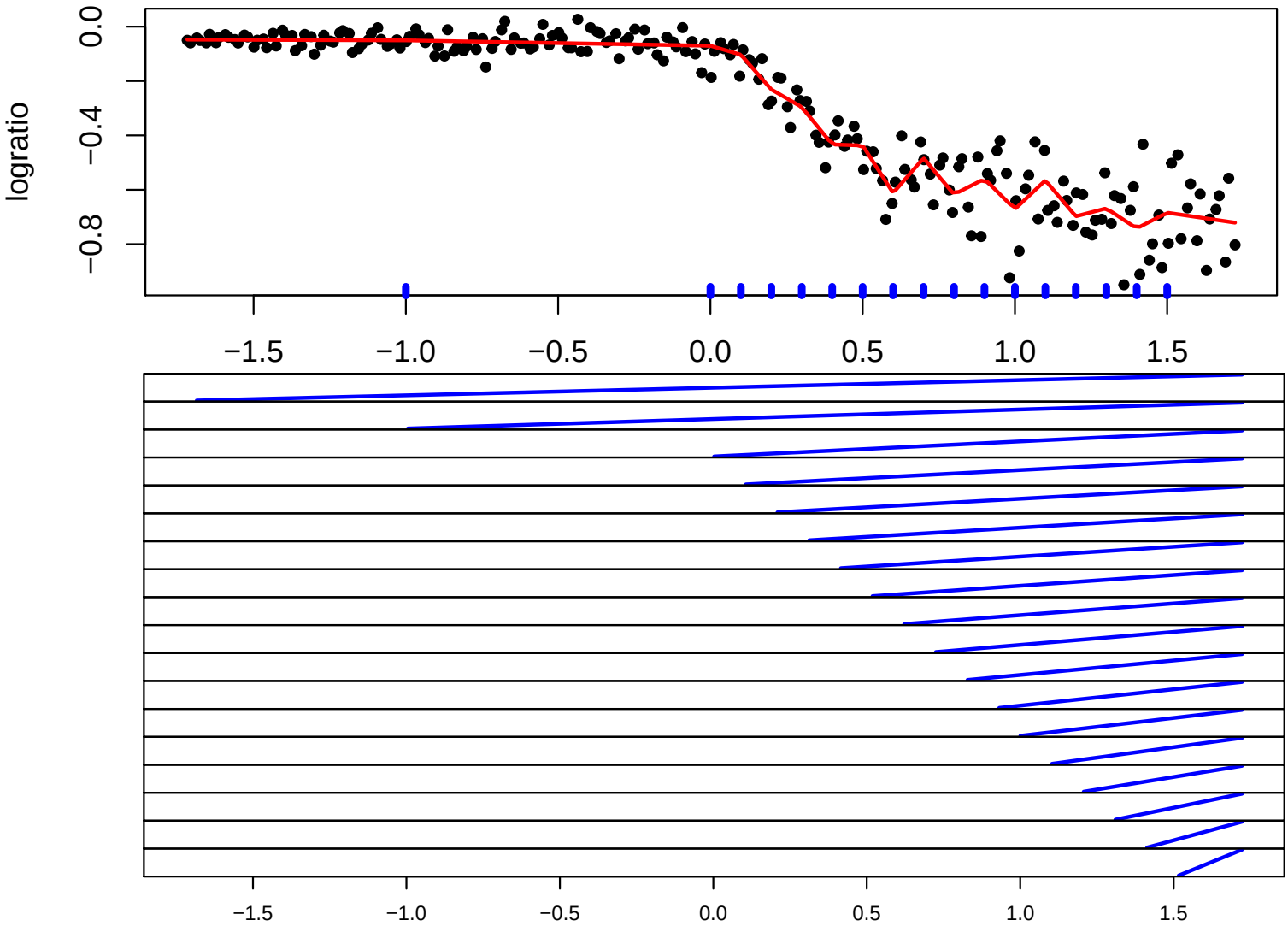
$$f(x) = \beta_1 + \beta_2 x + \sum_{k=1}^{K} b_k (x - \nu_k)_+$$

ed è tanto più liscia quanti meno nodi si usano (minore K).

Spline lineare: K nodi



Spline lineare: K nodi



Base con potenze troncate

Una naturale estensione della base lineare è data dalle potenze

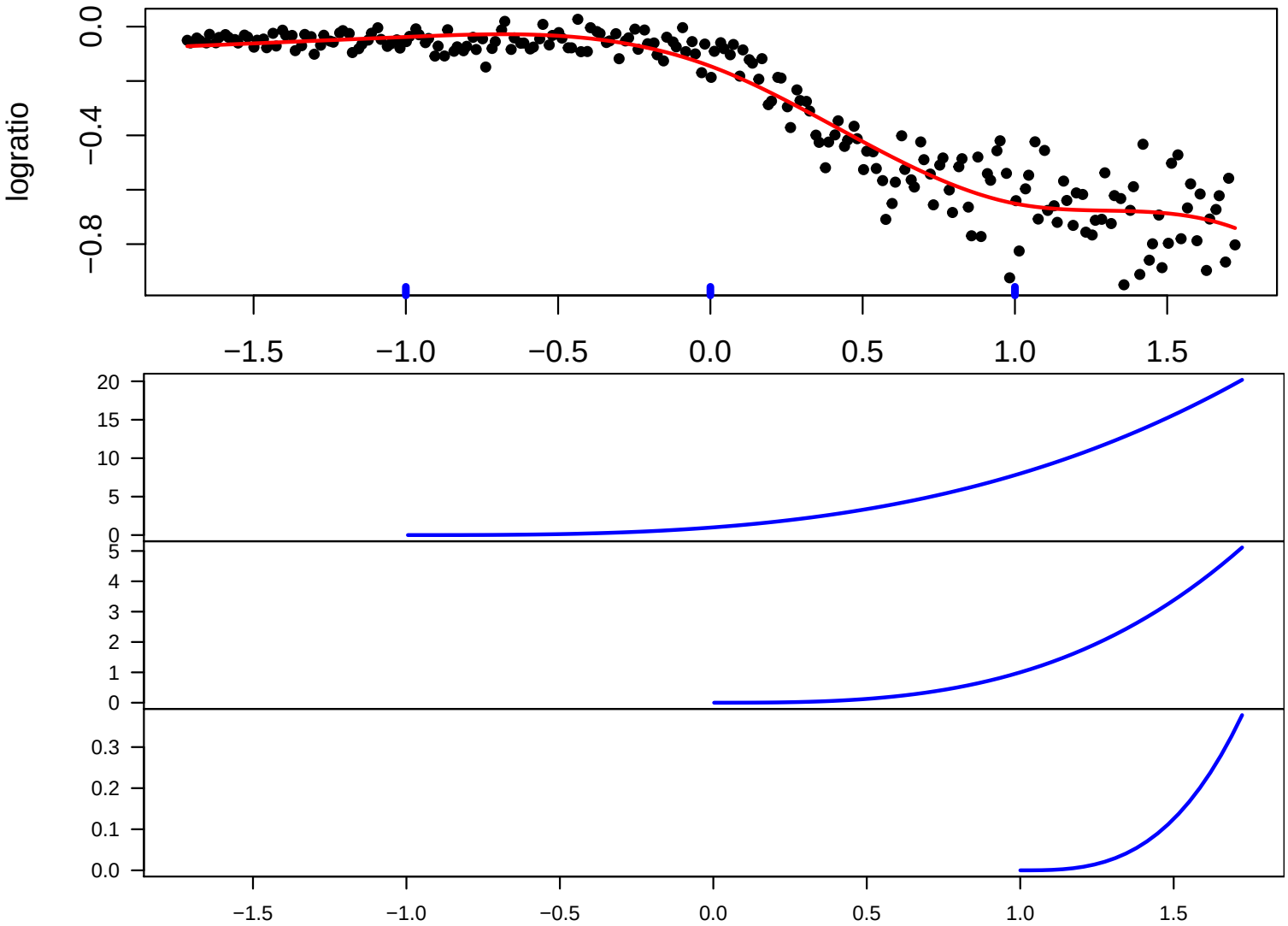
$$y_i = \beta_1 + \beta_2 x_i + \ldots + \beta_{p+1} x^p + \sum_{k=1}^{K} b_k (x_i - \nu_k)_+^p + \varepsilon_i$$

sicché la spline di ordine p con K nodi è

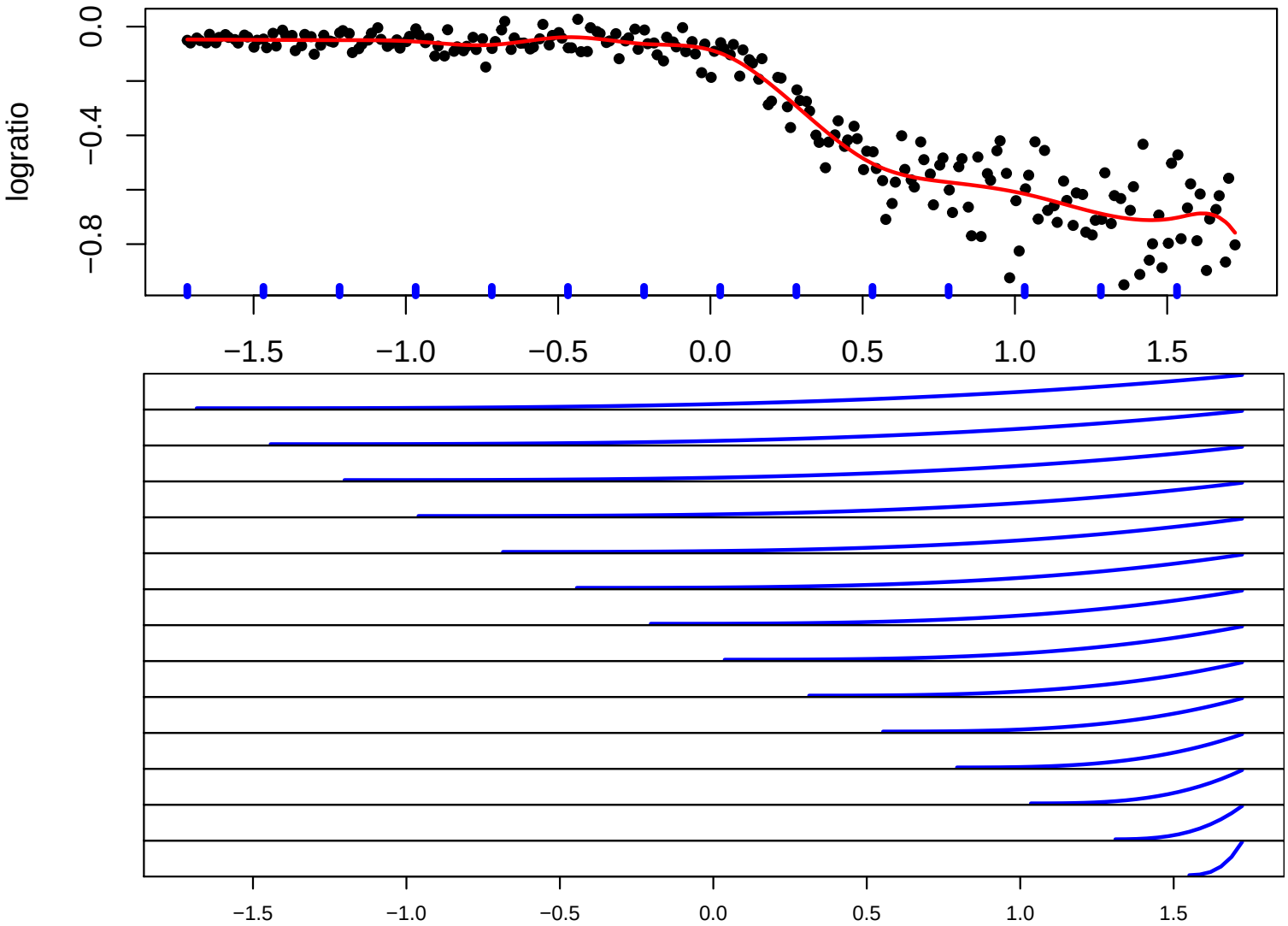
$$f(x) = \beta_1 + \beta_2 x + \ldots + \beta_{p+1} x^p + \sum_{k=1}^{K} b_k (x - \nu_k)_+^p$$

- Una spline di grado p ha $p - 1$ derivate continue,
- $p = 3$ è adeguato per gli scopi usuali.

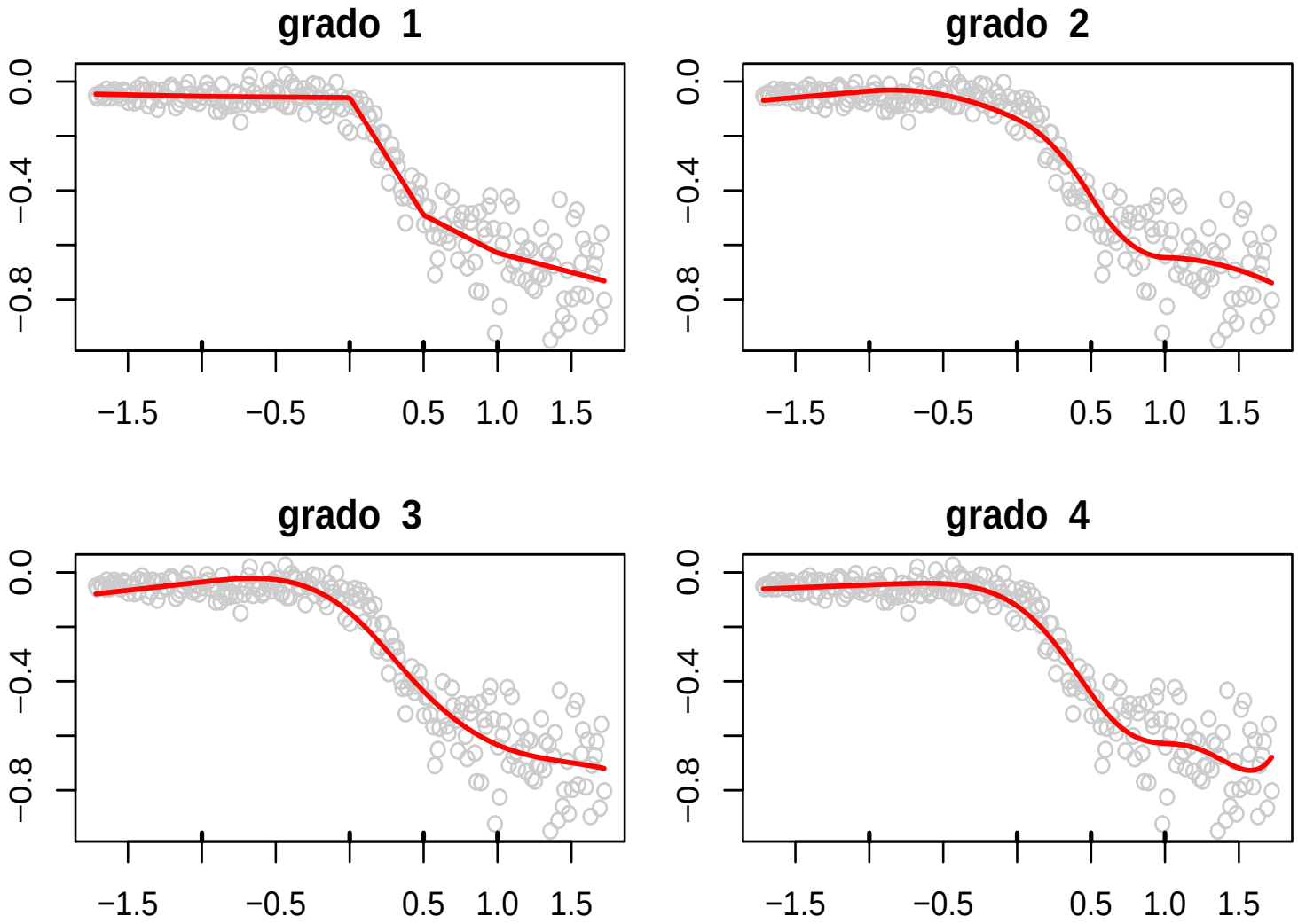
Base con potenze troncate



Base con potenze troncate



TPB: diversi gradi



Levigatezza (smoothness) della spline e numero di nodi

Per quanto visto sin qui, la spline è più o meno liscia (meno o più flessibile) a seconda del numero di nodi (fissato il grado):

- 0 nodi: si riduce a un polinomio di grado p;
- al crescere del numero di nodi, la funzione è sempre più flessibile (meno liscia);
- tanti nodi quante le osservazioni distinte: $\hat{f}(x)$ interpola i punti esattamente;
- più nodi delle osservazioni distinte: il modello non è identificato.

(Si noti anche che la posizione dei nodi determina in quali regioni la spline è più o meno liscia.)

D'altra parte, più sono i nodi, più sono i parametri da stimare, quindi la maggior flessibilità si "paga" in temrini di variabilità degli stimatori.

