Apprendimento statistico e machine learning

Splines di regressione e lisciamento e GAM

Leonardo Egidi

Ottobre 2024

Università di Trieste

Splines di regressione

Funzioni di base

- Per avvicinarci alle splines, dobbiamo prima familiarizzare con il concetto di *funzione di base*.
- L'idea è quella di avere a disposizione una serie di funzioni o trasformazioni che possano venire applicate alla variabile X, $b_1(X), b_2(X), \ldots, b_K(X)$. Invece di stimare un modello lineare in X, stimiamo:

$$y_i = \beta_0 + \beta_1 b_1(x_i) + \beta_2 b_2(x_i) + \ldots + \beta_K b_K(x_i) + \epsilon_i,$$

  con basi $b_1(\cdot), b_2(\cdot), \ldots, b_K(\cdot)$ conosciute e fissate.
- Nella regressione polinomiale $b_j(x_i) = x_i^j$, nella regressione a gradini $b_j(x_i) = I(c_j \leq x_i < c_{j+1})$.
- Metodi di stima canonici, dei minimi quadrati ordinari.
- Domande:
  - *Come costruire le basi?*
  - *Quante basi?*

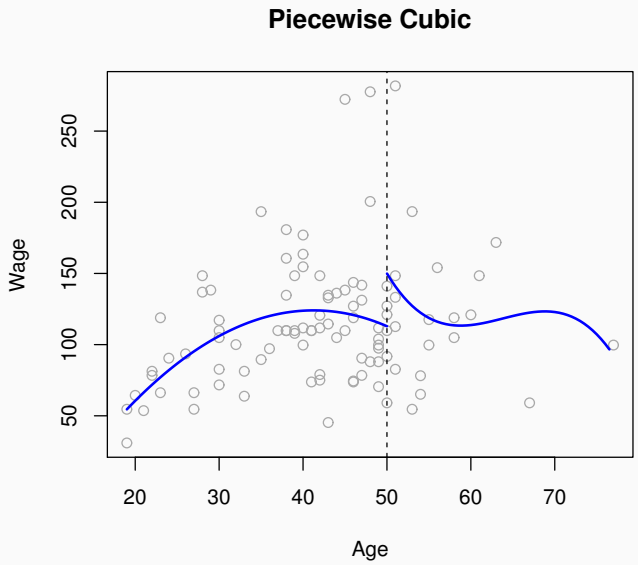
Polinomi a pezzi

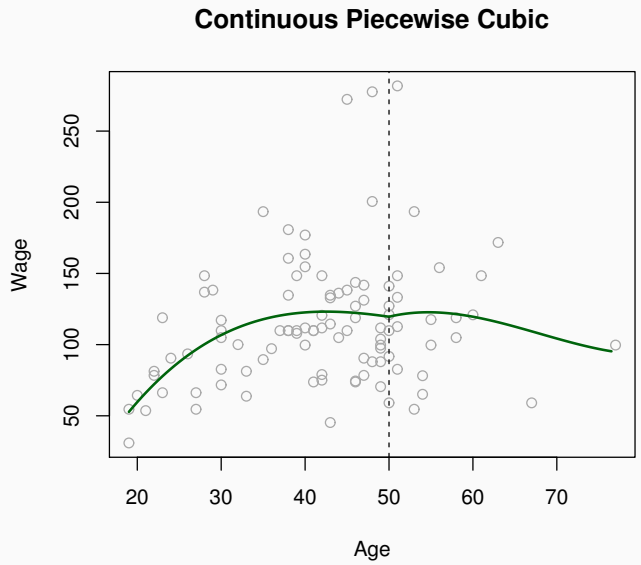
- Invece di usare un singolo polinomio sull'intero dominio di X, possiamo usare invece diversi polinomi in regioni definite dai *nodi*, ad esempio (con 1 nodo solo):

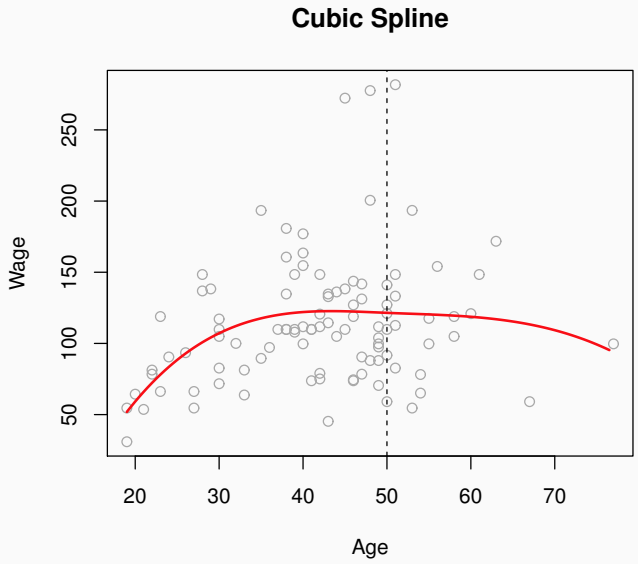
$$y_i = \begin{cases} \beta_{01} + \beta_{11} x_i + \beta_{21} x_i^2 + \beta_{31} x_i^3 + \epsilon_i & \text{if } x_i < c; \\ \beta_{02} + \beta_{12} x_i + \beta_{22} x_i^2 + \beta_{32} x_i^3 + \epsilon_i & \text{if } x_i \geq c. \end{cases}$$

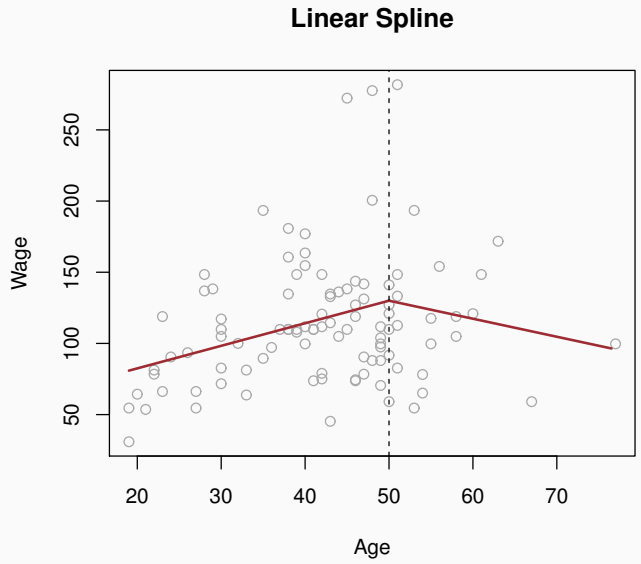
- E' meglio aggiungere vincoli ai polinomi, ad esempio quelli di continuità.
- Più nodi implicano più flessibilità. Con K nodi, avremo $K + 1$ distinti polinomi cubici (uno per sottosezione di X).
- Le *splines* hanno il 'massimo' ammontare di continuità.

Dati sui crediti









Dettagli della figura precedente

- I primi due grafici in alto (prima riga) mostrano polinomi a pezzi (*pieceweise*). Il primo mostra polinomi cubici *non vincolati*, mentre il secondo in alto a destra mostra polinomi cubici *vincolati* per essere continui nel valore `age = 50`.
- I secondi due grafici in basso (seconda riga) mostrano due splines. Nel grafico in basso a sinistra stimiamo un polinomio cubico a pezzi vincolato per essere continuo nel nodo `age = 50` e per avere derivata prima e seconda continue. Nel secondo grafico in basso a destra stimiamo invece un polinomio a pezzi lineare vincolato per essere continuo.
- Ogni vincolo che imponiamo 'libera' un *grado di libertà*, riducendo la complessità della risultante stima polinomiale a pezzi. Nel grafico in alto a sinistra abbiamo 8 gdl, mentre nel grafico in basso a sinistra imponiamo tre vincoli (continuità, continuità derivate prime e seconde), e abbiamo quindi solo 5 gdl. In generale una spline cubica con K nodi usa un totale di $4 + K$ gdl.

Splines lineari

Una spline lineare con nodi in ξ_k, $k = 1, 2, \ldots, K$ è un polinomio lineare a pezzi continuo in ogni nodo.