Levigatezza della spline e numero di nodi: distorsione v. varianza

La scelta del numero di nodi ha un ruolo analogo alla scelta del numero k di vicini più vicini, implicando un *trade off* tra distorsione e varianza

- più nodi $\leftrightarrow$ meno liscia $\leftrightarrow$ meno dist. più varianza;
- meno nodi $\leftrightarrow$ più liscia $\leftrightarrow$ più dist. meno varianza;

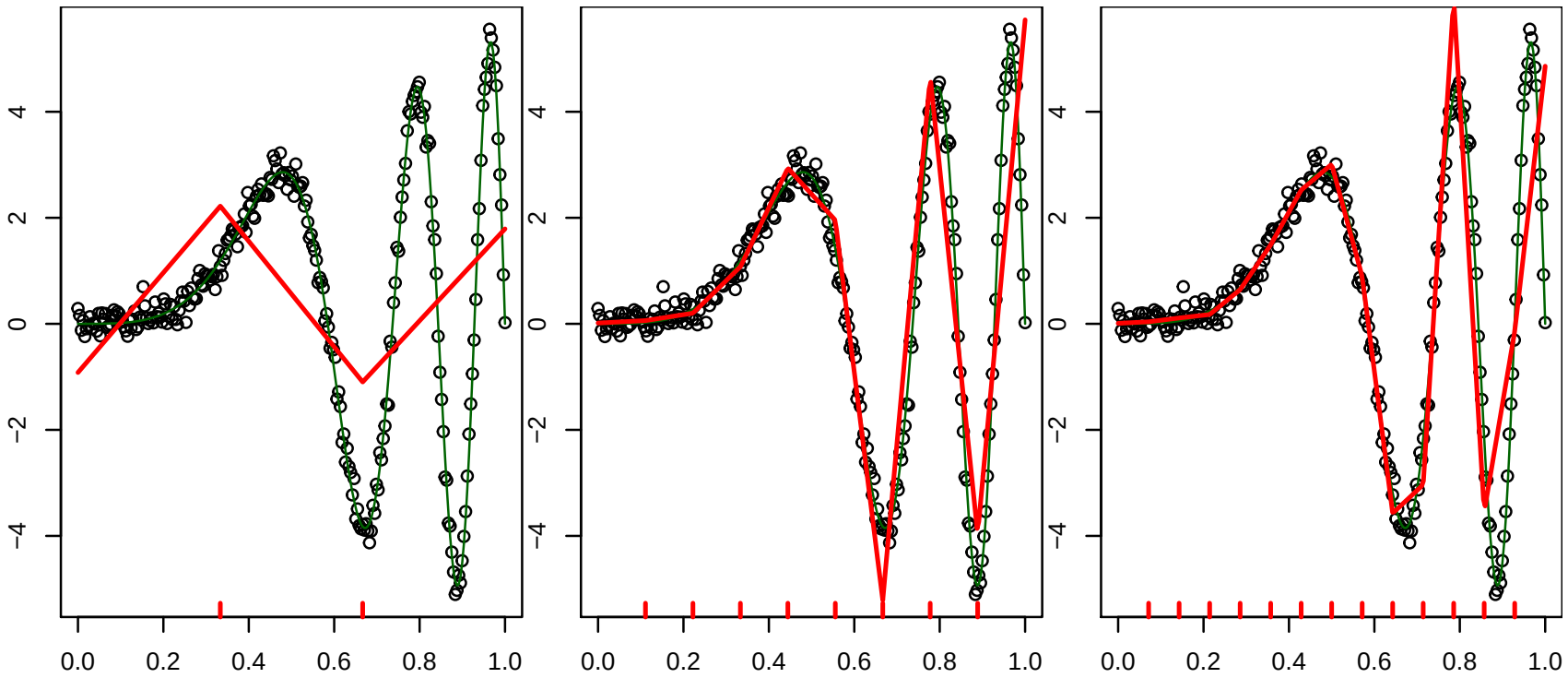
La scelta del grado di levigatezza della spline è cruciale.

In linea di principio, potremmo individuare il livello ottimale di levigatezza scegliendo il numero di nodi che minimizza l'errore quadratico medio: occcorrerebbe stimare funzioni spline corrispondenti a diverse scelte sul numero di nodi (glissiamo sul problema della loro posizione) e calcolare/stimare per ciascuna l'MSE.

Questa strategia è ragionevole ma complessa dal punto di vista numerico, nel seguito opteremo per un'alternativa.

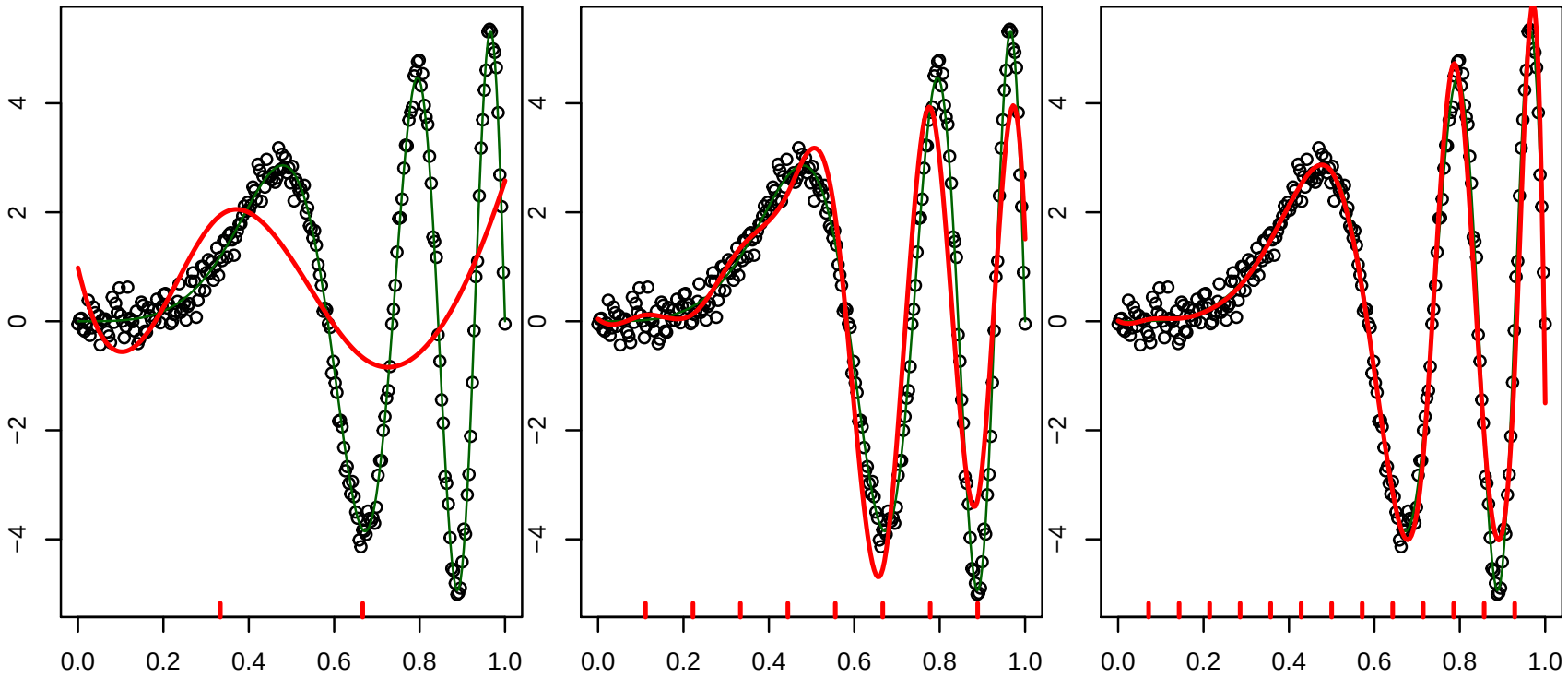
Numero di nodi e distorsione

Spline di ordine 1 (in rosso) per diverse scelte dei nodi (rappresentati sull'asse x), in verde la vera $f(x)$.



Numero di nodi e distorsione

Spline di ordine 2 (in rosso) per diverse scelte dei nodi (rappresentati sull'asse x), in verde la vera $f(x)$.



Levigatezza della spline: nodi fissati

Una strategia differente prevede di fissare i nodi e quindi

imporre qualche restrizione sui coefficienti tale che cambiando la restrizione si cambi il grado di levigatezza.

o

invece di stimare i coefficienti col metodo dei minimi quadrati, aggiungere una penalizzazione che favorisca funzioni più lisce.

Operando così, il grado di levigatezza dipende da un numero, che varia nel continuo.

Indice

Penalizzazione per la ruvidità

Dati i nodi e quindi una base, i parametri sono determinati minimizzando

$$\sum_{i=1}^{n}(y_i - f(x_i, \boldsymbol{\beta}, \boldsymbol{b}))^2 + \lambda S(f(x, \boldsymbol{\beta}, \boldsymbol{b}))$$

dove

- $S(f(x, \boldsymbol{\beta}, \boldsymbol{b}))$ è una misura di quanto "ruvida" è $f()$,
- $\lambda > 0$ è (almeno per ora) una costante fissata.

Penalizzazione: tipo ridge

Una penalizzazione semplice è data da

$$S(f(x)) = \sum_{i=1}^{K} b_i^2 = \boldsymbol{b}^T \boldsymbol{b}$$

Si può mostrare che usare una penalizzazione di questo tipo equivale a porre un vincolo del tipo

$$\sum_{i=1}^{K} b_i^2 < C$$

per qualche C.

TPB in notazione matriciale

Consideriamo la base delle potenze troncate

$$f(x) = \beta_1 + \beta_2 x + \ldots + \beta_{p+1} x^p + \sum_{k=1}^{K} b_k (x - \nu_k)_+^p$$

e poniamo

$$\boldsymbol{\theta} = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_{p+1} \\ b_1 \\ \vdots \\ b_K \end{bmatrix}, \quad X = \begin{bmatrix} 1 & x_1 & x_1^2 & \vdots & x_1^p & (x_1 - \nu_1)_+^p & \ldots & (x_1 - \nu_K)_+^p \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_i & x_i^2 & \vdots & x_i^p & (x_i - \nu_1)_+^p & \ldots & (x_i - \nu_K)_+^p \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \vdots & x_n^p & (x_n - \nu_1)_+^p & \ldots & (x_n - \nu_K)_+^P \end{bmatrix}$$

Quindi $f(\mathrm{x}) = X\boldsymbol{\theta}$ e il modello è

$$\mathrm{y} = X\boldsymbol{\theta} + \boldsymbol{\varepsilon}$$

Penalizzazione e TPB

Considerimao la penalizzazione ridge

$$S(f(x, \boldsymbol{\theta})) = \boldsymbol{\theta}^T D \boldsymbol{\theta}$$

dove $D = \text{diag}(0_{p+1}, 1_K)$,

Il minimo di

$$\sum_{i=1}^{n} (y_i - f(x_i, \boldsymbol{\theta}))^2 + \lambda \boldsymbol{\theta}^T D \boldsymbol{\theta}$$

si ha per

$$\hat{\theta} = (X^T X + \lambda D)^{-1} X^T y$$

Quindi la spline, scritta in forma di lisciatore lineare, è

$$\hat{y} = X(X^T X + \lambda D)^{-1} X^T y$$

Quanti sono i parametri?

"Nominalmente" il modello ha $K + p + 1$ parametri (e la varianza)

Però, questi non possono variare liberamente per via della penalizzazione.

Il numero effettivo di parametri è valutato come la traccia della matrice di lisciamento

$$\text{trace}(X(X^T X + \lambda D)^{-1} X^T)$$

(Analogamente, si ricordi che nel ML, il numero di parametri è la traccia della matrice di proiezione.)

Penalizzazione: derivata seconda

Un'alternativa è impiegare una derivata della funzione spline

$$S(f(x)) = \int (f^{(q)}(t))^2 dt$$

dove $q \leq$ grado della spline. (In genere $q = 2$ per una spline cubica.)
Si noti che, se la base è $\mathrm{B}() = (B_1(), \ldots, B_K())$, sicchè

$$\hat{f}(x) = \boldsymbol{b}^T \mathrm{B}(x)$$

allora

$$S(f(x)) = \int (f^{(q)}(t))^2 dt = \boldsymbol{b}^T D \boldsymbol{b}$$

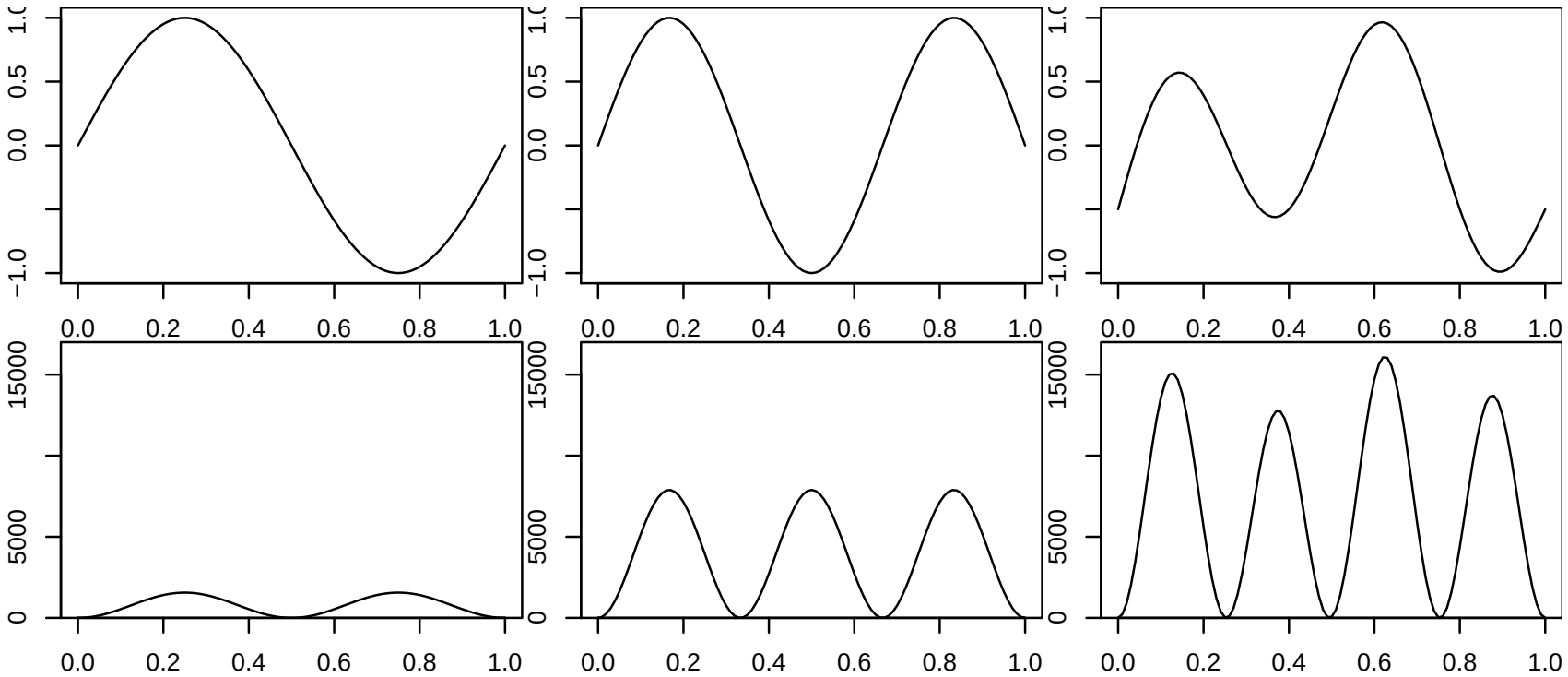
dove

$$D = \int_a^b \mathrm{B}^{(q)}(x)[\mathrm{B}^{(q)}(x)]^T dx$$

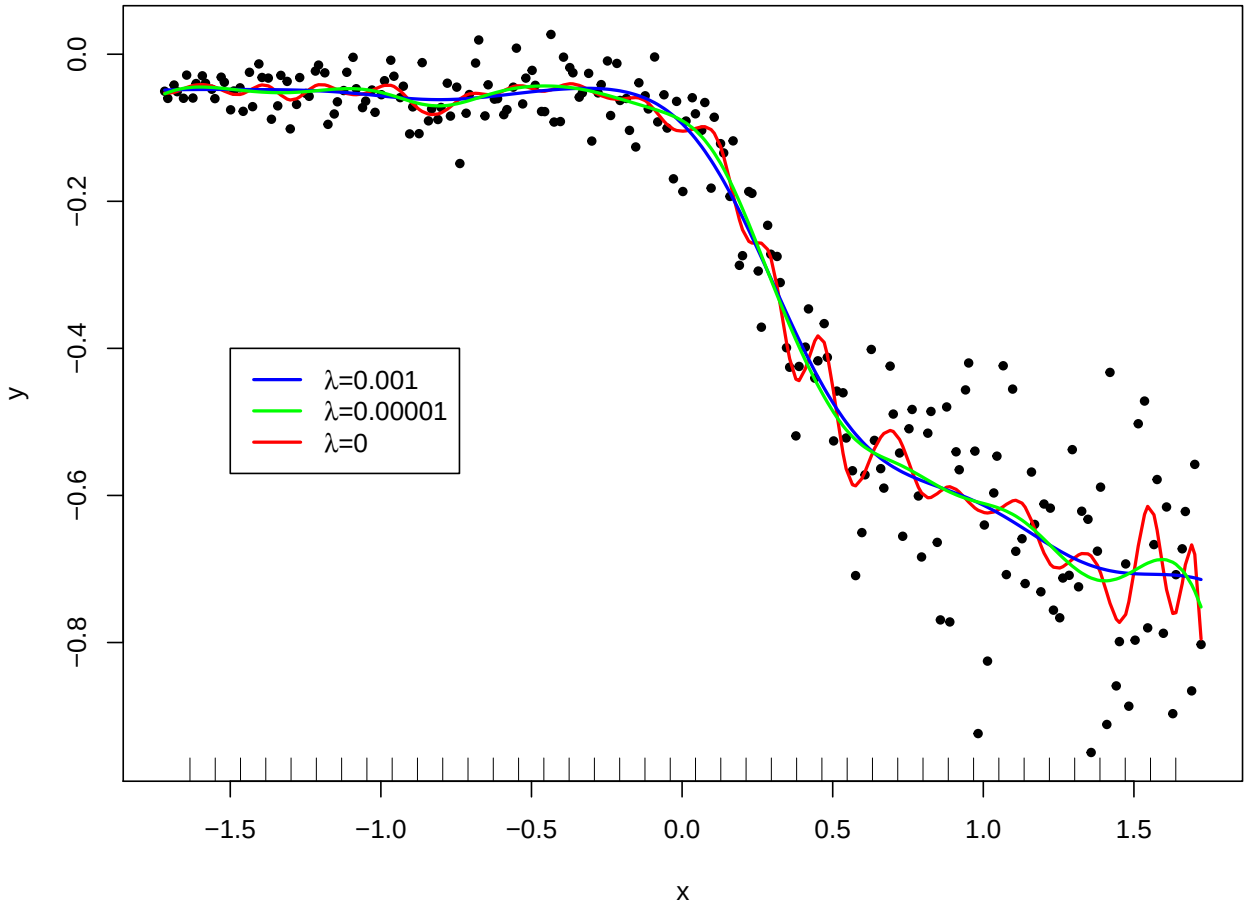
(In certi casi si usano approssimazioni.)

Funzioni e le loro derivate seconde

Si riportano tre funzioni e i quadrati delle derivate seconde.

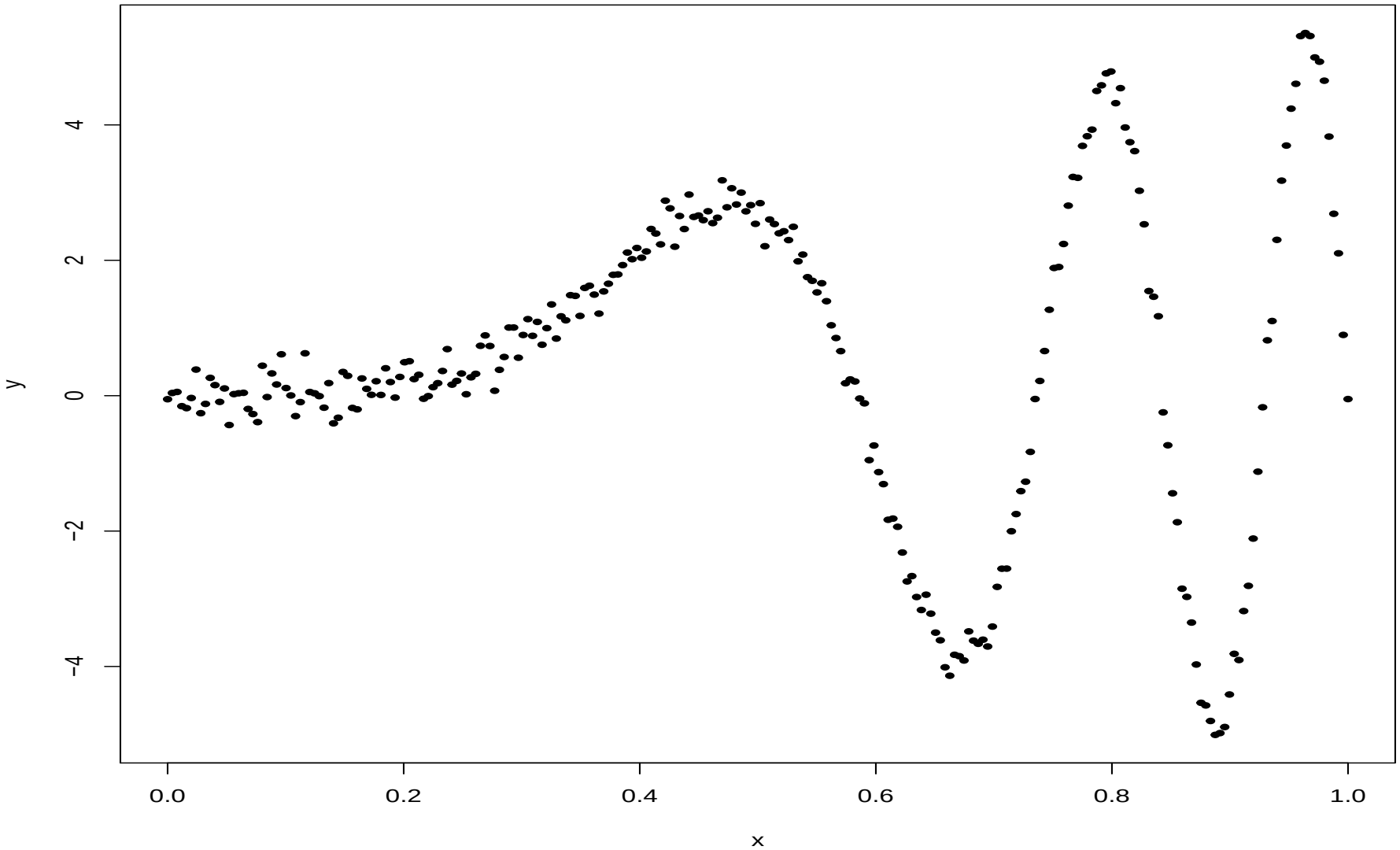


Spline penalizzate



Nel grafico si riportano 3 spline stimate usando 40 nodi con diversi pesi per la penalizzazione (tipo ridge).

Spline penalizzate e non



Quindi quanti nodi?

Il trucco della penalizzazione fa sì che possiamo fissare i nodi all'inizio, come però?

- L'idea è che, siccome si usa la penalizzazione, la scelta dei nodi è poco importante purché
  - non siano troppo pochi (in generale: da 20 a 40 minimo),
  - ci siano osservazioni tra i nodi (almeno 4-5 osservazioni tra uno e l'altro)
- Si possono fissare tanti nodi quante le osservazioni, può essere però computazionalmente oneroso e, in genere, non è necessario.
- Strategie tipiche per la scelta della posizione dei nodi sono
  - quantili empirici di x
  - nodi equispaziati.

Da qui in poi, i nodi $\nu_1, \ldots, \nu_K$ sono fissati in qualche modo.

(Si noti che possiamo verificare se i nodi sono in numero sufficiente stimando il modello con un numero maggiore e verificando se le cose cambiano.)

Indice

Scelta di λ

Il ruolo di λ è analogo alla scelta del numero di vicini o della banda.

Lo stesso criterio: validazione incrociata, può essere impiegato per la scelta.

Si noti che la spline è un lisciatore lineare, quindi i risultati visti in generale per i lisciatori lineari si possono impiegare anche ora.

Inotlre, anche la formula per il GCV è valida.

Errore quadratico complessivo

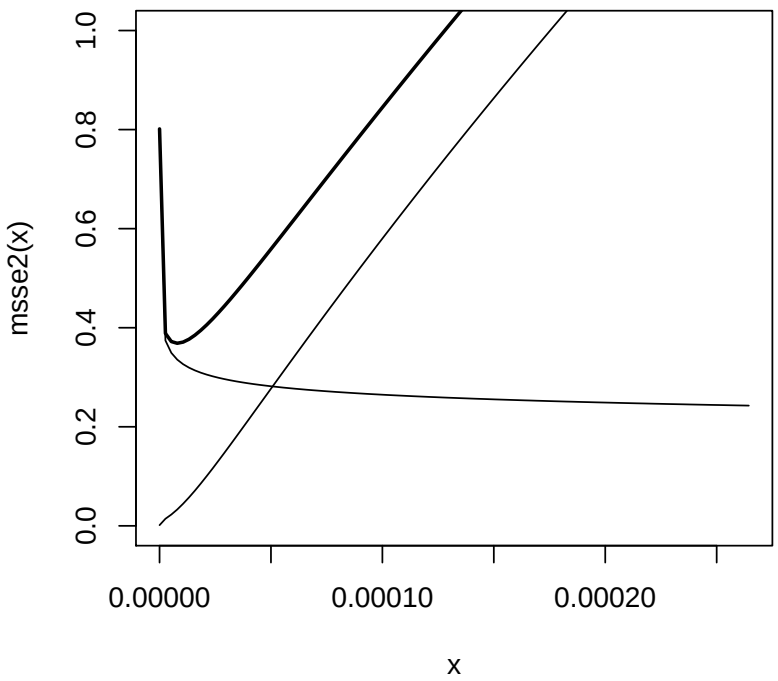
Let

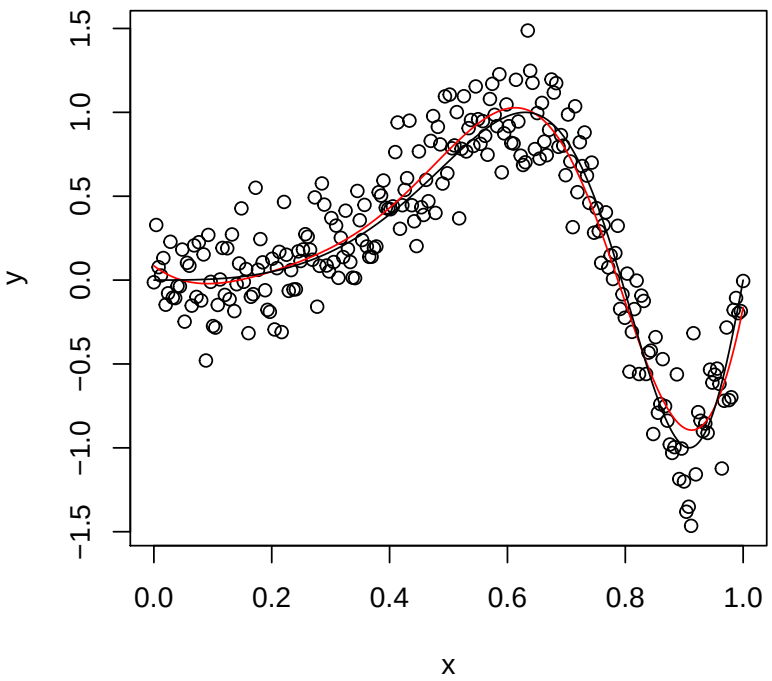
$$
\begin{aligned}
\mathrm{R}(\hat{f}) &= E\left(\sum_{i=1}^{n}(\hat{f}(x_i) - f(x_i))^2\right) \\
&= \sum_{i=1}^{n}[(E(\hat{f}(x_i)) - f(x_i))^2 + V(\hat{f}(x_i))] \\
&= (E(L\boldsymbol{y}) - \boldsymbol{f})^T(E(L\boldsymbol{y}) - \boldsymbol{f}) + \sum_{i=1}^{n} V(L\boldsymbol{y})_{ii} \\
&= \boldsymbol{f}^T(L - I)^T(L - I)\boldsymbol{f} + \mathrm{trace}[V(L\boldsymbol{y})] \\
&= \boldsymbol{f}^T(L - I)^T(L - I)\boldsymbol{f} + \mathrm{trace}[LV(\boldsymbol{y})L^T] \\
&= \boldsymbol{f}^T(L - I)^T(L - I)\boldsymbol{f} + \sigma_\varepsilon^2\mathrm{trace}[LL^T]
\end{aligned}
$$

In questa scomposizione, la prima parte rappresenta il quadrato della distorsione, la seconda è la varianza.

Errore quadratico complessivo

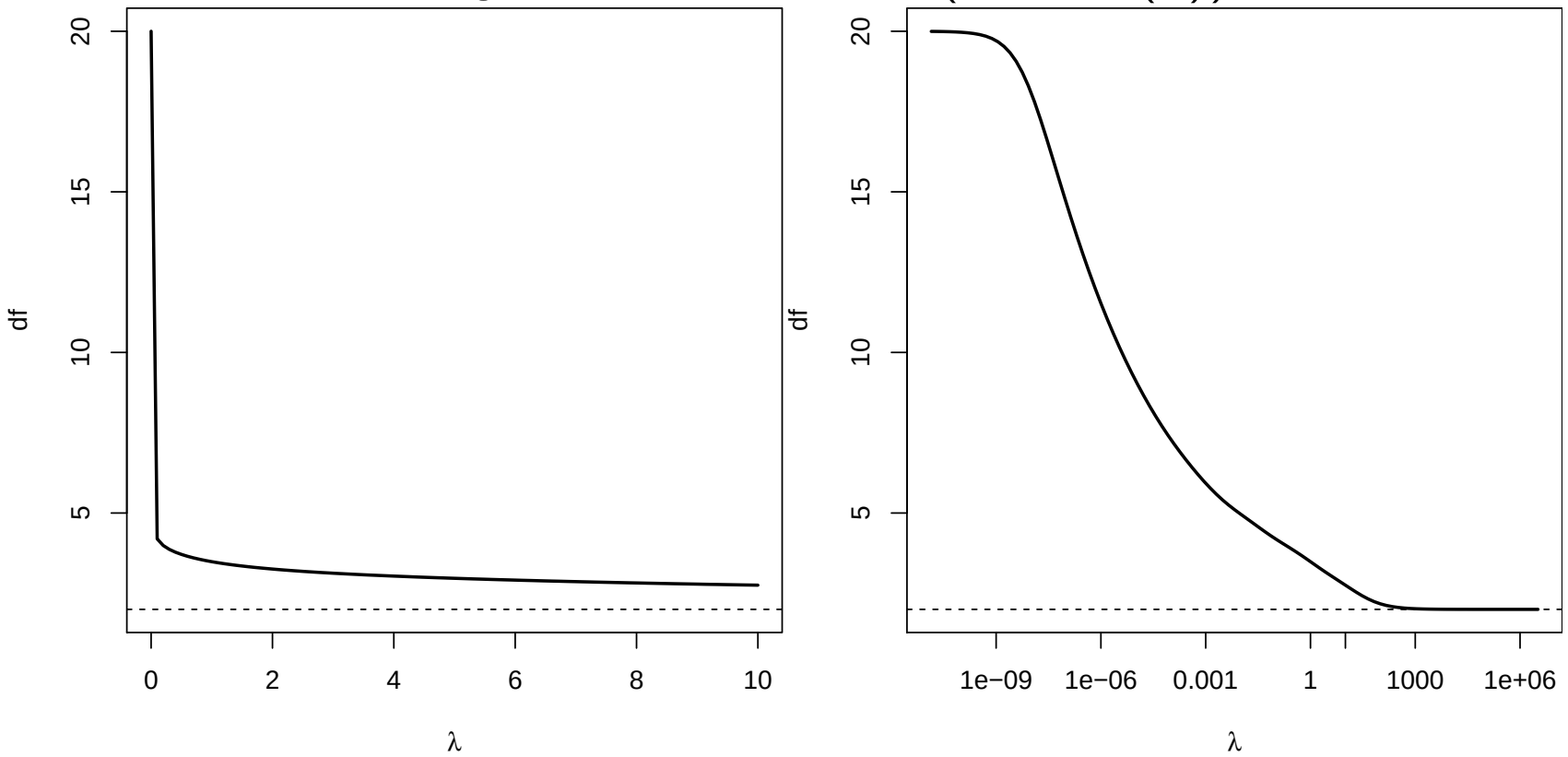
Esempio di scomposizione di R in distorsione e varianza





Gradi di libertà e penalizzazione

Il coefficiente λ determina la levigatezza della funzione stimata, è rilevante studiare come variano i gradi di libertà di $\hat{f}$ ($\mathrm{df} = \mathrm{tr}(L)$) in funzione di λ.



Spline in R: gam (mgcv)

ci sono numerosi pacchetti per la stima di spline in R, uno dei più potenti e versatili è il pacchetto `mgcv` di Wood.

Le funzioni del pacchetto presentano molte opzioni, nella forma più semplice, comunque:

```
fit=gam(y~s(x))
fit.s=summary(fit)
plot(fit)
```

si effettua una stima scegliendo via GCV il parametro di lisciamento λ (la scelta dei nodi è fatta in maniera predefinita di cui non ci preoccupiamo, volendo si possono cambiare).