Possiamo rappresentare questo modello come:

$$y_i = \beta_0 + \beta_1 b_1(x_i) + \beta_2 b_2(x_i) + \ldots + \beta_{K+1} b_{K+1}(x_i) + \epsilon_i,$$

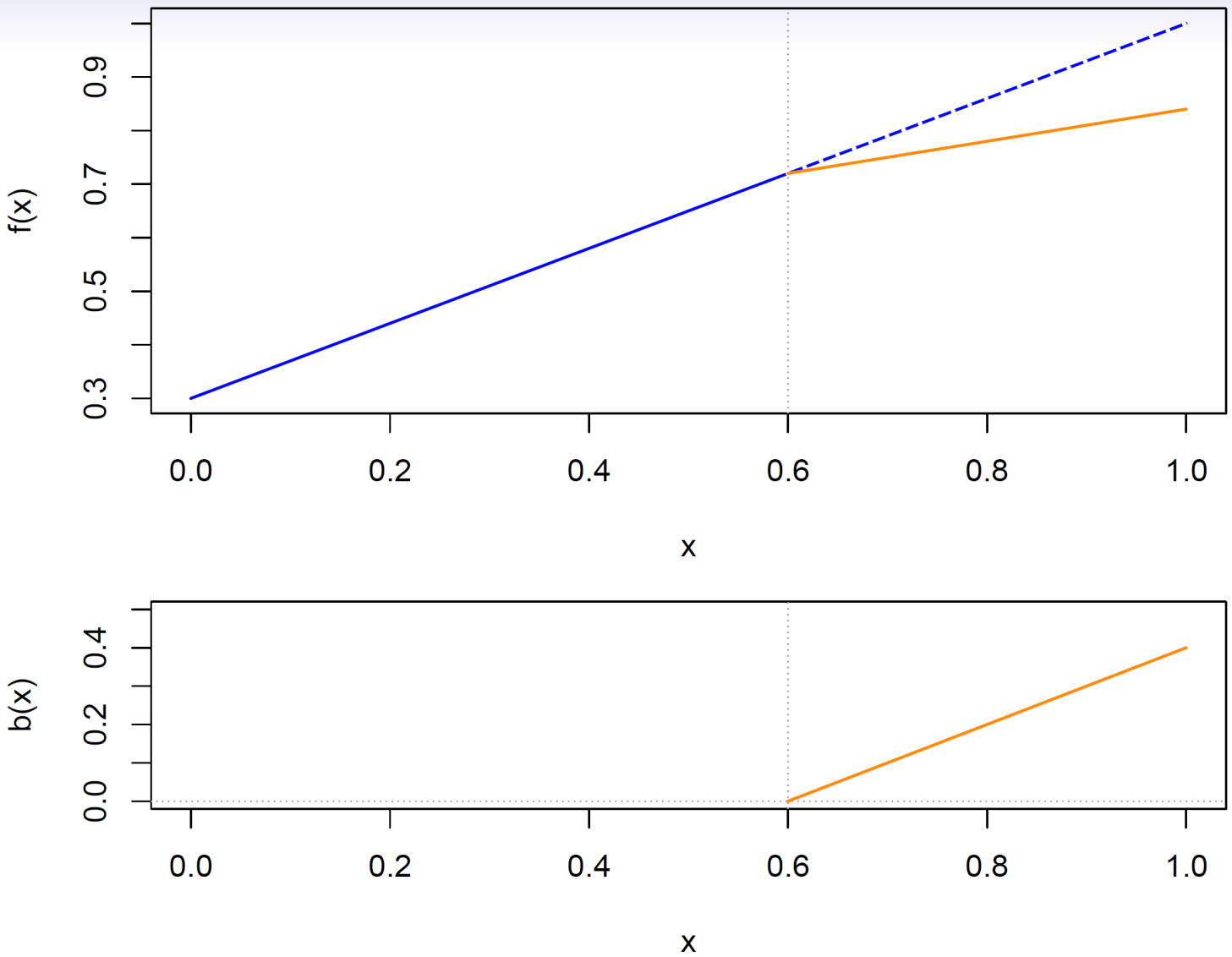
dove le b_k sono le *funzioni di base*:

$$\begin{aligned} b_1(x_i) &= x_i \\ b_{k+1}(x_i) &= (x_i - \xi_k)_+, \quad k = 1, \ldots, K. \end{aligned}$$

Qui il simbolo $()_+$ denota la *parte positiva*:

$$(x_i - \xi_k)_+ = \begin{cases} x_i - \xi_k & \text{if } x_i > \xi_k \\ 0 & \text{otherwise} \end{cases}$$

La parte positiva



Splines cubiche

Una spline cubica con nodi in ξ_k, $k = 1, 2, \ldots, K$ è un polinomio cubico a pezzi con derivate continue fino all'ordine 2 in ogni nodo.

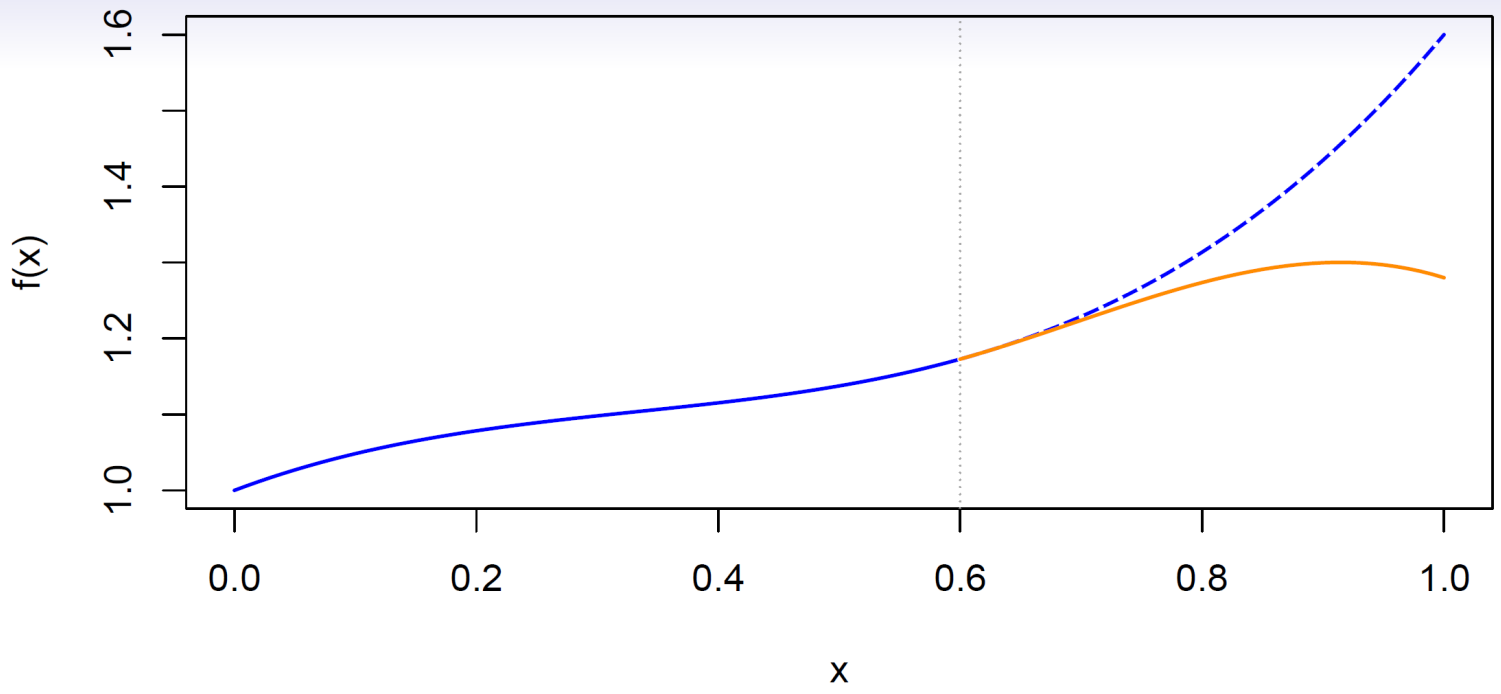
Possiamo rappresentare questo modello con *funzioni di basi troncate*:

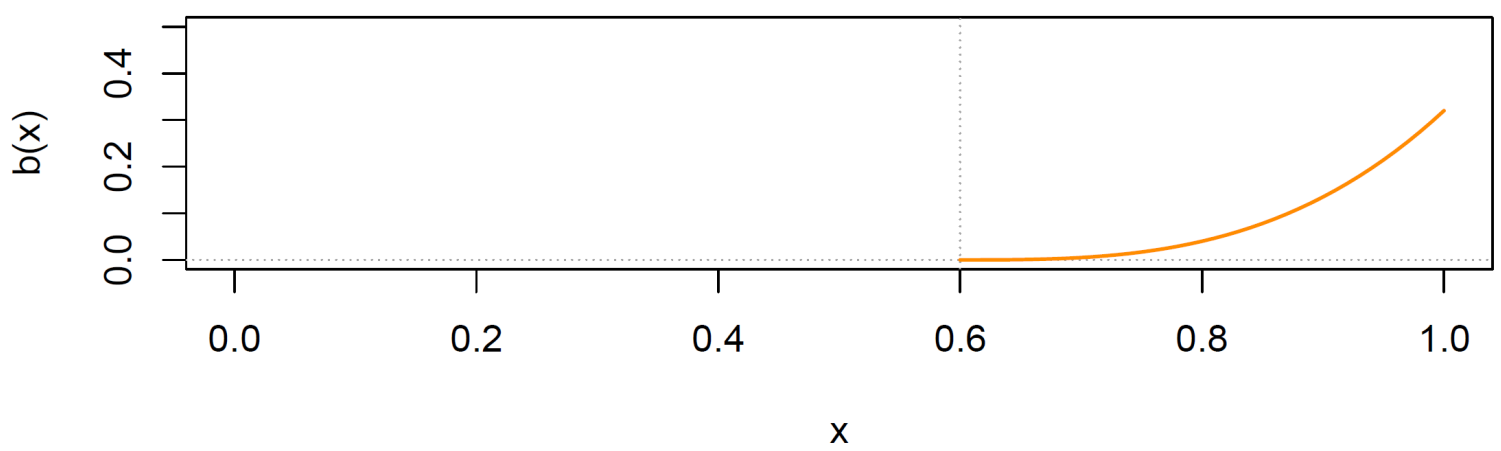
$$
\begin{aligned}
y_i = \ & \beta_0 + \beta_1 b_1(x_i) + \beta_2 b_2(x_i) + \ldots + \beta_{K+3} b_{K+3}(x_i) + \epsilon_i, \\
b_1(x_i) &= x_i \\
b_2(x_i) &= x_i^2 \\
b_3(x_i) &= x_i^3 \\
b_{k+3}(x_i) &= (x_i - \xi_k)_+^3, \quad k = 1, \ldots, K,
\end{aligned}
$$

dove la *funzione di base potenza troncata* è definita come:

$$
(x_i - \xi_k)_+^3 = \begin{cases} (x_i - \xi_k)^3 & \text{if } x_i > \xi_k \\ 0 & \text{otherwise} \end{cases}
$$

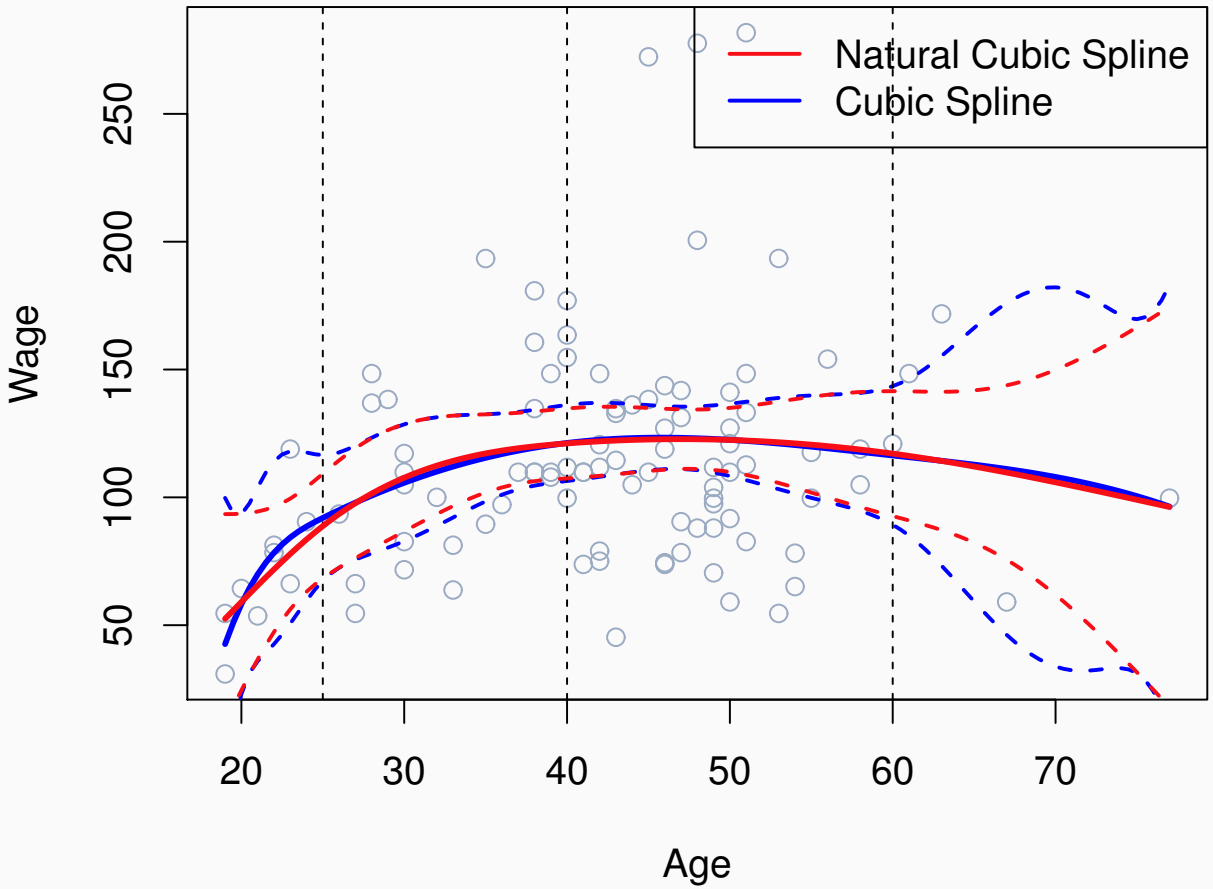
La parte positiva (cubica)





Splines naturali cubiche

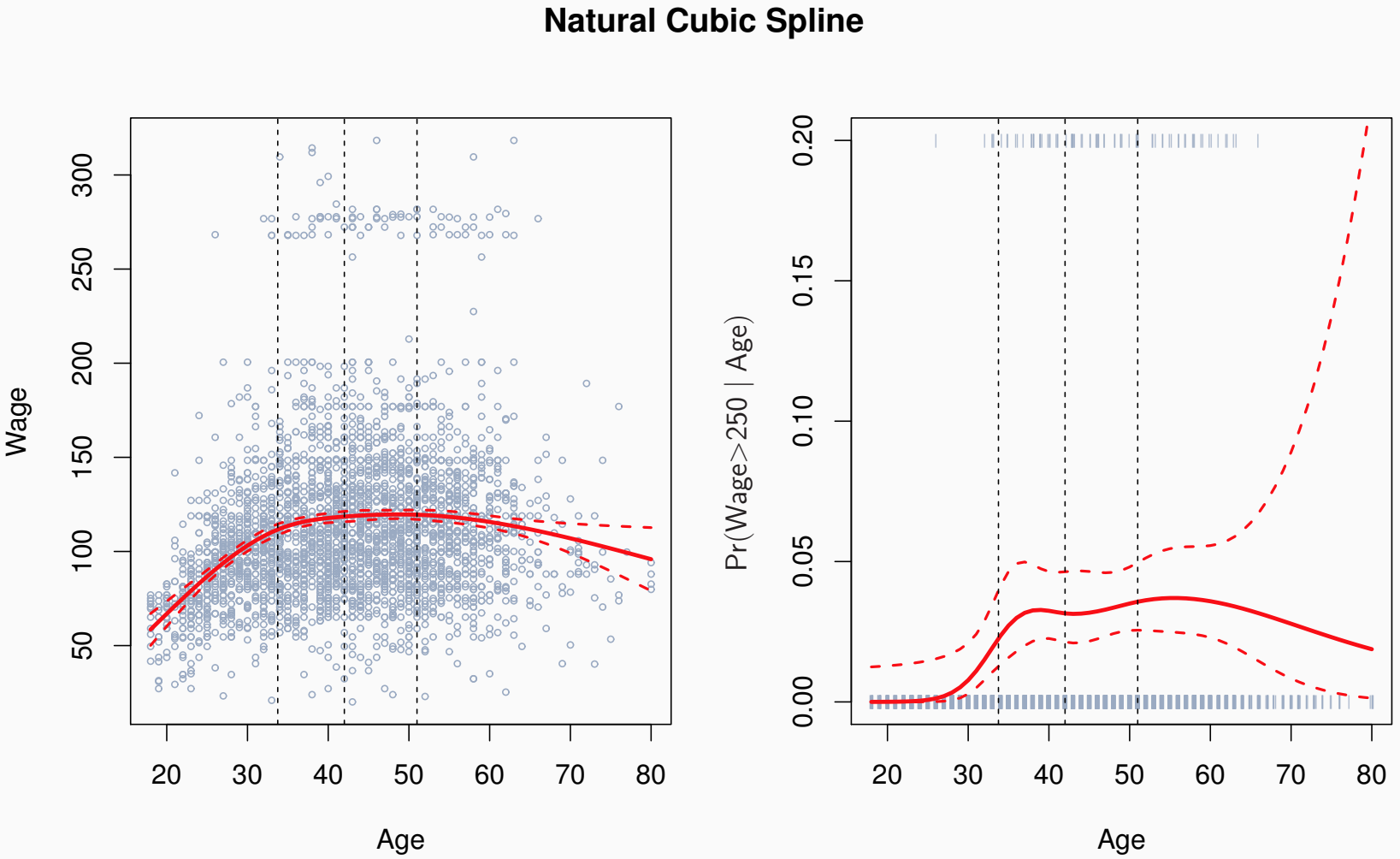
Una *spline naturale cubica* estrapola linearmente oltre i confini dei nodi. Questo aggiunge $4 = 2 \times 2$ extra vincoli, e ci consente di inserire più nodi interni per gli stessi gradi di libertà di una spline cubica regolare.