Stimare una spline con gam

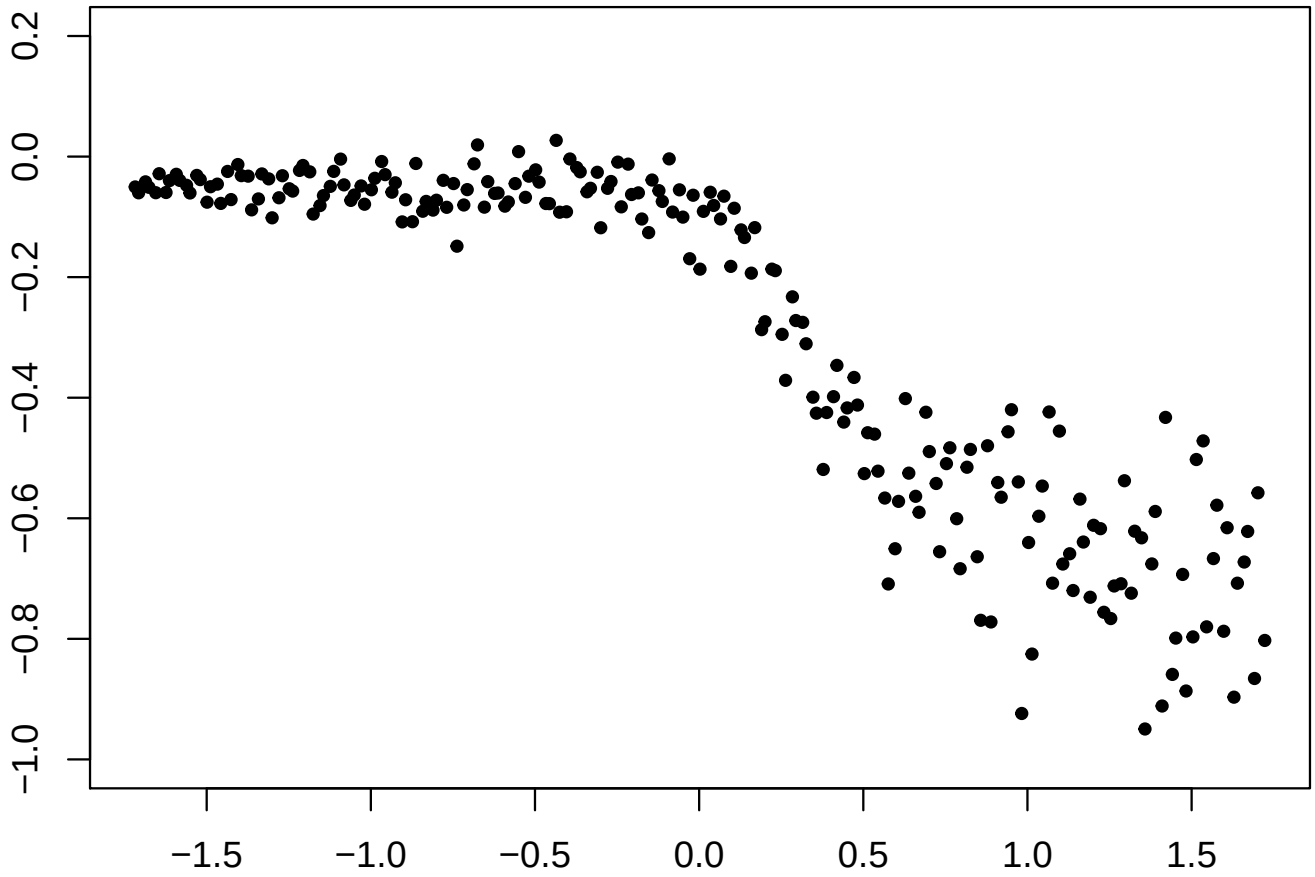
Tra gli argomenti della funzione `s()`

- `k` dimensione della base
- `fx` se usare una penalizzazione

```
fit2=gam(y~s(x,k=3))
fit4=gam(y~s(x,k=10))
plot(fit2)
plot(fit8)
```

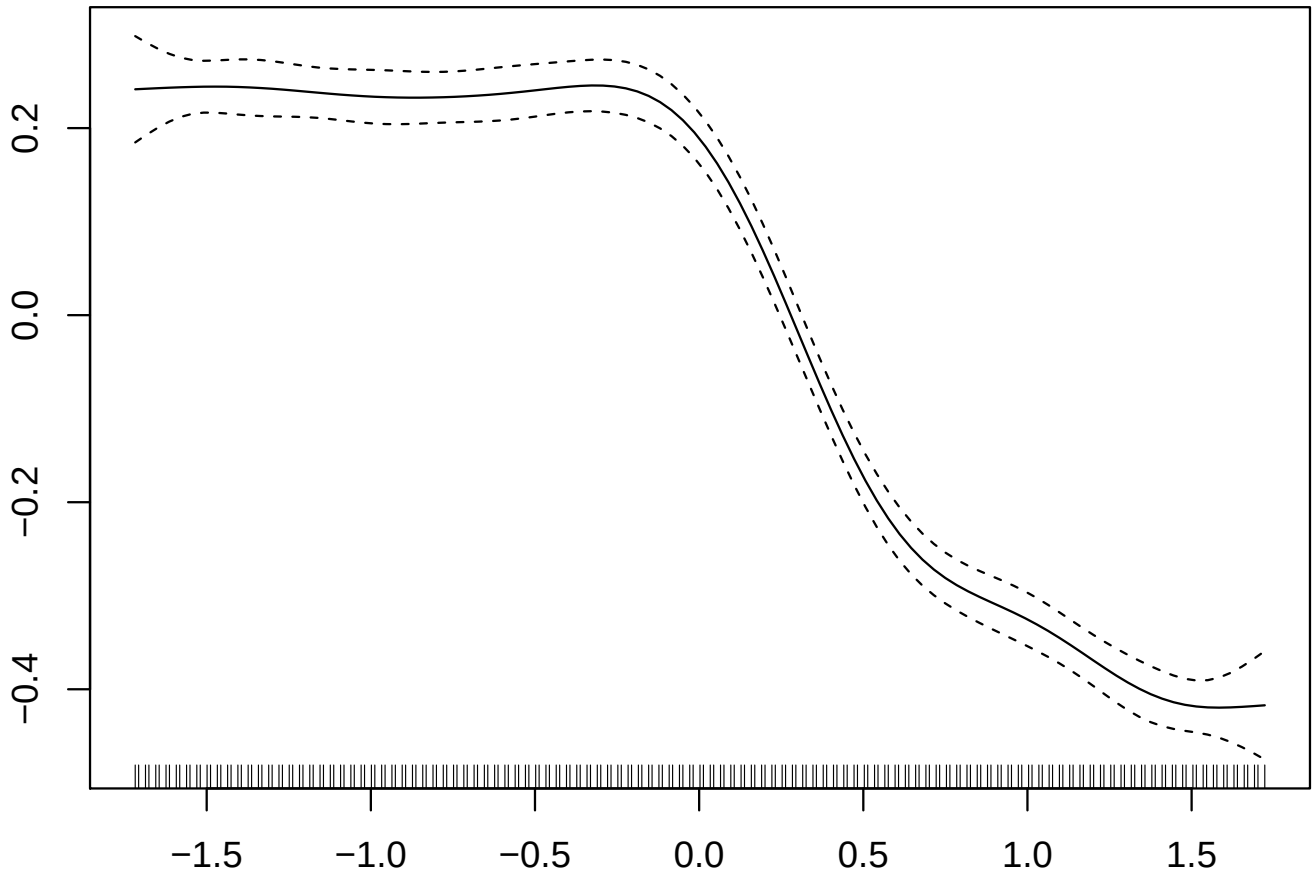
LIDAR: GAM

```
plot(lidar$range,lidar$logratio,pch=20,xlab="x",ylab="y",ylim=c(-1,0.2))
```



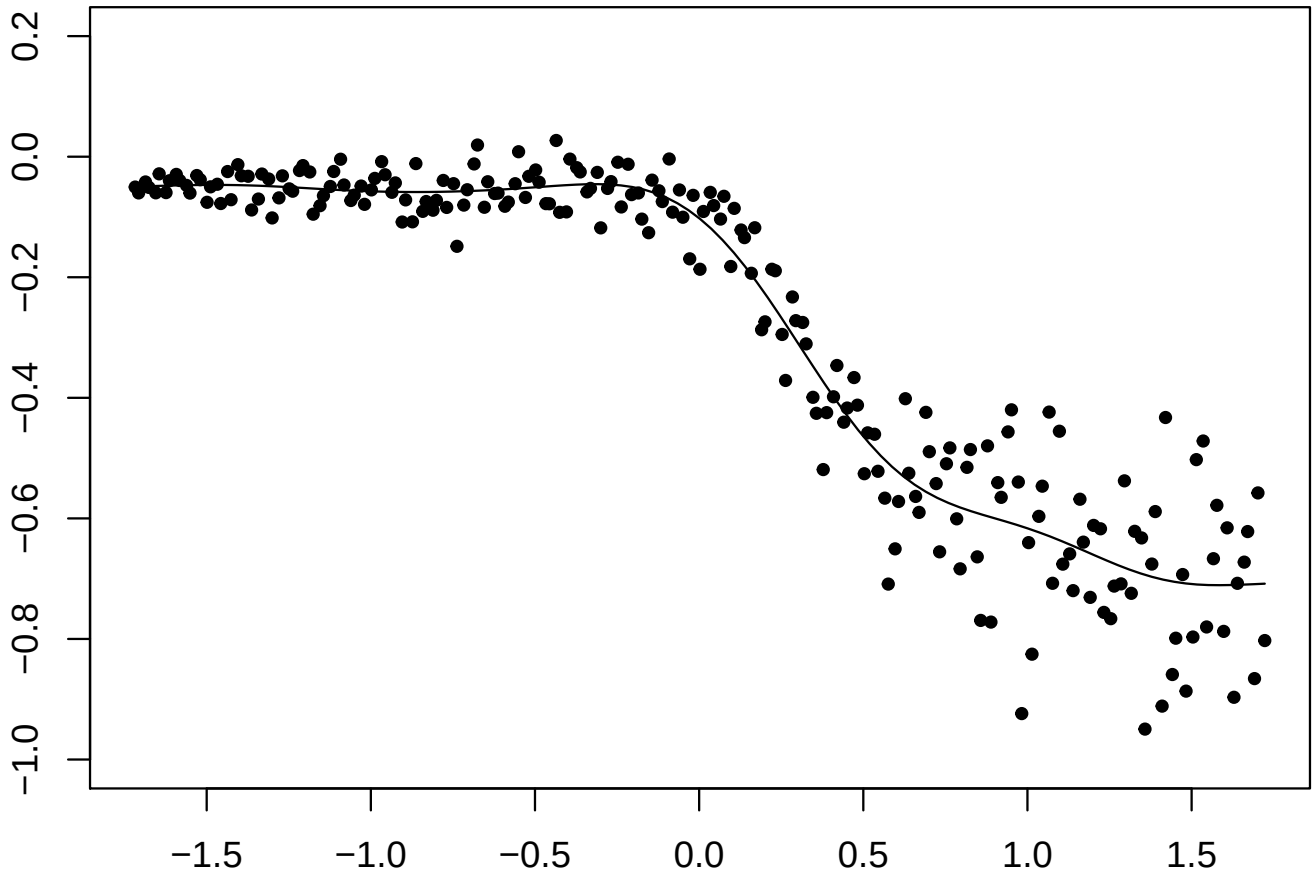
LIDAR: GAM

```
fit=gam(logratio~s(range),data=lidar)
plot(fit)
```



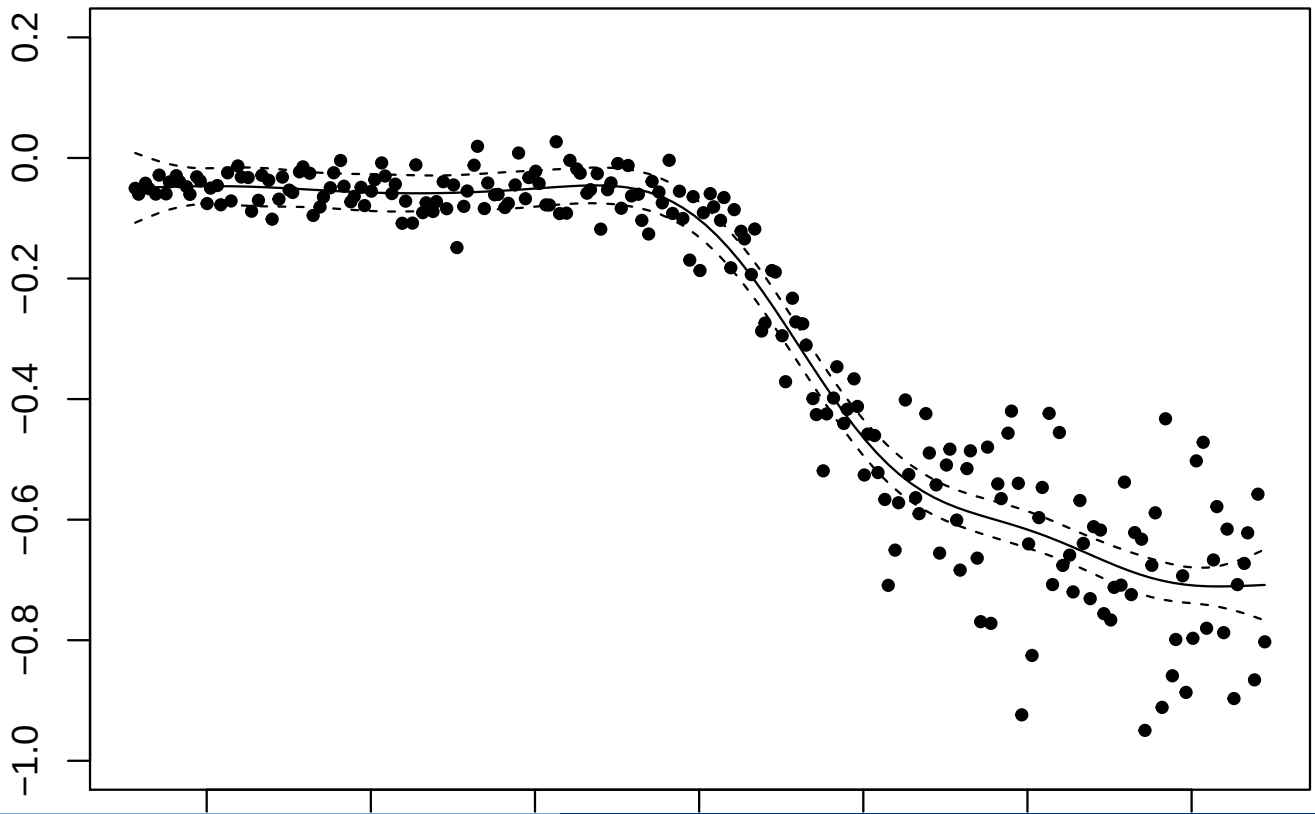
LIDAR: GAM

```
plot(lidar$range,lidar$logratio,pch=20,xlab="x",ylab="y",ylim=c(-1,0.2))
curve(predict(fit,newdata=data.frame(range=x)),add=TRUE)
```



LIDAR: GAM

```
plot(lidar$range,lidar$logratio,pch=20,xlab="x",ylab="y",ylim=c(-1,0.2))
xx=seq(min(lidar$range),max(lidar$range),length=100)
pr=predict(fit,newdata=data.frame(range=xx),
                se.fit=TRUE)
matlines(xx,cbind(pr$fit-2*pr$se.fit,pr$fit,pr$fit+2*pr$se.fit),
         type="l",lty=c(2,1,2),col="black")
```

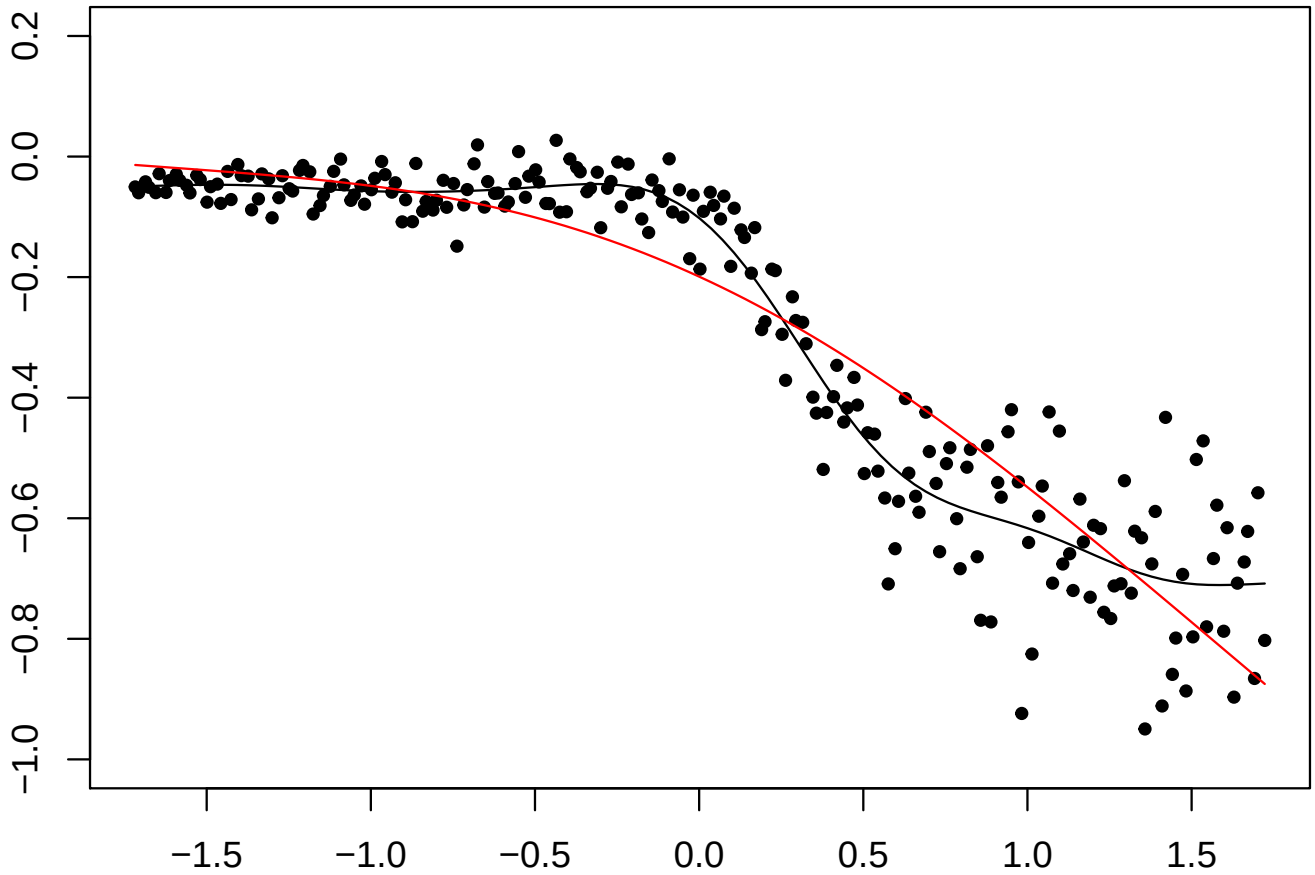


LIDAR: GAM

```
summary(fit)



Family: gaussian
Link function: identity

Formula:
logratio ~ s(range)

Parametric coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.291156   0.005327  -54.65   <2e-16

(Intercept) ***
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Approximate significance of smooth terms:
           edf Ref.df     F p-value
s(range) 7.874   8.67 297.8  <2e-16 ***
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.921   Deviance explained = 92.4%
GCV = 0.0065345  Scale est. = 0.0062721  n = 221
```

LIDAR: GAM

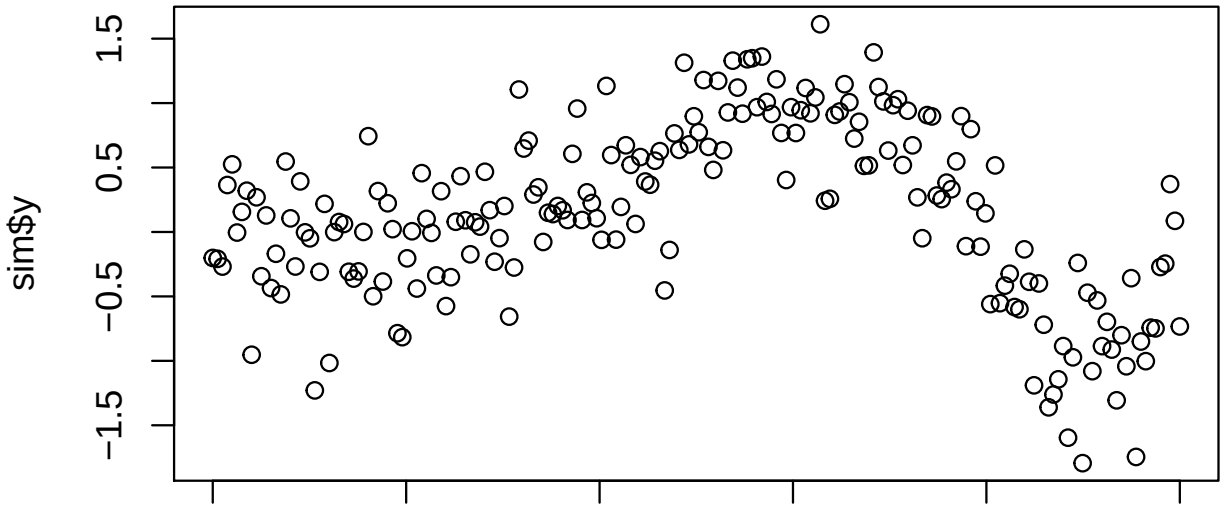
```
plot(lidar$range,lidar$logratio,pch=20,xlab="x",ylab="y",ylim=c(-1,0.2))
curve(predict(fit,newdata=data.frame(range=x)),add=TRUE)
fit1=gam(logratio~s(range,fx=TRUE,k=3),data=lidar)
curve(predict(fit1,newdata=data.frame(range=x)),add=TRUE,col="red")
```



Qualche esperimento con gam

"Verifichiamo" che il numero di nodi è indifferente purché abbastanza grande.

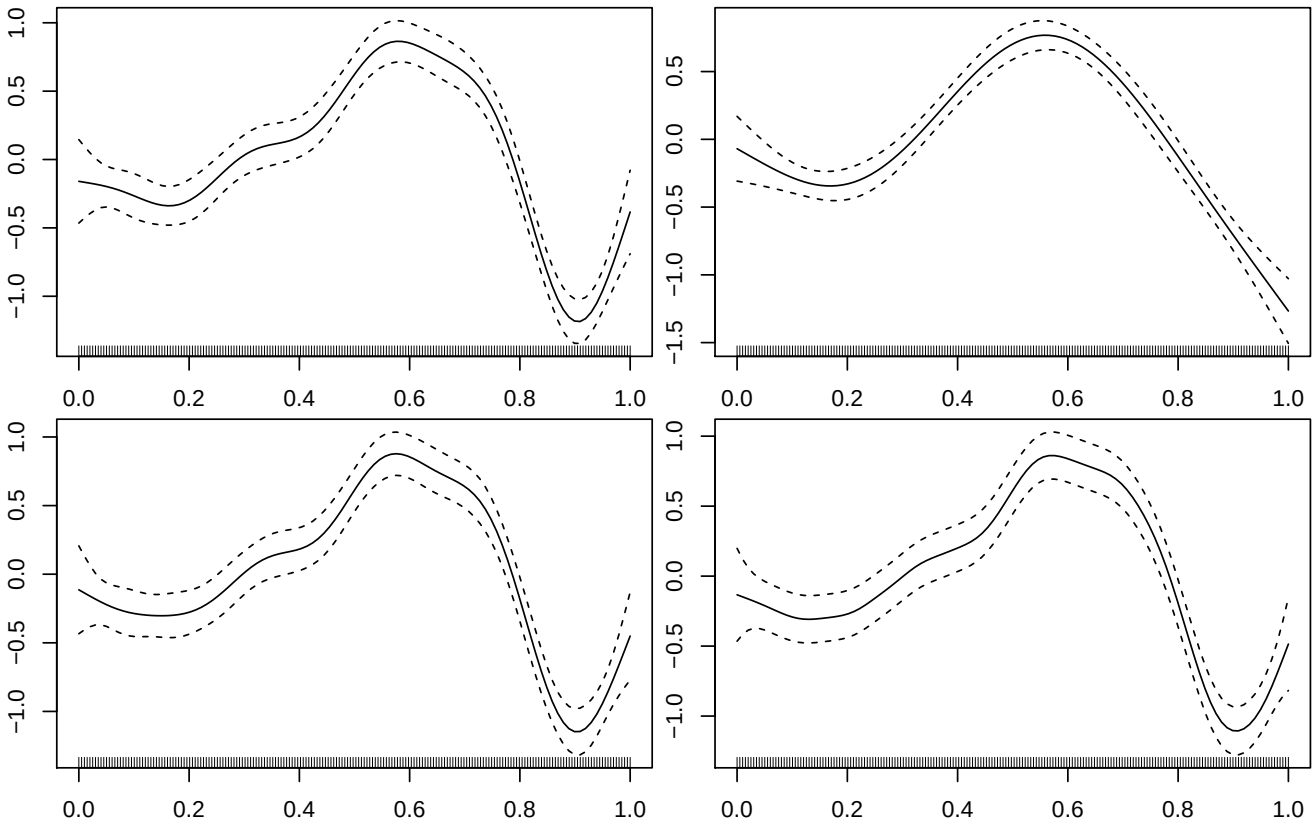
```
sim=data.frame(x=seq(0,1,length=200)) #sort(runif(150,0,1)))
sim$m=sin(2*pi*sim$x^3)
sim$y=sim$m+rnorm(nrow(sim),0,0.4)
plot(sim$x,sim$y)
```



Qualche esperimento con gam

```
fit0=gam(y~s(x),data=sim)
fit1=gam(y~s(x,k=5),data=sim)
fit2=gam(y~s(x,k=12),data=sim)
fit3=gam(y~s(x,k=30),data=sim)
```

Qualche esperimento con gam



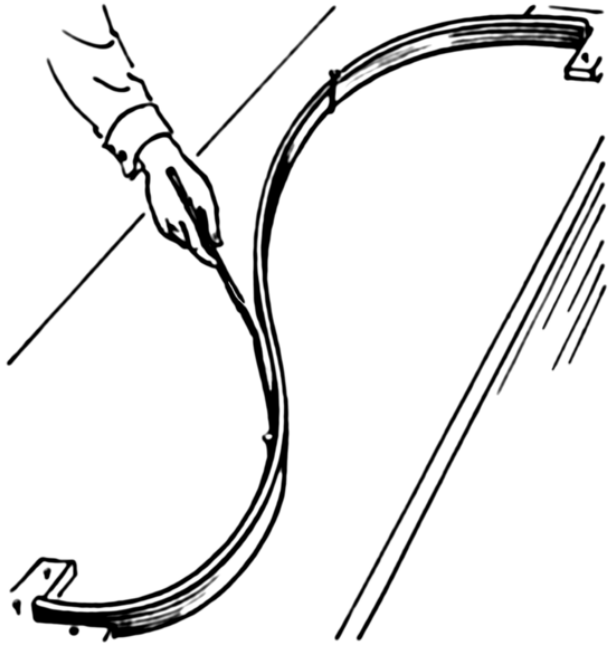
Indice

Spline

Una spline di ordine p con nodi $\nu_1, \ldots, \nu_K$ è un polinomio a tratti continuo con derivate continue fino all'ordine $p - 1$.

Il termine spline deriva dal nome di un attrezzo da disegno (una striscia di metallo flessibile usata per agevolare il disegno di curve).

Una **spline cubica** è una spline di ordine $p = 3$ (continua con derivata seconda continua).



A spline, or the more modern term flexible curve, consists of a long strip fixed in position at a number of points that relaxes to form and hold a smooth curve passing through those points for the purpose of transferring that curve to another material. (Wikipedia)

Le spline naturali cubiche sono interpolanti ottimali

Sia

- (x_i, y_i), $i = 1, \ldots, n$: dove $x_i < x_{i+1}$
- $g(x)$ la spline cubica naturale che interpola i punti (naturale significa che $g''(x_1) = g''(x_n) = 0$)

allora $g()$ è l'interpolante più liscia nel senso che minimizza

$$J(f) = \int_{x_1}^{x_n} f''(x)^2 dx$$

tra le funzioni f che interpolano i punti, sono assolutamente continue e hanno derivata prima continua.

In altre parole, le spline cubiche naturali sono le funzioni più lisce per interpolare dei punti.

Dimostrazione

Sia $f()$ interpolante (x_i, y_i) e sia $h = f - g$

$$\begin{aligned}\int_{x_1}^{x_n} f''(x)^2 dx &= \int_{x_1}^{x_n} (g''(x) + h''(x))^2 dx \\ &= \int_{x_1}^{x_n} g''(x)^2 dx + \int_{x_1}^{x_n} g''(x)h''(x)dx + \int_{x_1}^{x_n} h''(x)^2 dx\end{aligned}$$

Dimostrazione

Sia $f()$ interpolante (x_i, y_i) e sia $h = f - g$

$$\int_{x_1}^{x_n} f''(x)^2 dx = \int_{x_1}^{x_n} g''(x)^2 dx + \int_{x_1}^{x_n} g''(x)h''(x)dx + \int_{x_1}^{x_n} h''(x)^2 dx$$

Si ah anche, integrando per parti

$$\int_{x_1}^{x_n} g''(x)h''(x)dx = g''(x_n)h'(x_n) - g''(x_1)h'(x_1) - \int_{x_1}^{x_n} g'''(x)h'(x)dx$$

$$= -\int_{x_1}^{x_n} g'''(x)h'(x)dx$$

$$= -\sum_{i=1}^{n-1} g'''(x_i^+) \int_{x_1}^{x_n} h'(x)dx$$

$$= -\sum_{i=1}^{n-1} g'''(x_i^+)(h(x_{i+1}) - h(x_i)) = 0$$

Dimostrazione

Sia $f()$ interpolante (x_i, y_i) e sia $h = f - g$

$$\int_{x_1}^{x_n} f''(x)^2 dx = \int_{x_1}^{x_n} g''(x)^2 dx + \int_{x_1}^{x_n} g''(x)h''(x)dx + \int_{x_1}^{x_n} h''(x)^2 dx$$

Si ah anche, integrando per parti

$$\int_{x_1}^{x_n} g''(x)h''(x)dx = 0$$

Quindi

$$\int_{x_1}^{x_n} f''(x)^2 dx = \int_{x_1}^{x_n} g''(x)^2 dx + \int_{x_1}^{x_n} h''(x)^2 dx \geq \int_{x_1}^{x_n} g''(x)^2 dx$$

dove si ha l'eguaglianza sse $h''(x) = 0$ per $x_1 < x < x_n$ cioè solo se $f = g$.

Conseguenza

La proprietà sopra significa che se minimizzo

$$\sum_{i=1}^{n}(y_i - f(x_i))^2 + \lambda \int f''(x)^2 dx$$

tra tutte le funzioni f continue con derivata prima continua su $[x_1, x_n]$, allora il minimo è ua spline cubica naturale.

DIM: supponiamo che f^* minimizzi l'espressione sopra e non sia una spline cubica naturale, allora si prenda la spline cubica naturale che interpola $(x_i, f^*(x_i))$, questa comporta la medesima somma dei quadrati degli scarti, ma con una minore penalizzazione.