In R

- Stimare una spline in R è semplice: `bs(x,...)` per le splines di ogni grado, e `ns(x,...)` per splines naturali cubiche, nel pacchetto `splines`.
- Grafico slide seguente: spline naturale cubica con 4 gdl sui dati `Wage`. I 3 nodi sono stati fissati automaticamente al 25-esimo, 50-esimo e 75-esimo percentile della variabile `age`. *Perché ci sono 4 gdl? Consulta sezione 7.4.4 del libro ISL!*

Dati sui crediti: spline naturale cubica



Generalizziamo!

- *Una spline di grado d è un polinomio a pezzi di grado d, con continuità delle derivate fino all'ordine $d-1$ in ogni nodo.*
- La scelta delle basi di una spline determina il suo grado intrinseco di 'liscezza' e adattamento ai dati.
- Sfortunatamente le splines possono esibire una grande *varianza* agli estremi del dominio dei predittori. Ecco perché si usano spesso splines *naturali*, che piazzano dei vincoli aggiuntivi ai confini, ovverossia che la funzione sia lineare ai confini del range dei predittori (vedi grafico in slide 10).

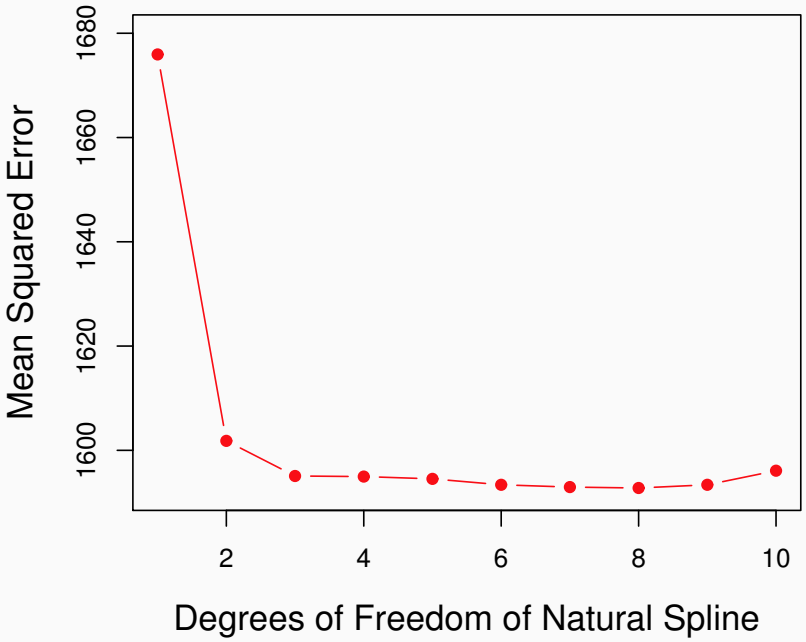
Piazzamento dei nodi: scelta usuale

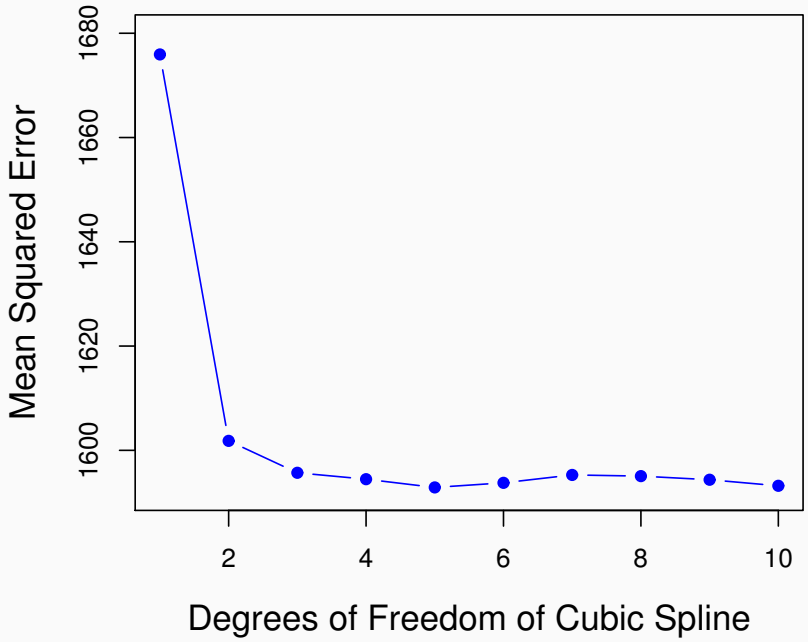
- Domanda ricorrente: *dove piazzare i nodi?* Di solito i nodi si mettono uniformemente sul range di X.
- Una strategia è fissare K, il numero dei nodi, per poi fissarli in corrispondenza di opportuni quantili di X.

Piazzamento dei nodi: cross-validation

- Una strategia in qualche modo più oggettiva è invece usare la cross-validazione.
- Si rimuove una porzione dei dati di training, si stima una spline con un certo numero di nodi sui dati rimanenti e poi si usa la spline stessa per fare previsione sulla porzione di dati *held-out*.
- Si ripete questo processo più volte fino a quando ogni osservazione sia stata tenuta fuori una volta, e si calcola la RSS cross-validata.
- La procedura può essere ripetuta per diversi valori di K, e il valore di K in corrispondenza alla RSS minima viene scelto (vedi slide seguente per i dati `Wage`, a sinistra slide cubica naturale, a destra slide cubica: *il fit lineare non è per nulla adeguato!*).