Indice

Stima della varianza

La varianza dell'errore σ^2 può essere stimata, analogamente a quanto si fa nel ML, con

$$\hat{\sigma}^2 = \frac{\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2}{n - \mathrm{df}} = \frac{RSS}{n - \mathrm{df}}$$

dove i gradi di libertà della spline sono $\mathrm{tr}(L)$.
Si noti però che

$$\begin{aligned} E(RSS) &= E((\mathrm{y} - \hat{\mathrm{y}})^T(\mathrm{y} - \hat{\mathrm{y}})) \\ &= E(\mathrm{y}^T(L - I)^T(L - I)\mathrm{y}) \\ &= \mathrm{f}^T(L - I)^T(L - I)\mathrm{f} + \sigma^2\mathrm{tr}((L - I)^T(L - I)) \\ &= \mathrm{f}^T(L - I)^T(L - I)\mathrm{f} + \sigma^2(\mathrm{tr}(LL^T) - 2\mathrm{tr}(L) + n) \end{aligned}$$

quindi, assumendo che la distorsione sia trascurabile, uno stimatore non distorto per σ^2 è

$$\tilde{\sigma}^2 = \frac{RSS}{n - 2\mathrm{tr}(L) + \mathrm{tr}(LL^T)}$$

Stima della varianza

La varianza dell'errore σ^2 può essere stimata, analogamente a quanto si fa nel ML, con

$$\hat{\sigma}^2 = \frac{\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2}{n - \mathrm{df}} = \frac{RSS}{n - \mathrm{df}}$$

dove i gradi di libertà della spline sono $\mathrm{tr}(L)$.
Si noti però che

$$\begin{aligned} E(RSS) &= E((\mathrm{y} - \hat{\mathrm{y}})^T(\mathrm{y} - \hat{\mathrm{y}})) \\ &= \mathrm{f}^T(L - I)^T(L - I)\mathrm{f} + \sigma^2(\mathrm{tr}(LL^T) - 2\mathrm{tr}(L) + n) \end{aligned}$$

quindi, assumendo che la distorsione sia trascurabile, uno stimatore non distorto per σ^2 è

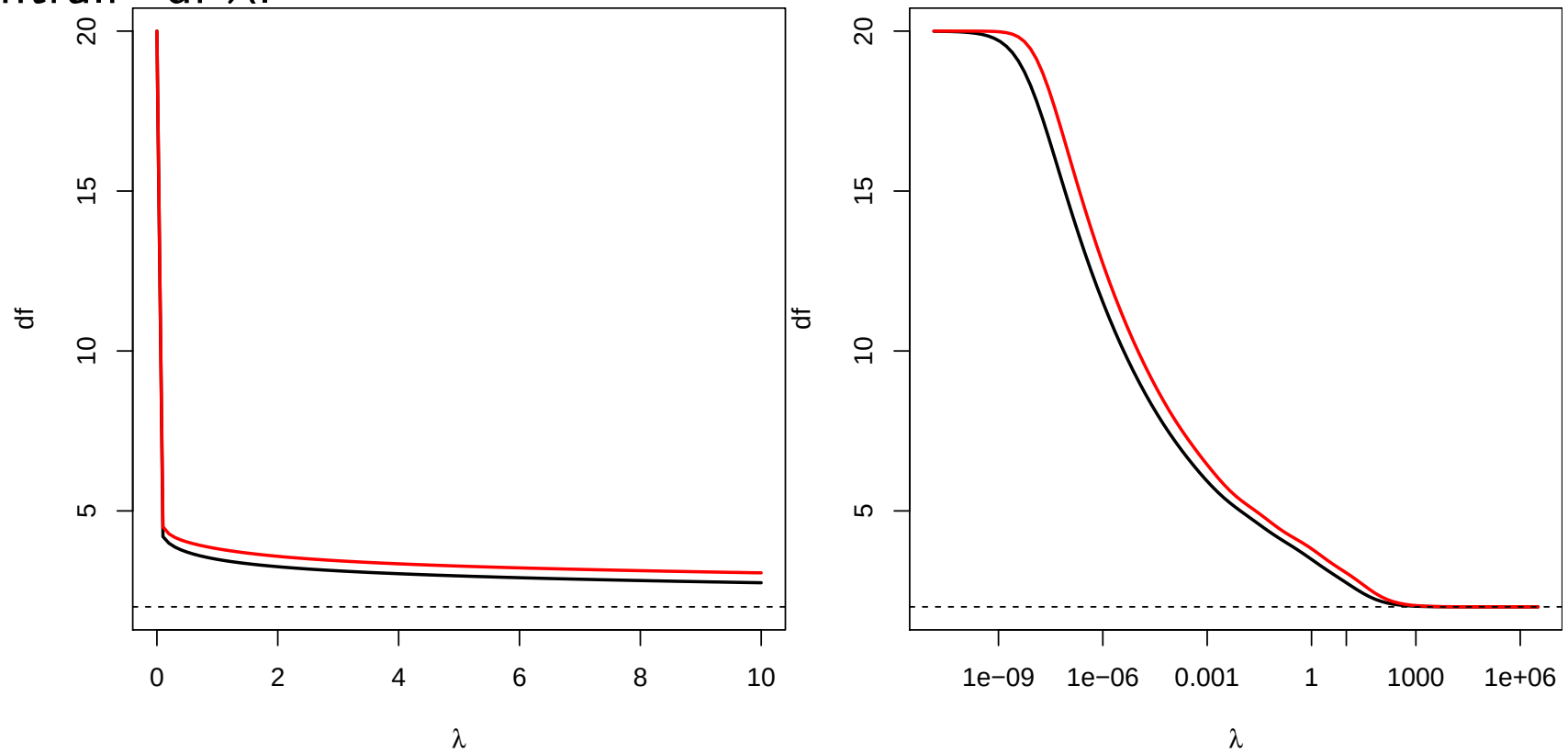
$$\tilde{\sigma}^2 = \frac{RSS}{n - 2\mathrm{tr}(L) + \mathrm{tr}(LL^T)}$$

Dove si ha che

- $n - 2\mathrm{tr}(L) + \mathrm{tr}(LL^T)$ sono i GdL residui del modello
- $2\mathrm{tr}(L) - \mathrm{tr}(LL^T)$ è una misura alternativa dei GdL del modello

Gradi di libertà: le due misure

Le due misure dei gradi di libertà differiscono in particolare per valori "centrali" di λ.



Indice

Spline: diverse rappresentazioni

- In termini generali la rappresentazione è

$$y_i = \beta_1 + \sum_{k=1}^{K} b_k B_k(z_i) + \text{other variables} + \varepsilon_i$$

  per un dato insieme di funzioni note $B_1, \ldots, B_K$ che è detta **base**.
- Questo fa sì che lo spazio di funzioni in cui cerchiamo un'approssimazione di f contiene funzioni del tipo

$$s(z) = \sum_{k=1}^{K} b_k B_k(z)$$

- Ci sono diverse scelte di $B_k(z)$ che fan sì che si ottengano forme flessibili a partire da un numero ridotto di $B_k(z)$ (basso K),
- una buona scelta di queste funzioni rende gli stimatori più efficienti.

Basi alternative

La base a polinomi troncati è la più agevole da trattare dal punto di vista teorico, non è ottimale dal punto di vista computazionale.

Il problema principale è che la matrice di disegno X ha colonne fortemente correlate, il che può comportare instabilità numerica.

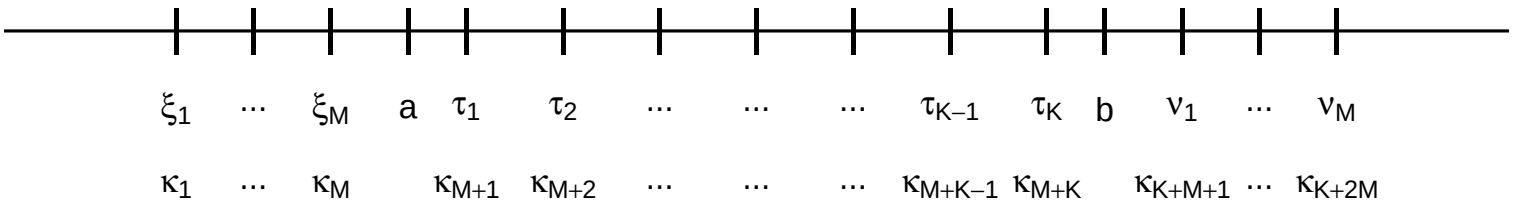
Esistono numerose alternative

- B-spline
- P-spline
- base radiale
- ...

Si tenga presente che, in linea di principio, un cambio di base non comporta un cambiamento del modello.

B-spline: costruzione della base

- siano $\tau_1 < \ldots < \tau_K$ i nodi interni;
- sia $[a, b]$ il supporto di x ($a < \tau_1$, $b > \tau_K$);
- fissati, arbitrariamente, $\xi_1 \leq \ldots \xi_M \leq a$ e $\chi_M \geq \ldots \geq \chi_1 \geq b$ (ad es. $\xi_i = a$ e $\nu_i = b$);
- si ha allora la sequenza $\nu_1, \ldots, \nu_{K+2M}$.



B-spline: costruzione della base

Sia $B_{i,m}$ l'i-esima funzione base di ordine $m < M$, $i = 1, \ldots, K + 2M - m$, questa è definita ricorsivamente da

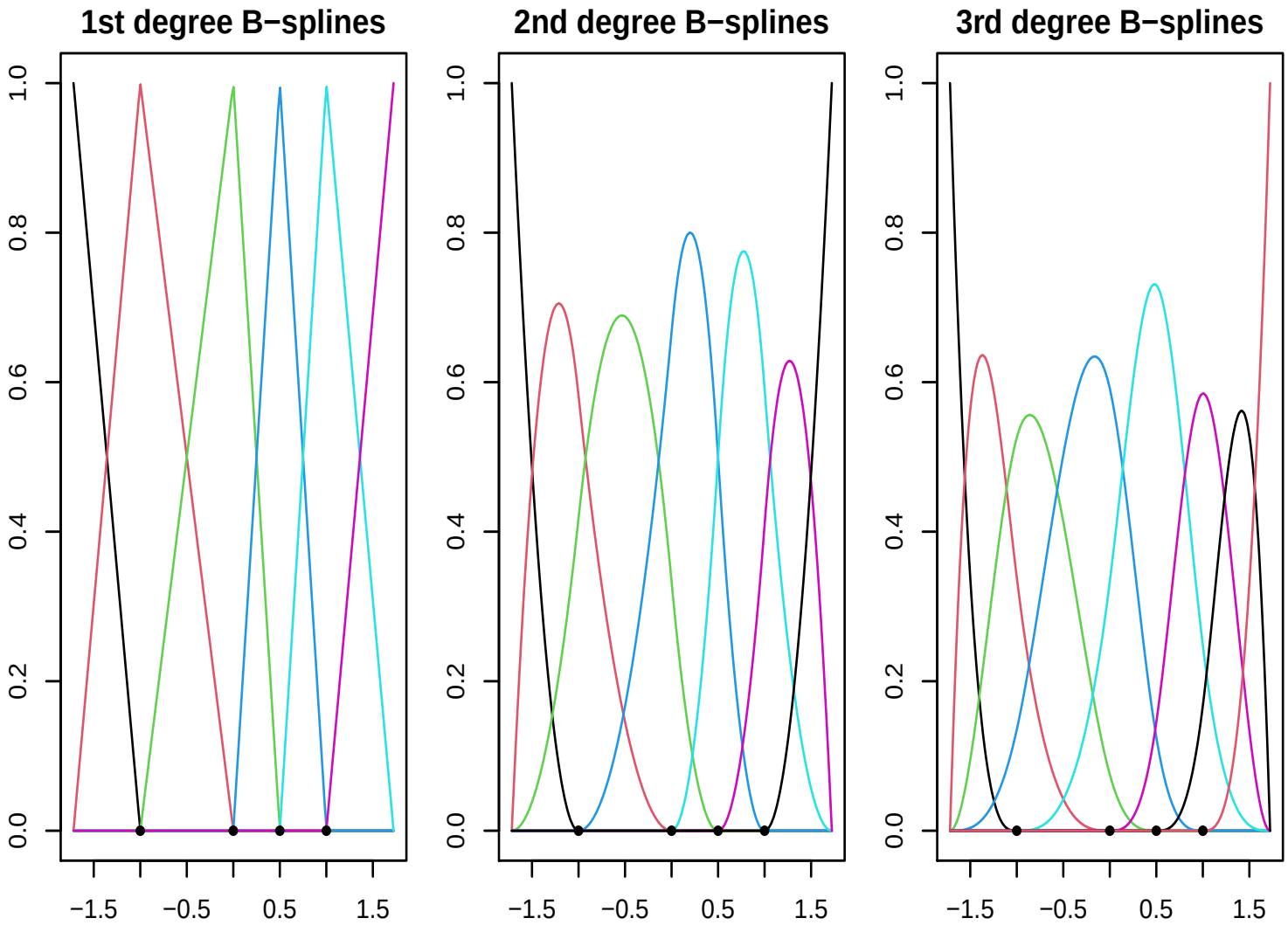
$$B_{i,1}(x) = \begin{cases} 1 & \text{if } \nu_i \leq x < \nu_{i+1} \\ 0 & \text{altrimenti} \end{cases}$$

per $i = 1, \ldots, K + 2M - 1$ e

$$B_{i,m}(x) = \frac{x - \nu_i}{\nu_{i+m-1} - \nu_i} B_{i,m-1}(x) + \frac{\nu_{i+m} - x}{\nu_{i+m} - \nu_{i+1}} B_{i+1,m-1}(x)$$

$i = 1, \ldots, K + 2M - m$.
Per $M = 4$ si ottengono $K + 4$ spline cubiche.

B-spline


1st degree B−splines
2nd degree B−splines
3rd degree B−splines
0.0
0.2
0.4
0.6
0.8
1.0
−1.5
−0.5
0.5
1.5

B-spline: matrice di penalizzazione

Fissato $M = 4$, la matrice di penalizzazione ha elemento i, j

$$\Omega_{ij} = \int_a^b B_i''(x) B_j''(x) dx$$

Wand and Ormerod (2009) propongono, per il calcolo di Ω,

$$\Omega = (\tilde{B}'')^T \text{diag}(\text{w}) \tilde{B}''$$

where

$$[\tilde{B}'']_{ij} = \tilde{B}_j(\tilde{x}_i) \in \mathcal{M}_{3(K+7)\times(K+4)}$$

$$\tilde{x} = \left(\nu_1, \frac{\nu_1 + \nu_2}{2}, \nu_2, \ldots, \nu_{K+7}, \frac{\nu_{K+7} + \nu_{K+8}}{2}, \nu_{k+8} \right)$$

$$\text{w} = \left(\frac{1}{6}(\Delta\nu)_1, \frac{4}{6}(\Delta\nu)_1, \frac{1}{6}(\Delta\nu)_1, \ldots, \frac{1}{6}(\Delta\nu)_{K+7}, \frac{4}{6}(\Delta\nu)_{K+7}, \frac{1}{6}(\Delta\nu)_{K+7} \right)$$

where $(\Delta\nu)_h = \nu_{h+1} - \nu_h$.

P-spline

Le P-spline sono costituite da

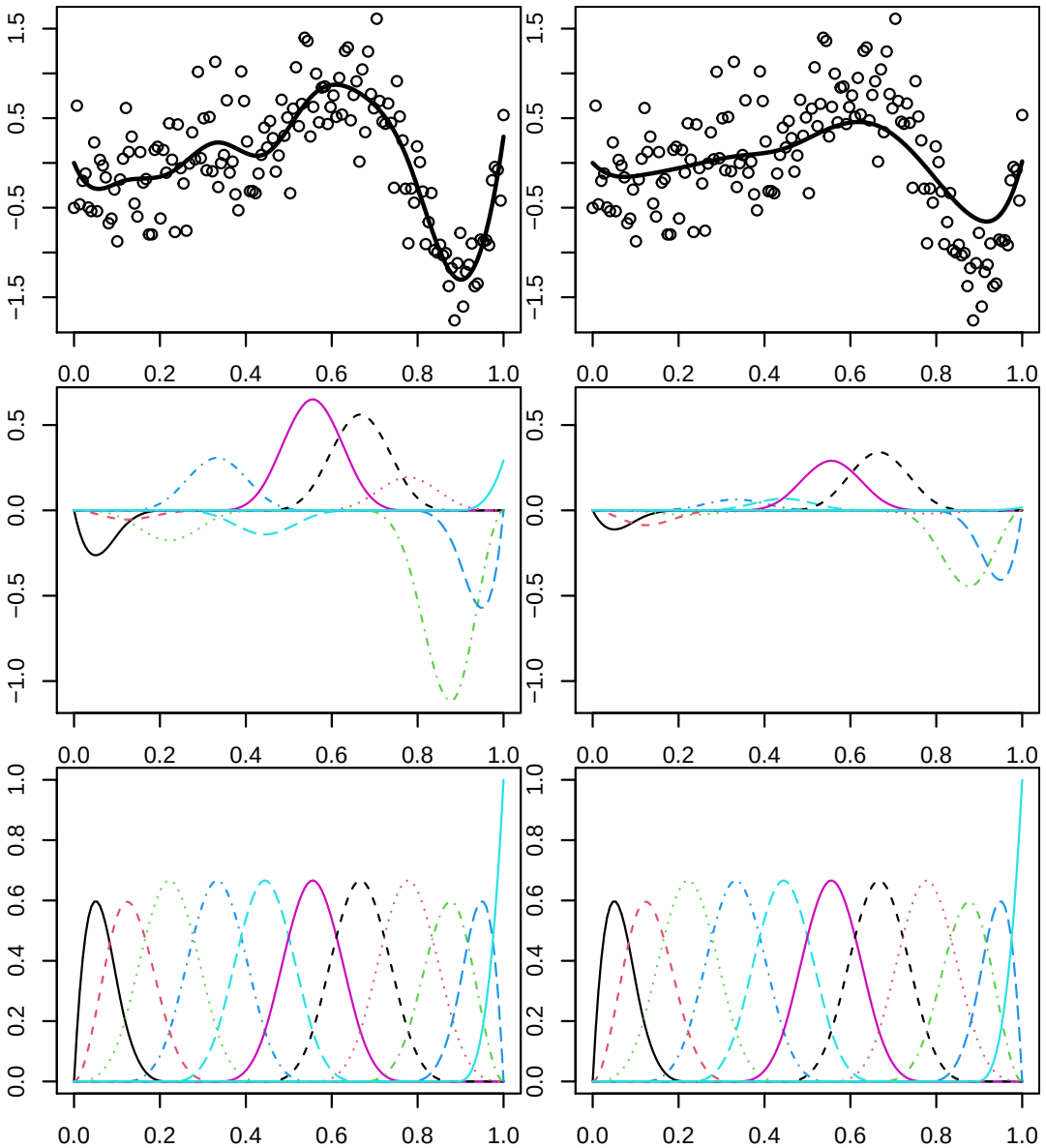
- una base di B-spline, di solito su nodi equidistanti
- la penalizzazione

$$\sum_{i=1}^{K-1} (b_{i+1} - b_i)^2$$

ovvero

$$\boldsymbol{b}^T \begin{bmatrix} 1 & -1 & 0 & \cdot & \cdot & \\ -1 & 2 & -1 & \cdot & \cdot & \\ 0 & -1 & 2 & \cdot & \cdot & \\ & \cdot & \cdot & \cdot & \cdot & \cdot \\ & \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix} \boldsymbol{b}$$