Dati sui crediti: scelta di K con cv

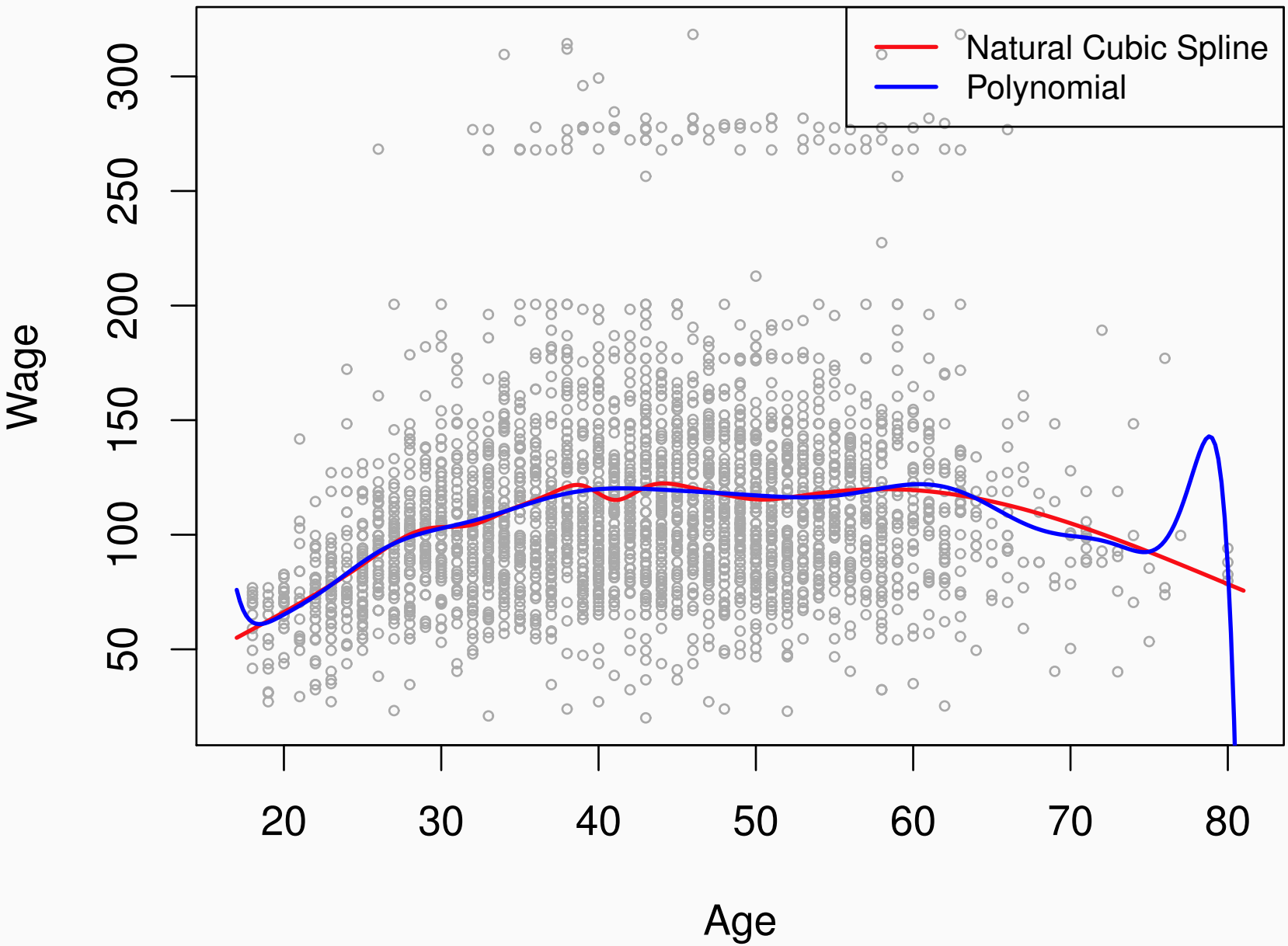




Piazzamento dei nodi: gradi di libertà

- Una spline cubica con K nodi ha $K + 4$ parametri o gradi di libertà.
- Una spline cubica naturale con K nodi ha K gradi di libertà.
- Grafico slide seguente: comparazione tra un polinomio di grado 14 (`poly(age, 14)`) e una spline naturale cubica, (`ns(age, 14)`) entrambi con 15 gdl. *Effetti indesiderati del polinomio ai margini di X!*
- A differenza dei polinomi, che devono usare un grado alto, le splines introducono flessibilità incrementando il numero di nodi ma mantenendo un grado fissato relativamente basso. *Le splnes forniscono generalmente risultati migliori dei polinomi!*

Dati sui crediti: spline naturale cubica e polinomio



Splines di lisciamento

Dove siamo

- Fino ad adesso, abbiamo parlato di *splines di regressione*, che consistono in:
  - scelta nodi
  - creazione basi
  - stima coefficienti con metodo minimi quadrati
- Introduciamo ora un metodo un po' diverso - diciamo 'regolarizzato'- per produrre una spline, dove si tiene conto anche della *liscezza* (smoothness) della curva di regressione.

Splines di lisciamento

- Nello stimare una funzione $g(\cdot)$ su alcuni dati, vogliamo spesso raggiungere due obiettivi: trovare una $g(\cdot)$ che renda RSS piccola e che sia *liscia* (smooth).
- Consideriamo questo criterio per stimare una funzione *liscia* $g(x)$ su alcuni dati:

$$\min_{g \in \mathcal{S}} \sum_{i=1}^{n} (y_i - g(x_i))^2 + \lambda \int g''(t)^2 dt.$$

- Il primo termine è RSS, misura adattamento di $g(x)$ ai dati ad ogni x_i.
- Il secondo termine è una *penalità di lisciamento* (roughness penalty) che controlla quanto ondulata è $g(x)$. Viene gestita tramite il *parametro di tuning* $\lambda \geq 0$.
  - Più piccolo è λ, più ondulata è la funzione, eventualmente interpolante y_i quando $\lambda = 0$.
  - Per $\lambda \to \infty$, la funzione $g(x)$ diviene lineare.

Splines di lisciamento: qualche dettaglio matematico

- $g(\cdot)$ che minimizza la formula sopra è detta *spline di lisciamento (smoothing spline)*.
- La derivata prima $g'(t)$ misura la pendenza della curva in t, mentre la derivata seconda $g''(t)$ ci dà una misura della concavità/convessità della curva, ci dice cioè quanto la pendenza della curva sta cambiando in t.
- La derivata seconda è quindi una funzione della *liscezza* della curva: sarà tanto più grande in valore assoluto quanto la curva sarà ondulata vicino t, e piccola quanto più liscia sarà.
- $\int g''(t)^2 dt$ è una misura della variazione totale della funzione $g'(t)$ su tutto intero il suo range.

Splines di lisciamento: la soluzione

- Si può matematicamente provare che la soluzione di tale minimizzazione è una *spline naturale cubica*, con un nodo per ogni valore unico di x_i. La penalità di lisciamento controlla ancora la 'liscezza' tramite λ.
- **Attenzione**: non è esattamente uguale alla spline naturale cubica che otterremmo con l'approccio di funzioni di base visto prima, con nodi in $x_1, \ldots, x_n$. Piuttosto, è una versione *regolarizzata (shrunken)* di quella spline cubica naturale, dove il valore del parametro di tuning λ controlla il livello di *shrinkage*.

Splines di lisciamento

Alcuni dettagli:

- Le splines di lisciamento evitano il problema della selezione dei nodi, con la scelta solamente di un unico λ.
- In `R` la funzione `smooth.spline()` stima una spline di lisciamento.
- Il vettore di n valori stimati può essere scritto come $\hat{\mathbf{g}}_\lambda = \mathbf{S}_\lambda \mathbf{y}$, dove $\mathbf{S}_\lambda$ è una matrice $n \times n$ (determinata da x_i e λ).
- I *gradi di libertà effettivi* sono dati da:

$$df_\lambda = \sum_{i=1}^{n} \{\mathbf{S}_\lambda\}_{ii}.$$

I gradi di libertà effettivi rappresentano il numero di *parametri liberi*. Nonostante una spline di lisciamento abbia n parametri, molti di questi sono pesantemente vincolati (*shrunken*). Più alto è df_λ e più la spline è flessibile (varianza alta, bias basso).

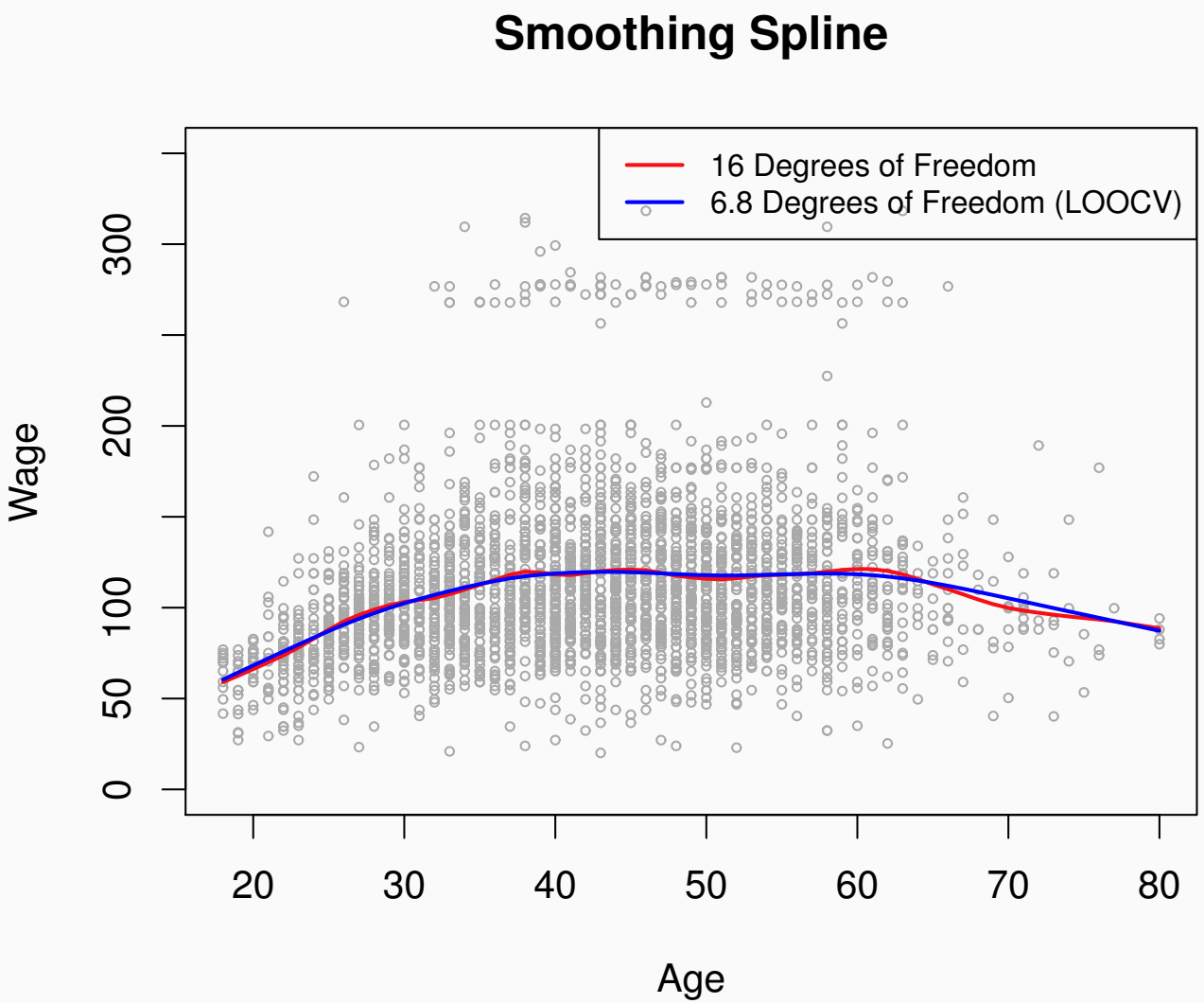
Splines di lisciamento - Scelta di λ

- Possiamo specificare *df* al posto di λ! In `R`: `smooth.spline(age, wage, df = 10)`.
- Quando stimiamo una spline di lisciamento, non serve specificare il numero e la posizione dei nodi! *Ma come stimiamo/scegliamo* λ?
- Possiamo usare l'approccio CV. L'errore di leave-one-out cross-validation è dato da:

$$\text{RSS}_{cv}(\lambda) = \sum_{i=1}^{n}(y_i - \hat{g}_{\lambda}^{(-i)}(x_i))^2 = \sum_{i=1}^{n}\left[\frac{y_i - \hat{g}_{\lambda}(x_i)}{1 - \{\mathbf{S}_{\lambda}\}_{ii}}\right]^2,$$

  dove $\hat{g}_{\lambda}^{(-i)}(x_i)$ indica il valore stimato per la spline in x_i, dove la stima usa tutte le osservazioni di training tranne la$i$-esima, (x_i, y_i). In `R`: `smooth.spline(age, wage)`.
- Con questa formula possiamo calcolare tutti gli errori di leave-one-out cv usando solamente $\hat{g}_{\lambda}$, la stima originale su *tutti i dati*.

Dati sui crediti: usare cv



Dettagli figura precedente

- Confronto tra una spline di lisciamento pre-specificata con 16 gdl (curva rossa) e una spline di lisciamento con λ scelto mediante CV (curva blu).
- Per questi dati c'è una trascurabile differenza tra le due splines, con quella con 16 gdl leggermente più 'selvaggia'. Quindi, per il rasoio di Occam, conviene scegliere quella blu, con meno gdl.

GAM

Generalized Additive Models

- Un modo naturale di estendere il canonico modello lineare multiplo

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip} + \epsilon_i$$

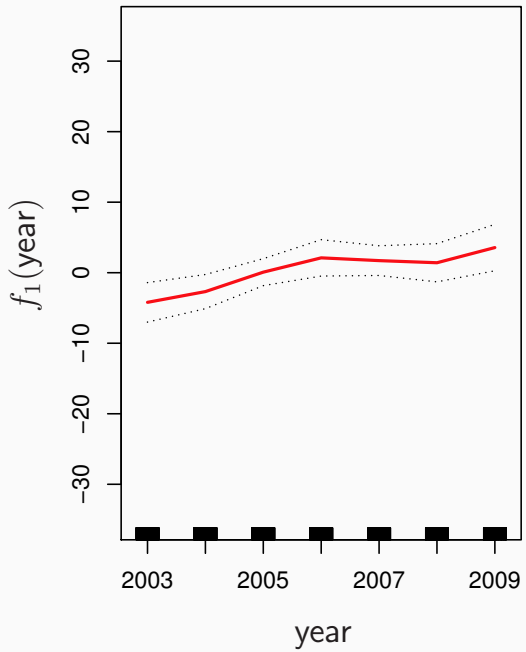
per prendere in considerazioni possibili effetti non lineari delle covariate sulla variabile risposta è quello di sostituire ad ogni componente lineare $\beta_j x_{ij}$ una *funzione liscia non lineare* *(smooth non-linear function)* $f_j(x_{ij})$.

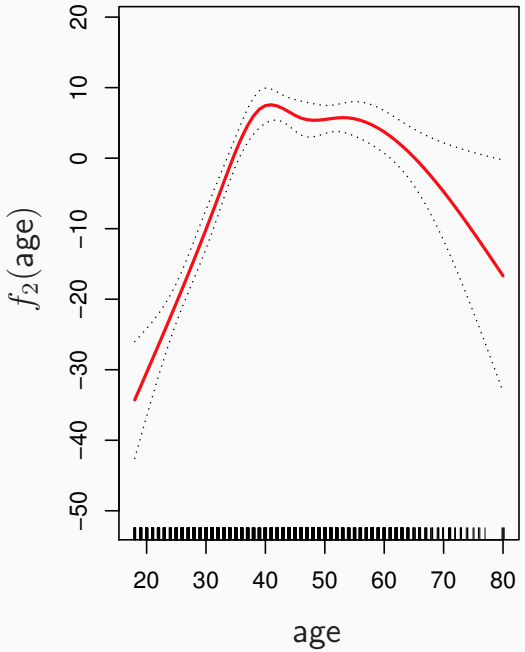
- Il modello diverrebbe quindi:

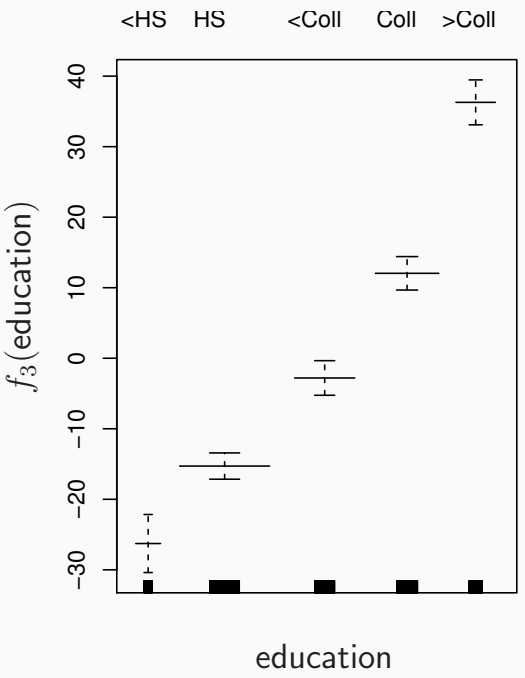
$$y_i = \beta_0 + f_1(x_{i1}) + f_2(x_{i2}) + \ldots + f_p(x_{ip}) + \epsilon_i = \beta_0 + \sum_{j=1}^{p} f_j(x_{ij}) + \epsilon_i.$$

- Questo è l'esempio di un *GAM (Modello additivo generalizzato)*, in cui la struttura dei predittori è ancora per l'appunto *additiva*.
- La bellezza dei GAM è nella loro flessibilità: possiamo usare lo 'scheletro' dei GAM per introdurre un numero desiderato di predittori e stimare splines per ciascuno o parte di questi.