Un esempio di *P*-spline



Due *P*-spline, per quella a sinistra

$$\sum_{i=1}^{K-1}(b_{i+1}-b_i)^2 = 9.95$$

mentre per quella a destra

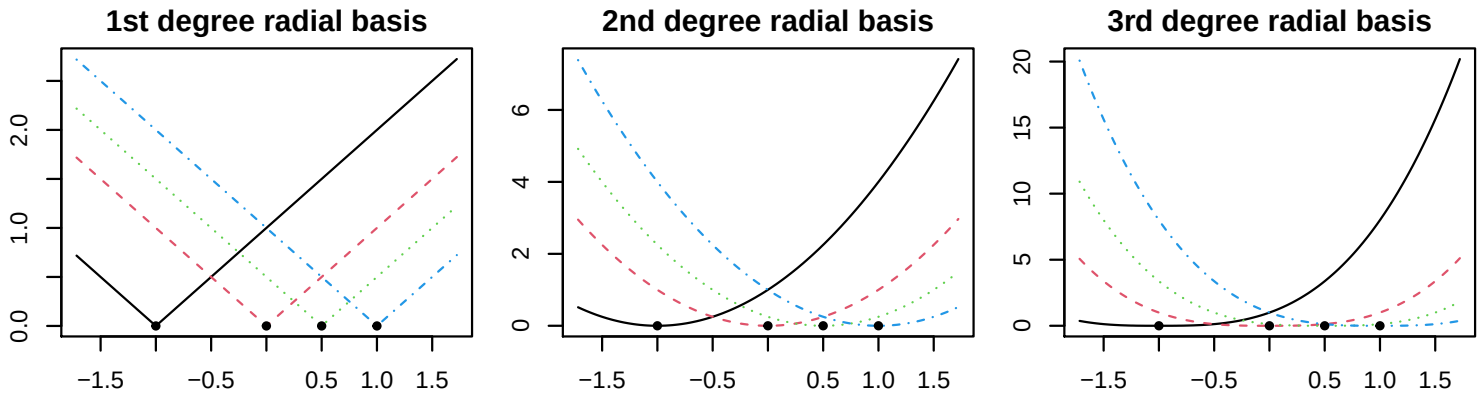
$$\sum_{i=1}^{K-1}(b_{i+1}-b_i)^2 = 1.45$$

(pannello di mezzo: funzioni base moltiplicate per i rispettivi coefficienti, cioè la cui somma è la curva finale.)

Base radiale

La base radiale di ordine m con nodi $\nu_1, \ldots, \nu_K$ è

$$1, x, \ldots, x^m, B_k(x) = |x - \nu_k|^m$$



Con matrice di penalizzazione

$$[D]_{ij} = |\nu_i - \nu_j|^3$$

Indice

Più esplicative

Supponiamo che le osservazioni includano più esplicative

$$x_i, z_i, u_{i1}, \ldots, u_{iq}$$

Si possono ipotizzare diversi modelli

- componenti parametriche e una componente non parametrica

$$y_i = \boldsymbol{\beta}^T \mathrm{u}_i + f(x_i) + \varepsilon_i$$

- componenti parametriche e non parametriche

$$y_i = \boldsymbol{\beta}^T \mathrm{u}_i + f_x(x_i) + f_z(z_i) + \varepsilon_i$$

- componenti parametriche e non parametrica multipla

$$y_i = \boldsymbol{\beta}^T \mathrm{u}_i + f(x_i, z_i) + \varepsilon_i$$

Componenti parametriche e non parametrica

Data per la spline f la rappresentazione

$$f(x_i) = b_0 + b_1 x_i + \sum_{j=1}^{K} B_j(x_i) b_{1+j}$$

con matrice di penalizzazione S il modello

$$y_i = \boldsymbol{\beta}^T \mathrm{u} + f(x_i) + \varepsilon_i$$

è stimato minimizzando la funzione obiettivo

$$\sum_{i=1}^{n} (y_i - \boldsymbol{\beta}^T \mathrm{u}_i - f(x_i))^2 + \lambda \mathrm{b}^T S \mathrm{b}$$

in forma matriciale

$$||\boldsymbol{y} - H\boldsymbol{\theta}||^2 + \lambda \boldsymbol{\theta}^T S' \boldsymbol{\theta}$$

dove

$$H = [U\ X], \quad \boldsymbol{\theta}^T = (\boldsymbol{\beta}, \mathrm{b}), \quad S' = ?$$

Componenti parametriche e non parametriche

Siano le due spline f_x, f_z rappresentate da

$$f_x(x_i) = b_0 + b_1 x_i + \sum_{j=1}^{K_B} B_j(x_i) b_{1+j}$$

$$f_z(z_i) = d_1 z_i + \sum_{j=1}^{K_D} D_j(z_i) d_{1+j}$$

con penalizzazioni S_B, S_D.
Si noti che la rappresentazione di f_z non include l'intercetta per garantire l'identificabilità del modello

$$y_i = \boldsymbol{\beta}^T \mathrm{u}_i + f_x(x_i) + f_z(z_i) + \varepsilon_i$$

Componenti parametriche e non parametriche

Il modello

$$y_i = \boldsymbol{\beta}^T \mathrm{u}_i + f_x(x_i) + f_z(z_i) + \varepsilon_i$$

è stimato minimizzando la funzione obiettivo

$$\sum_{i=1}^{n} (y_i - \boldsymbol{\beta}^T \mathrm{u}_i - f_x(x_i) - f_z(z_i))^2 + \lambda \mathrm{b}^T S_x \mathrm{b} + \lambda \mathrm{d}^T S_z \mathrm{d}$$

in forma matriciale

$$||\boldsymbol{y} - H\boldsymbol{\theta}||^2 + \lambda \mathrm{b}^T S_x \mathrm{b} + \lambda \mathrm{d}^T S_z \mathrm{d}$$

dove

$$H = [U\ X\ Z], \quad \boldsymbol{\theta}^T = (\boldsymbol{\beta}, \mathrm{b}, \mathrm{d})$$

Componente non parametrica multivariata

Il modello

$$y_i = \beta^T \mathrm{u}_i + f(x_i, z_i) + \varepsilon_i$$

richiede si definisca una spline bivariata.

- in linea di principio questo funziona allo stesso modo
- **problema della dimensionalità**
  - l'onere computazionale può crescere esponenzialmente con la dimensione
  - MSE: se il campione ha dimensione n e la spline dimensione d allora tipicamente

    $$\mathrm{MSE} \approx \frac{c}{n^{4/(4+d)}}$$

    cioè la numerosità campionaria richiesta per mantenere l'MSE a un livello prespecificato cresce esponenzilmente con d:

    $$n \approx (c/\delta)^{d/4}$$

Regressione non lineare in breve

Dettagli tecnici a parte, i punti essenziali sono che

- possiamo specificare un modello per y in cui

$$\begin{array}{rcccccc} y_i & = & s(z_i) & + & \beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip} & + & \varepsilon_i \\ y_i & = & s(z_i) & + & \text{eff. lineari altre var.} & + & \text{error} \end{array}$$

  dove $s(\cdot)$ è una funzione liscia.
- disponiamo di strumenti per stimare $s(\cdot)$ con un prefissato grado di lisciamento (fix K, fix λ).
- disponiamo di strumenti per stimare $s(\cdot)$ con un grado di lisciamento ottimale.

Con questa strategia la forma della relazione tra y e z può essere qualunque (o quasi) senza bisogno di fare specifiche assunzioni.

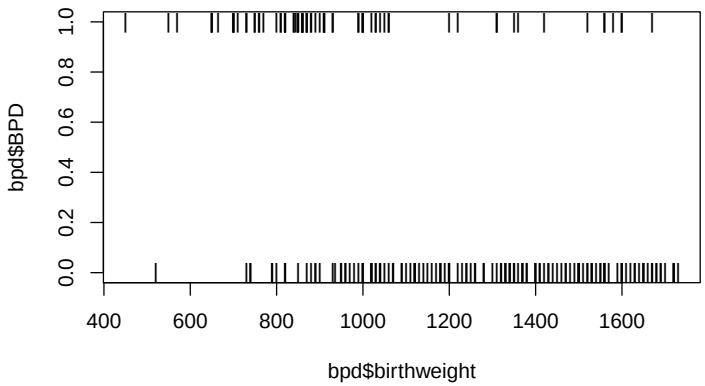
Indice

BPD data

La displasia broncopolmonare (BPD) è una malattia tipica dei bambini nati prematuramente, la cui insorgenza è plausibilmente legata al peso alla nascita (birthweight).

Per 223 bambini si è osservato

- birthweight
- insorgenza di displasia broncopolmonare (BPD)



La v. risposta è dicotomica $\Rightarrow$ GLM.

Modelli additivi generalizzati

La generalizzazione a risposta non gaussiana funziona in modo analogo all'estensione da ML a GLM.

$$\begin{array}{ccc} \text{LM} & \rightarrow & \text{GLM} \\ Y_i \sim \mathcal{N}(\mu_i, \sigma^2) & & Y_i \sim \text{Expon}(\theta_i, \phi_i) \\ E(Y_i) = \eta_i & & g(E(Y_i)) = \eta_i \\ \eta_i = \mu_i = \mathsf{x}_i\boldsymbol{\beta} & & \eta_i = \mathsf{x}_i\boldsymbol{\beta} \end{array}$$

$$\begin{array}{ccc} \text{AM} & \rightarrow & \text{GAM} \\ Y_i \sim \mathcal{N}(\mu_i, \sigma^2) & & Y_i \sim \text{Expon}(\theta_i, \phi_i) \\ E(Y_i) = \eta_i & & g(E(Y_i)) = \eta_i \\ \eta_i = \mu_i = f(x_i) & & \eta_i = f(x_i) \end{array}$$

dove

$$\ell(\beta, \mathsf{b}, \phi) = \sum_{i=1}^{n} \log(p(y_i; \theta_i)) = \sum_{i=1}^{n} (y_i\theta_i - r_i(\theta_i)))/\phi + c(\phi; y_i)$$

Modelli additivi generalizzati

La generalizzazione a risposta non gaussiana funziona in modo analogo all'estensione da ML a GLM. Data una spline

$$f(x) = \beta_1 + \beta_2 x + \sum_{j=1}^{K} \beta_{1+j} B_j(x)$$

il criterio dei minimi quadrati penalizzati

$$\sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda S(f(x))$$

diventa il criterio della verosimiglianza penalizzata

$$\ell(\beta, \mathrm{b}, \phi) - \lambda S(f(x))$$

dove

$$\ell(\beta, \mathrm{b}, \phi) = \sum_{i=1}^{n} \log(p(y_i; \theta_i)) = \sum_{i=1}^{n} (y_i \theta_i - r_i(\theta_i)))/\phi + c(\phi; y_i)$$

P-IRLS

In termini pratici si usa poi l'algoritmo IRLS usato nei GLM, con passo k

1. calcolare pseudodati

$$z_i^{[k]} = g'(\mu_i^{[k]})(y_i - \mu_i^{[k]}) + \eta_i^{[k]}$$

   e la matrice diagonale dei pesi

$$W_{ii}^{[k]} = \frac{1}{V(\mu_i^{[k]})g'(\mu_i^{[k]})^2}$$

2. porre

$$\boldsymbol{\beta}^{[k+1]} \leftarrow \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left\| \sqrt{W^{[k]}}(\mathrm{z}^{[k]} - X\boldsymbol{\beta}) \right\|^2$$

In un modello additivo la funzione obiettivo nel secondo passo è

$$\boldsymbol{\beta}^{[k+1]} \leftarrow \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left\| \sqrt{W^{[k]}}(\mathrm{z}^{[k]} - X\boldsymbol{\beta}) \right\|^2 + \lambda \boldsymbol{\beta}^T S \boldsymbol{\beta}$$

Indice

Errore di previsione con dati non Gaussiani

Con dati Gaussiani, la scelta di λ è basata sulla stima dell'errore

$$\sum_{i=1}^{n}(\hat{f}(x_i) - y_i)^2$$

ottenuta via CV, da cui il criterio del GCV (per via della linearià delle spline come lisciatori).

La stessa strategia può essere usata ora **ma** il lisciatore non è lineare, quindi il GCV e le derivazioni teoriche sono approssimate.

GCV per i GAM

L'obiettivo nella stima GAM può essere scritto in termini della devianza

$$D(\boldsymbol{\beta}) = 2(\ell(\boldsymbol{\beta}_{\text{max}}) - \ell(\boldsymbol{\beta}))$$

come

$$D(\boldsymbol{\beta}) + \sum_{j=1}^{d} \lambda_j \boldsymbol{\beta}^T S_j \boldsymbol{\beta}$$

la cui approssimazione quadratica è, per un λ fissato,

$$\left\| \sqrt{W}(z - X\boldsymbol{\beta}) \right\|^2 + \sum_{j=1}^{d} \lambda_j \boldsymbol{\beta}^T S_j \boldsymbol{\beta}$$

GCV per i GAM

L'obiettivo nella stima GAM può essere scritto come

$$D(\boldsymbol{\beta}) + \sum_{j=1}^{d} \lambda_j \boldsymbol{\beta}^T S_j \boldsymbol{\beta}$$

la cui approssimazione quadratica è, per un λ fissato,

$$\left\| \sqrt{W}(\mathrm{z} - X\boldsymbol{\beta}) \right\|^2 + \sum_{j=1}^{d} \lambda_j \boldsymbol{\beta}^T S_j \boldsymbol{\beta}$$

da cui il GCV (valido localmente)

$$\frac{n \left\| \sqrt{W}(\mathrm{z} - X\boldsymbol{\beta}) \right\|^2}{n - \mathrm{tr}(L)}$$

GCV per i GAM

L'obiettivo nella stima GAM può essere scritto come

$$D(\boldsymbol{\beta}) + \sum_{j=1}^{d} \lambda_j \boldsymbol{\beta}^T S_j \boldsymbol{\beta}$$

la cui approssimazione quadratica è, per un λ fissato,

$$\left\| \sqrt{W}(z - X\boldsymbol{\beta}) \right\|^2 + \sum_{j=1}^{d} \lambda_j \boldsymbol{\beta}^T S_j \boldsymbol{\beta}$$

da cui il GCV (valido localmente) e quindi il GCV applicabile globalmente

$$\frac{n \left\| \sqrt{W}(z - X\boldsymbol{\beta}) \right\|^2}{n - \mathrm{tr}(L)} \quad \rightarrow \quad \frac{nD(\hat{\beta})}{n - \mathrm{tr}(L)}$$

UBRE

Ricordando che la CV nasce dal rischio di previsione, $E((m(x) - \hat{m}(x))^2)$, cioè

$$E\left(\|\boldsymbol{\mu} - L\mathrm{y}\|^2\right) = \frac{1}{n}E\left(\|\mathrm{y} - L\mathrm{y}\|^2\right) - \sigma^2 + 2\mathrm{tr}(L)\frac{\sigma^2}{n}$$

dove il parametro di scala σ^2 è noto si può impiegare l'UBRE (Unbiased Risk Estimator)

$$\frac{1}{n}\|\mathrm{y} - L\mathrm{y}\|^2 - \sigma^2 + 2\mathrm{tr}(L)\frac{\sigma^2}{n}$$

L'UBRE è appropriato per i GAM in cui il parametro di scala è noto.