Dati sui crediti: GAM



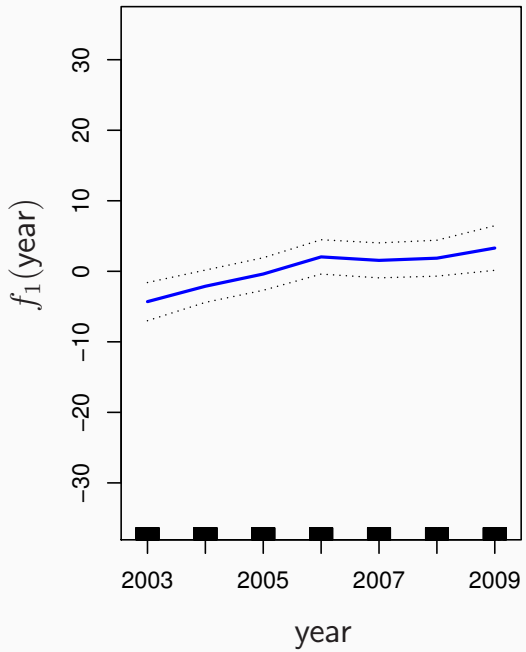


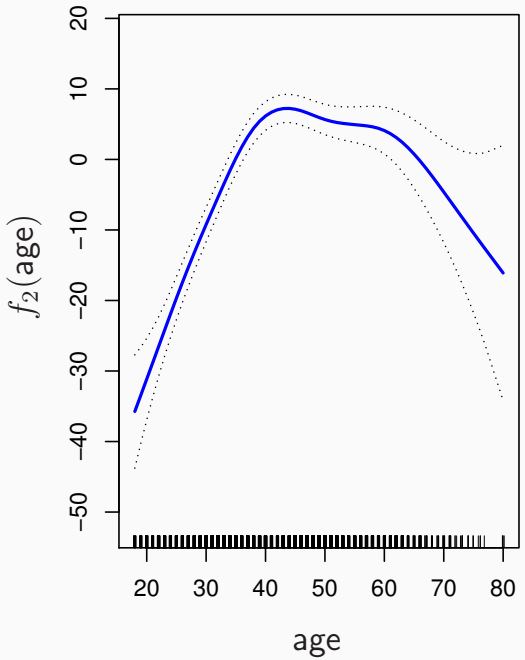


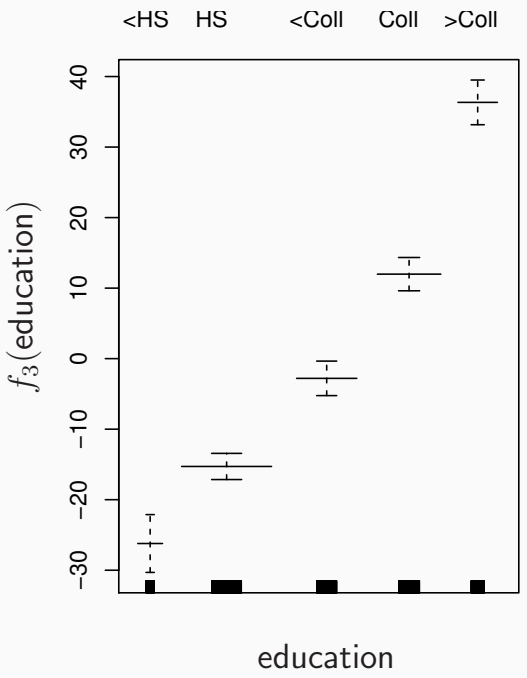
Dettagli della figura precedente

- $\texttt{wage} = \beta_0 + f_1(\texttt{year}) + f_2(\texttt{age}) + f_3(\texttt{education}) + \epsilon$.
- Grafico della relazione tra ogni predittore e la variabile risposta `wage`, con anche una misura di standard error per ciascuna variabile considerata. Le prime due funzioni sono splines naturali per `year` e `age`, con rispettivamente 4 e 5 gdl. La terza è una funzione a gradini per la variabile qualitativa `education`.
- L'intero modello è quindi una regressione con variabili espresse mediante splines e con variabili dummy, tutte 'impacchettate' all'interno di una matrice di predittori.
- Interpretazione: nel grafico a sinistra, tenendo le variabili `age` ed `education` fissate, `wage` tende a crescere leggermente con `year`. Nel grafico centrale, tenendo le variabili `year` ed `education` fissate, `wage` tende a essere più grande per valori intermedi di `age`, e più bassa per individui molto giovani e molto anziani. Nel grafico a destra si evince che il salario cresce con l'educazione.

Dati sui crediti: GAM







Dettagli della figura precedente

- Stesso modello di slide 29, con la differenza che stavolta le due funzioni f_1 e f_2 sono splines di lisciamento, con rispettivamente 4 e 5 gdl.
- Stimare un modello GAM con splines di lisciamento non è agevole come stimare un GAM con splines naturali, siccome i minini quadrati non sono disponibili!
- Tuttavia, con la funzione `gam()` del pacchetto `gam`, si può usare un cosiddetto approccio di *backfitting*: questo metodo stima un modello con predittori multipli aggiornando ripetutamente la stima per ciascun predittore uno alla volta, tenendo gli altri fissati. La bellezza di questo approccio è che di fatto usa i *residui parziali* per la variabile di interesse.
- Interpretazioni sulle variabili pressoché identiche a quelle condotte per il modello con splines naturali.

Dettagli computazionali in R

- Per fittare un semplice GAM usando splines naturali si può usare il comando:

  `lm(wage ~ ns(year, 5) + ns(age, 5) + education)`

- I coefficienti stimati non saarannno di per sé interessanti, piuttosto lo saranno le funzioni stimate. Produrre plots come quelli nelle slides precedenti con la funzione `plot.gam()` dei pacchetti `mgcv` o `gam`.
- Si possono combinare effetti lineari e non lineari, e poi usare il comando `anova` per comparare modelli.
- Per usare smoothing splines, funzione `gam` del pacchetto `gam`:

  `gam(wage ~ s(year, 5) + s(age, 5) + education)`

- Si possono usare anche regressioni locali con la funzione `lo`, per esempio `lo(age, span = .5)`
- I GAM sono additivi, tuttavia si possono includere interazioni della forma `ns(age, 5):ns(year, 5)`.

Algoritmo di *backfitting*

Dato il modello additivo nella forma:

$$Y = \beta_0 + \sum_{j=1}^{p} f_j(X_j) + \epsilon,$$

supponiamo il seguente criterio di penalizzazione per la stima:

$$\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} f_j(x_{ij}) \right)^2 + \sum_{j=1}^{p} \lambda_j \int f_j''(t_j)^2 dt_j,$$

dove i $\lambda_j \geq 0$ sono dei parametri di tuning. Si può mostrare che la soluzione che minimizza questo criterio è una spline cubica additiva, ovverossia ogni f_j è una spline cubica in X_j con nodi in ognuno dei valori unici di x_{ij}. Tuttavia, senza ulteriori restrizioni sul modello, la soluzione non è unica! Ecco perché si usa il seguente algoritmo di *backfitting*.

Algoritmo di *backfitting*

1. Inizializzare $\beta_0 = n^{-1} \sum_{i=1}^{n} y_i, \quad \hat{f}_j \equiv 0, \ \forall i, j.$
2. Data $\mathcal{S}_j$ una spline di lisciamento cubica per la componente X_j, applicare per ogni $j = 1, 2, \ldots, p$:

$$\hat{f}_j \leftarrow \mathcal{S}_j \left[\{ y_i - \hat{\beta}_0 - \sum_{k \neq j} \hat{f}_k(x_{ik}) \}_i^n \right],$$

$$\hat{f}_j \leftarrow \hat{f}_j - n^{-1} \sum_{i=1}^{n} \hat{f}_j(x_{ij}),$$

fino a quando la funzione $\hat{f}_j$ cambia meno di una pre-specificata soglia.

GAM per la classificazione

- I GAM possono essere utilizzati anche in casi in cui Y è qualitativa (se non proprio dicotomica).
- Se $p(X) = \Pr(Y = 1|X)$, possiamo scrivere:

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + f_1(X_1) + f_2(X_2) + \ldots + f_p(X_p).$$

(Capiremo nei dettagli la *ratio* di questa specificazione quando introdurremo la regressione logistica).

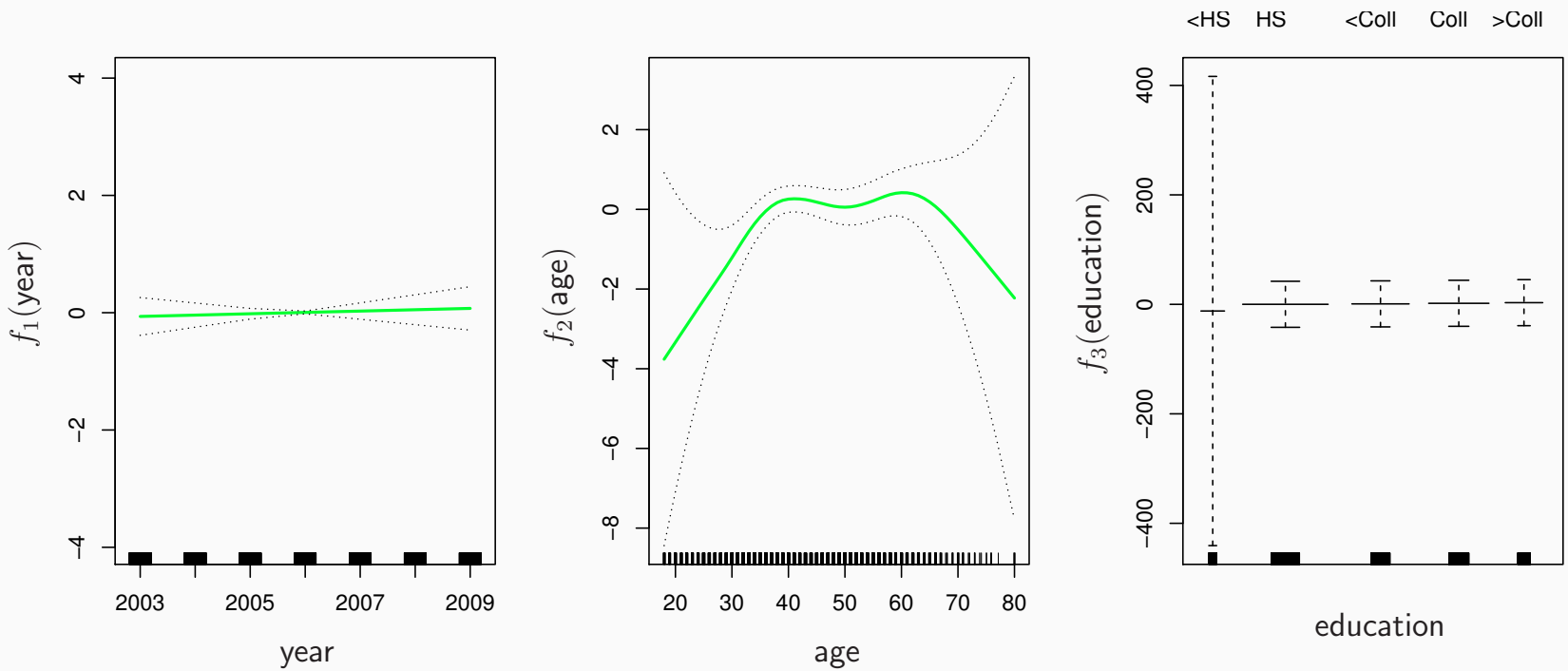
- Posso stimare un GAM per i dati `Wage` nel seguente modo:

```
gam(I(wage > 250) ~ year + s(age; df = 5) + education;
                    family = binomial)
```

dove

$$p(X) = \Pr(\texttt{wage} > 250|\texttt{age}, \texttt{year}, \texttt{education})$$

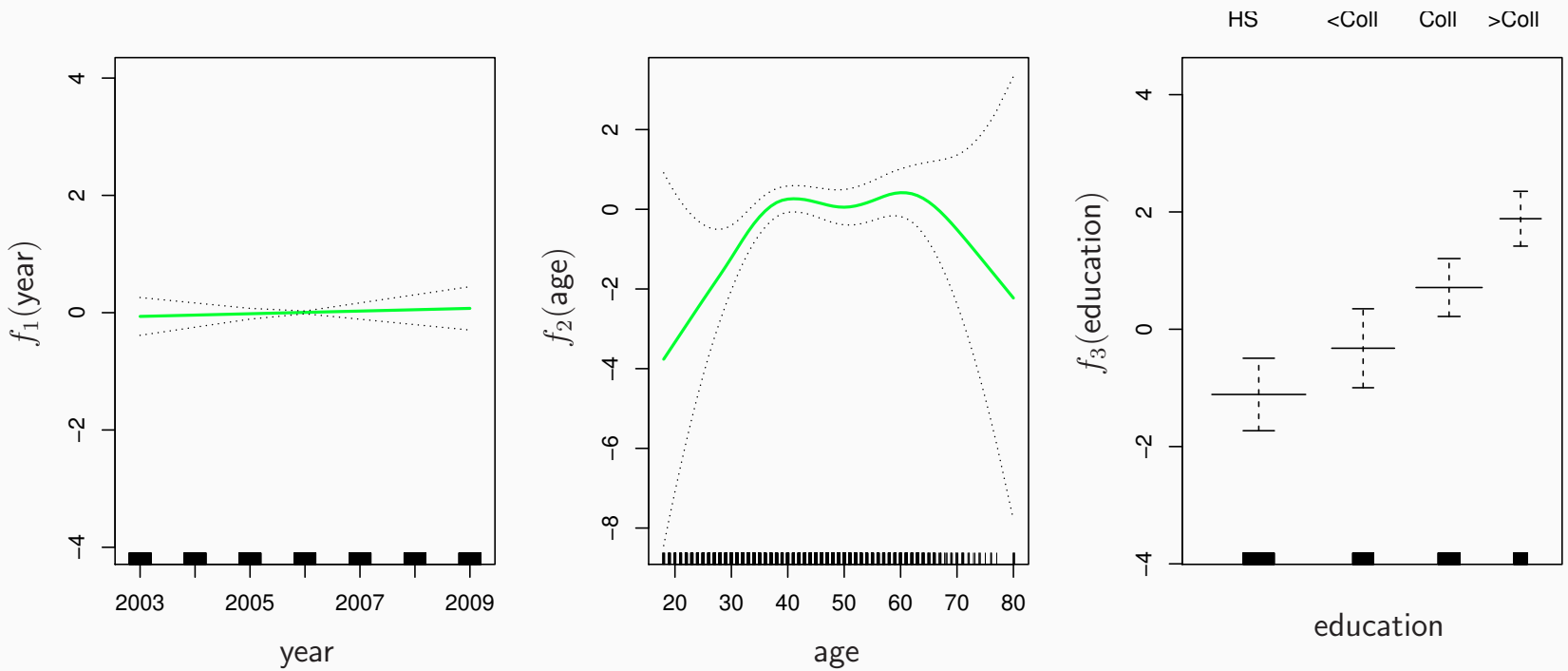
Dati sui crediti: GAM per classificazione



Dettagli della figura precedente

- f_2 è una spline di lisciamento con 5 gdl, mentre f_3 è una funzione a gradini.
- Strano comportamento nel terzo pannello. . . non ci sono individui per la categoria di `education < HS` che guadagnino più di 250,000$ dollari all'anno! Possiamo allora ristimare il GAM escludendo tali individui (slide seguente).

Dati sui crediti: GAM per classificazione



Vantaggi e svantaggi dei GAM

- ▼ I GAM consentono l'uso di funzioni non lineari f_j per ogni X_j per modellare eventuali effetti non lineari tra le X e la Y: non dobbiamo quindi inserire manualmente alcuna trasformazione delle variabili.
- ▼ Le stime non lineari producono previsioni probabilmente più accurate per la variabile risposta Y.
- ▼ Siccome il modello è additivo, possiamo ancora valutare l'*effetto marginale* di ogni X_j su Y tenendo fisse tutte le altre variabili.
- ▼ La 'liscezza' delle funzioni f_j per la variabile X_j può essere espressa tramite i gdl.
- ▼ Possiamo flessibilmente inserire nei GAM tutte le strutture di regressione viste finora, come ad esempio regressione polinomiale, a gradini, splines, regressione locale, etc.
- ▼ Il modello GAM è vincolato ad essere additivo. Tuttavia, come nei LM possiamo includere interazioni della forma $X_j \times X_k$, e/o funzioni di interazione del tipo $f_{jk}(X_j, X_k)$, usando *smoothers* bidimensionali.