Calcolo dell'UBRE

A partire da

$$D(\boldsymbol{\beta}) + \sum_{j=1}^{d} \lambda_j \boldsymbol{\beta}^T S_j \boldsymbol{\beta}$$

e dall'approssimazione quadratica

$$\left\| \sqrt{W}(z - X\boldsymbol{\beta}) \right\|^2 + \sum_{j=1}^{d} \lambda_j \boldsymbol{\beta}^T S_j \boldsymbol{\beta}$$

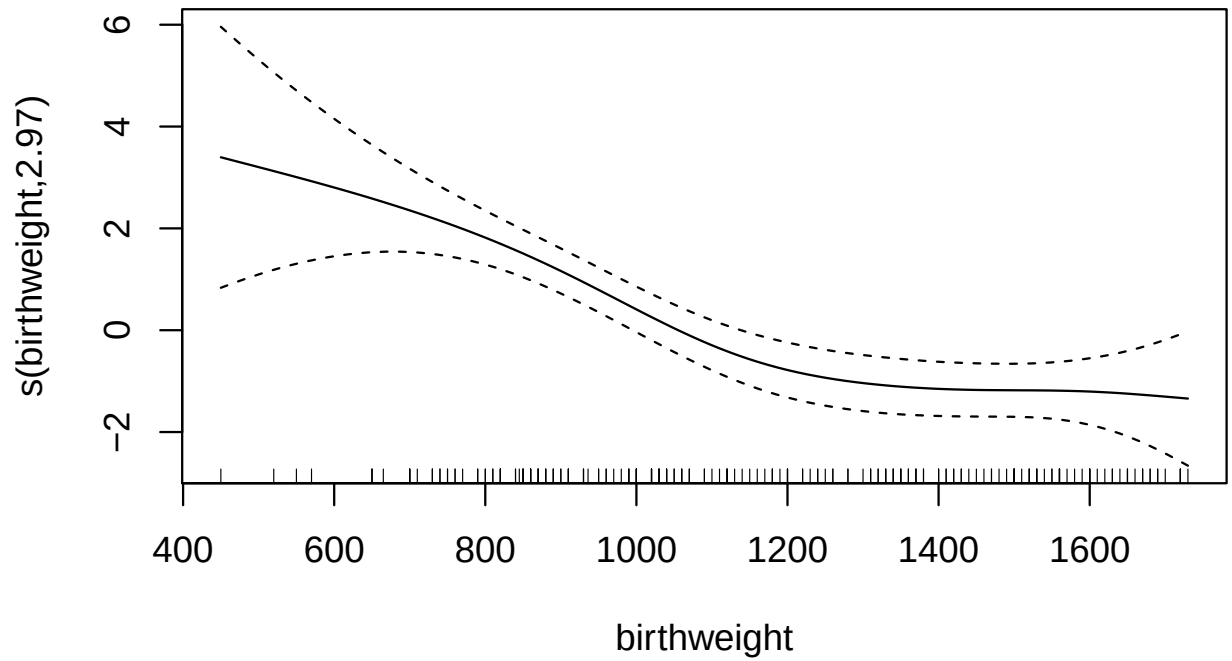
si ottiene il criterio UBRE

$$\frac{1}{n} \left\| \sqrt{W}(z - X\boldsymbol{\beta}) \right\|^2 - \sigma^2 + \frac{2\sigma^2}{n} \mathrm{tr}(L)$$

$$\frac{1}{n} D(\hat{\beta}) - \sigma^2 + \frac{2\sigma^2}{n} \mathrm{tr}(L)$$

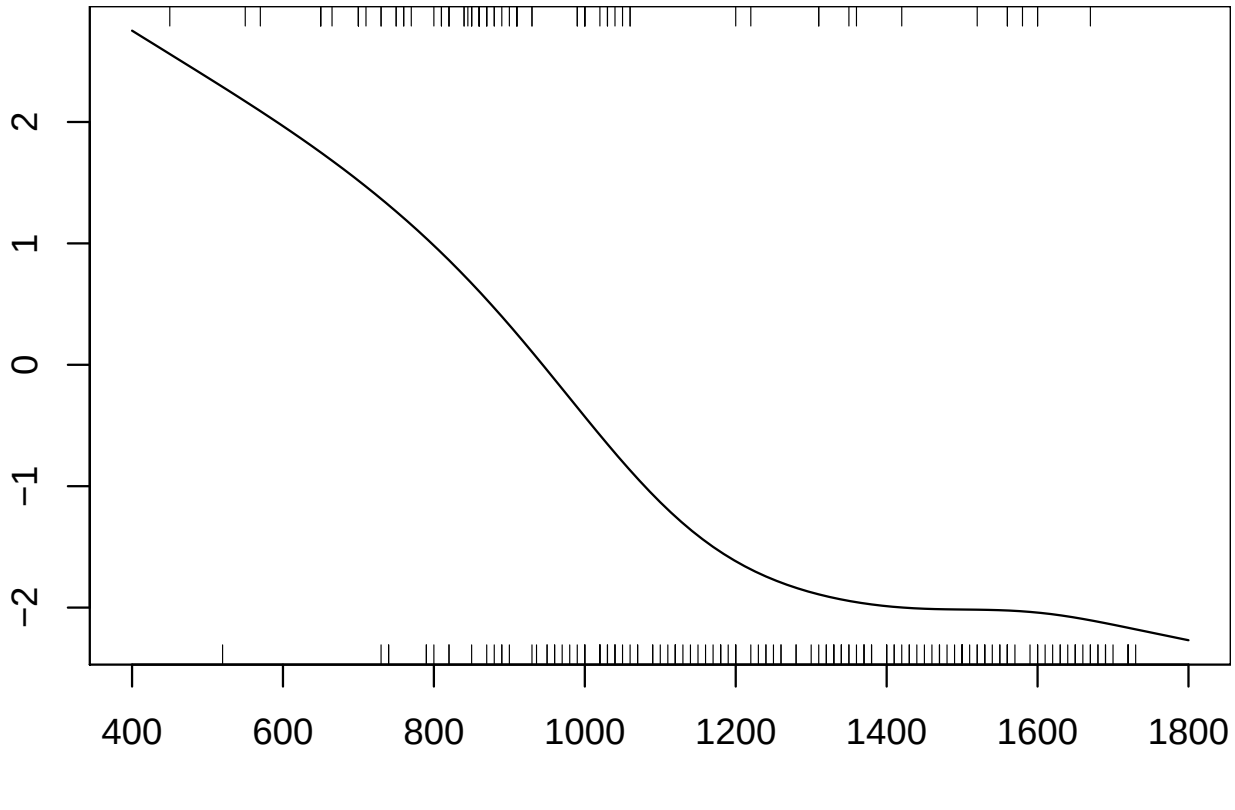
BPD data - modello stimato

```
fit=gam(BPD~s(birthweight),
        data=bpd,
        family=binomial)
plot(fit)
```



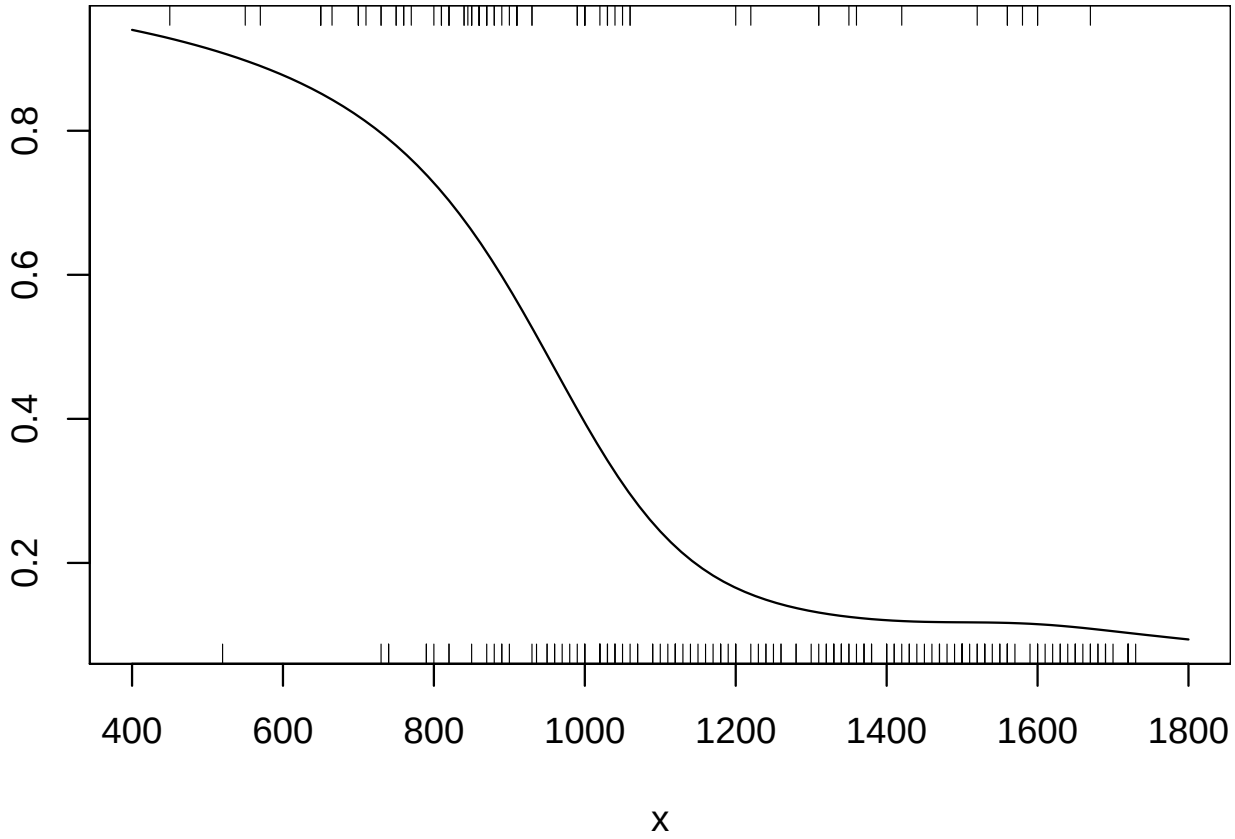
BPD data - modello stimato

```
curve(predict(fit,newdata=data.frame(birthweight=x)),
      from=400,to=1800,ylab="")
rug(bpd$birthweight[bpd$BPD==0],side=1)
rug(bpd$birthweight[bpd$BPD==1],side=3)
```



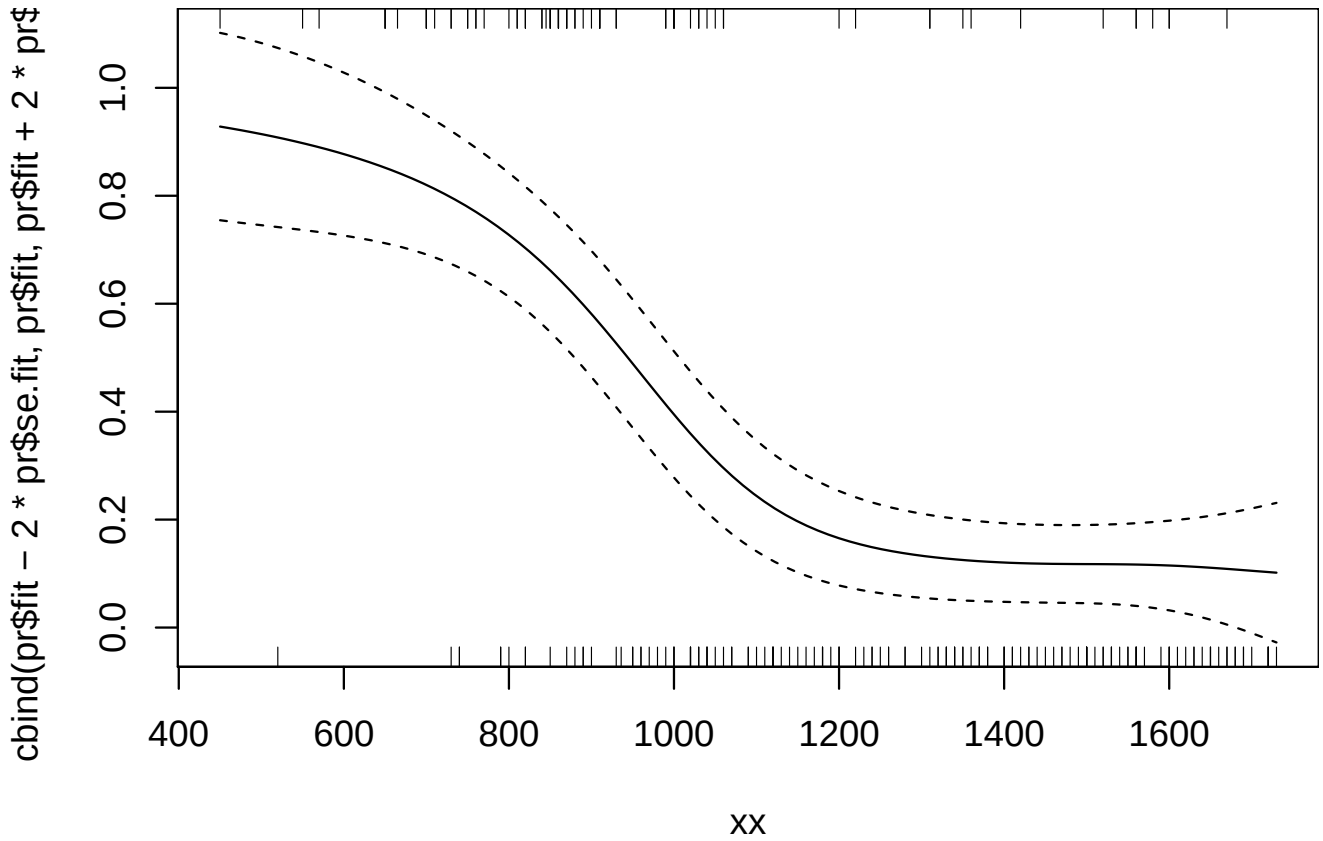
BPD data - modello stimato

```
curve(predict(fit,newdata=data.frame(birthweight=x),
              type="response"),
      from=400,to=1800,ylab="")
rug(bpd$birthweight[bpd$BPD==0],side=1)
rug(bpd$birthweight[bpd$BPD==1],side=3)
```



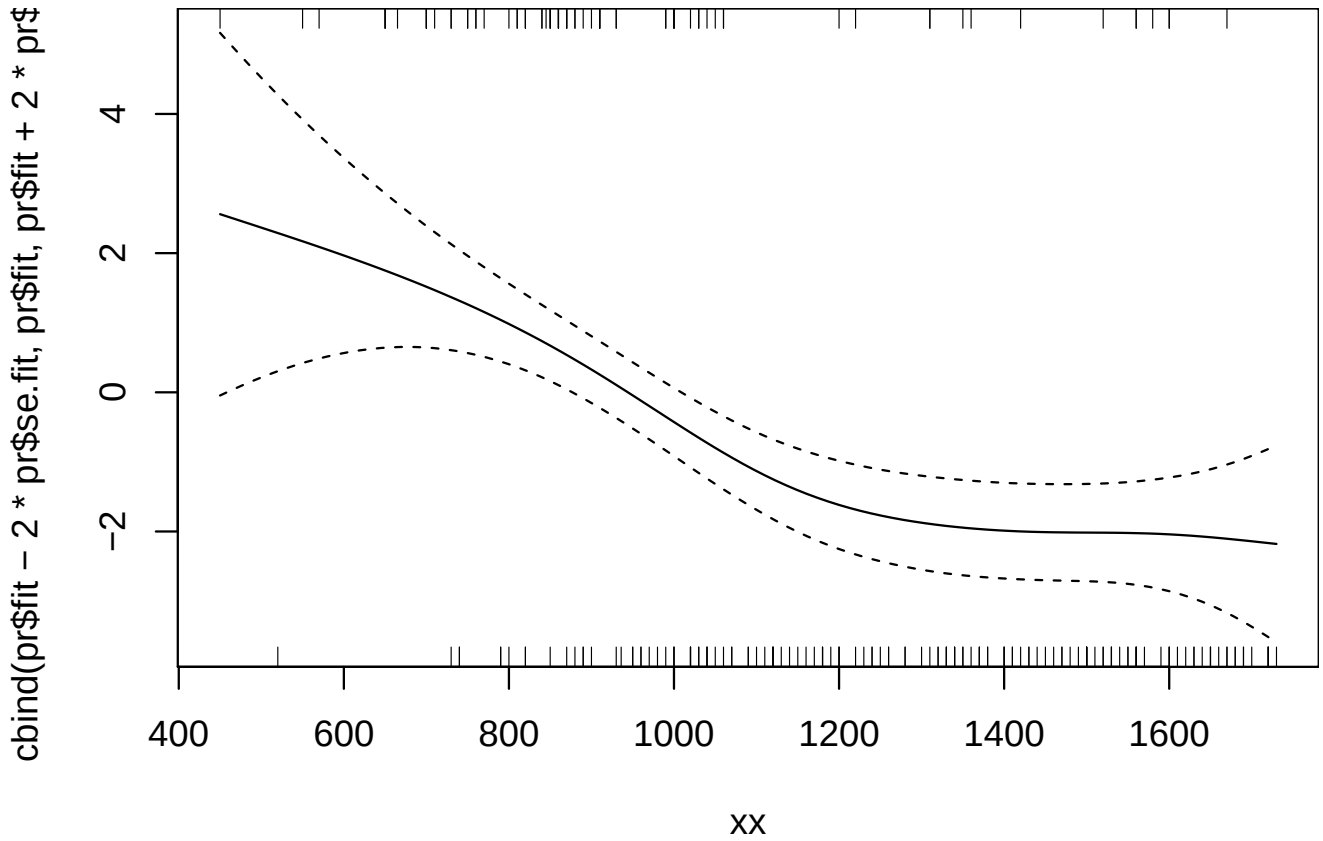
BPD data - modello stimato

```
xx=seq(450,1730,length=100)
pr=predict(fit,newdata=data.frame(birthweight=xx),
               type="response",se.fit=TRUE)
matplot(xx,cbind(pr$fit-2*pr$se.fit,pr$fit,pr$fit+2*pr$se.fit),
        type="l",lty=c(2,1,2),col="black")
```



BPD data - modello stimato

```
xx=seq(450,1730,length=100)
pr=predict(fit,newdata=data.frame(birthweight=xx),
               se.fit=TRUE)
matplot(xx,cbind(pr$fit-2*pr$se.fit,pr$fit,pr$fit+2*pr$se.fit),
        type="l",lty=c(2,1,2),col="black")
```



Regressione semiparametrica, i rischi