Apprendimento statistico e machine learning

Regressione polinomiale e locale

Leonardo Egidi

Ottobre 2024

Università di Trieste

Oltre la linearità

La realtà non è (quasi) mai lineare!

Ma spesso l'assunzione di linearità è sufficiente per cogliere alcuni dettagli della realtà.

Quando quest'assunzione non basta, abbiamo diverse scelte:

- polinomi
- funzioni a gradini (step)
- splines
- regressione locale
- modelli additivi generalizzati.

Questi sono tutti strumenti che offrono grande flessibilità. Ci soffermeremo su qualcuno di questi, mostrando come conservino spesso la facilità di implementazione e interpretazione dei modelli lineari.

Regressione polinomiale

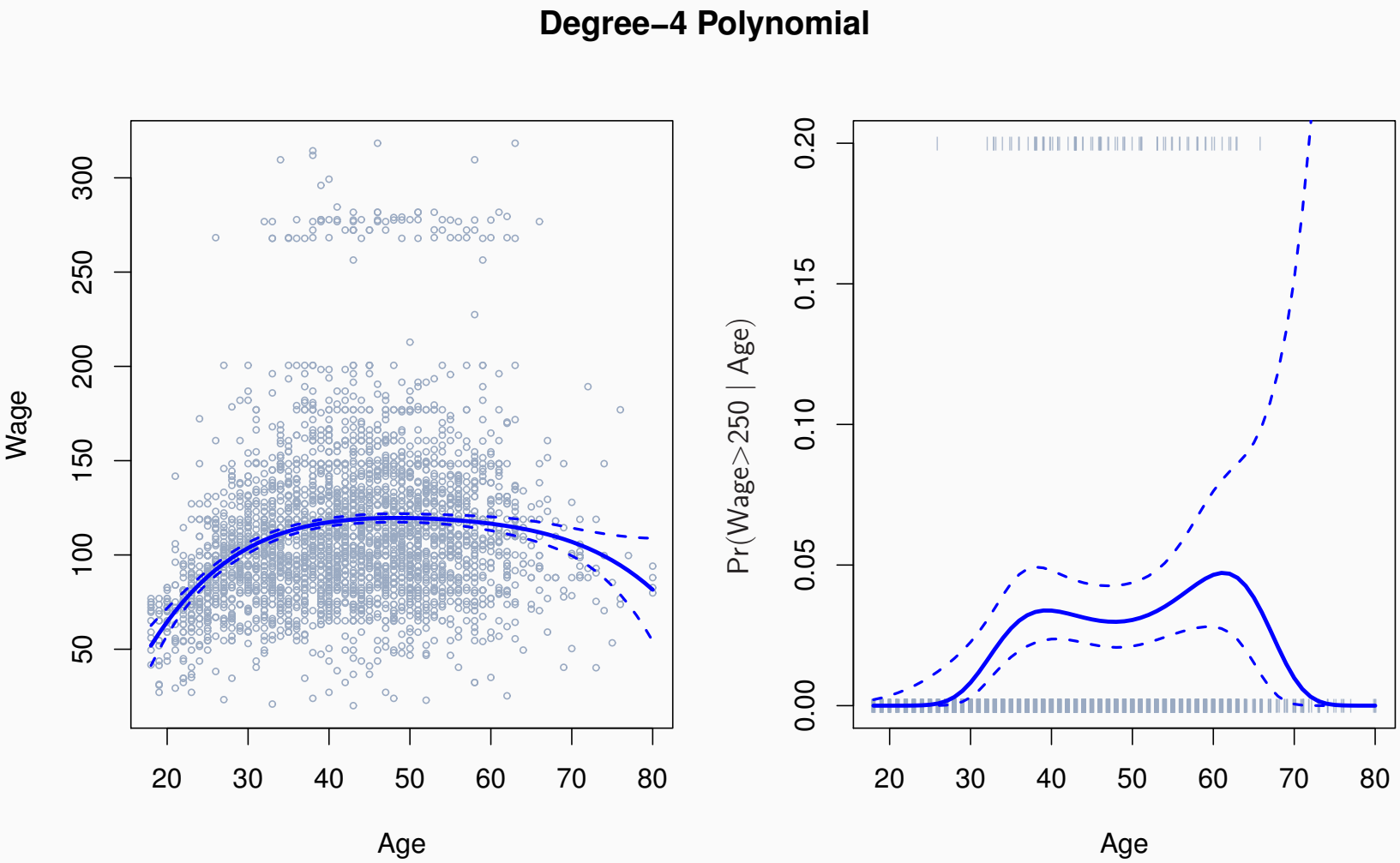
Regressione polinomiale

- Lo storico modo di estendere la regressione lineare è quello di inserire un polinomio di grado d:

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \ldots + \beta_d x_i^d + \epsilon_i,$$

  dove ϵ è la solita componente d'errore.
- Parliamo in questo caso di *regressione polinomiale*. Per un grado d abbastanza alto, una regressione di questo tipo produce una curva piuttosto non lineare.
- I coefficienti $\beta_0, \beta_1, \ldots, \beta_d$ possono venir stimati usando i minimi quadrati ordinari, perché si tratta in realtà di un modello lineare standard con predittori $x_i, x_i^2, x_i^3, \ldots, x_i^d$.
- Generalmente è inusuale che d sia maggiore di 3 o 4, questo per evitare fenomeni di *overfitting* o strani comportamenti in prossimità del supporto di X.

Dati sui salari



Dettagli sulla figura precedente

- Dati `Wage`, che contengono il salario e informazioni demografiche per maschi residenti nella regione centrale atlantica degli Stati Uniti.
- Sinistra: la curva blu è un polinomio di quarto grado per la variabile `wage` (in migliaia di dollari) in funzione della variabile dipendente `age` stimato coi minimi quadrati. Le curve tratteggiate denotano un intervallo di confidenza stimato del 95%.
- Destra: modelliamo l'evento dicotomico `wage>250` usando una regressione logistica, ancora con un polinomio di quarto grado. La probabilità stimata che il salario superi i 250,000$ è mostrata in blu, assieme a un intervallo di confidenza stimato.

Dettagli

- Creiamo nuove variabili $X_1 = X$, $X_2 = X^2$, etc. e trattiamo il modello come una regressione lineare multipla.
- Non siamo davvero interessati ai coefficienti in sè, ma piuttosto al valore stimato della funzione polinomiale ad un dato valore reale x_0:

$$\widehat{f}(x_0) = \hat{\beta}_0 + \hat{\beta}_1 x_0 + \hat{\beta}_2 x_0^2 + \hat{\beta}_3 x_0^3 + \hat{\beta}_4 x_0^4.$$

- Siccome $\widehat{f}(x_0)$ è una funzione lineare dei coefficienti $\hat{\beta}_l$ si può avere una misura della *varianza puntuale* della stima $\text{Var}[\widehat{f}(x_0)]$. Possiamo usare le stime delle varianze della procedura dei minimi quadrati per ciascuno dei coefficienti $\hat{\beta}_l$ per calcolare lo standard error puntuale stimato di $\widehat{f}(x_0)$: se $\widehat{\mathbf{C}}$ è una matrice di varianza-covarianza stimata per i $\hat{\beta}$, e se $\ell_0^T = (1, x_0, x_0^2, x_0^3, x_0^4)$, allora $\text{Var}[\widehat{f}(x_0)] = \ell_0^T \widehat{\mathbf{C}} \ell_0$. Quelle che mostriamo nel grafico sono le curve $\widehat{f}(x_0) \pm 2 \times \text{se}[\widehat{f}(x_0)]$.
- Per scegliere d si può ricorrere a cross-validazione, o fissarlo a qualche valore relativamente piccolo.

Dettagli (continua)

- La regressione logistica segue naturalmente. Per esempio, nella figura modelliamo:

$$\Pr(y_i > 250|x_i) = \frac{\exp\{\beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \ldots \beta_d x_i^d\}}{1 + \exp\{\beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \ldots \beta_d x_i^d\}}.$$

- Per calcolare gli intervalli di confidenza calcoliamo gli estremi inferiore e superiore sulla scala logit e poi invertiamo la relazione sulla scala probabilistica.
- Questo si può fare per diverse variabili.
- *Caveat*: notoriamente i polinomi hanno code pesanti. Quindi non rappresentano una buona procedura per estrapolare/fare previsioni.
- Si può fittare la curva polinomiale con la semplice funzione `y~poly(x, degree = 3)`.

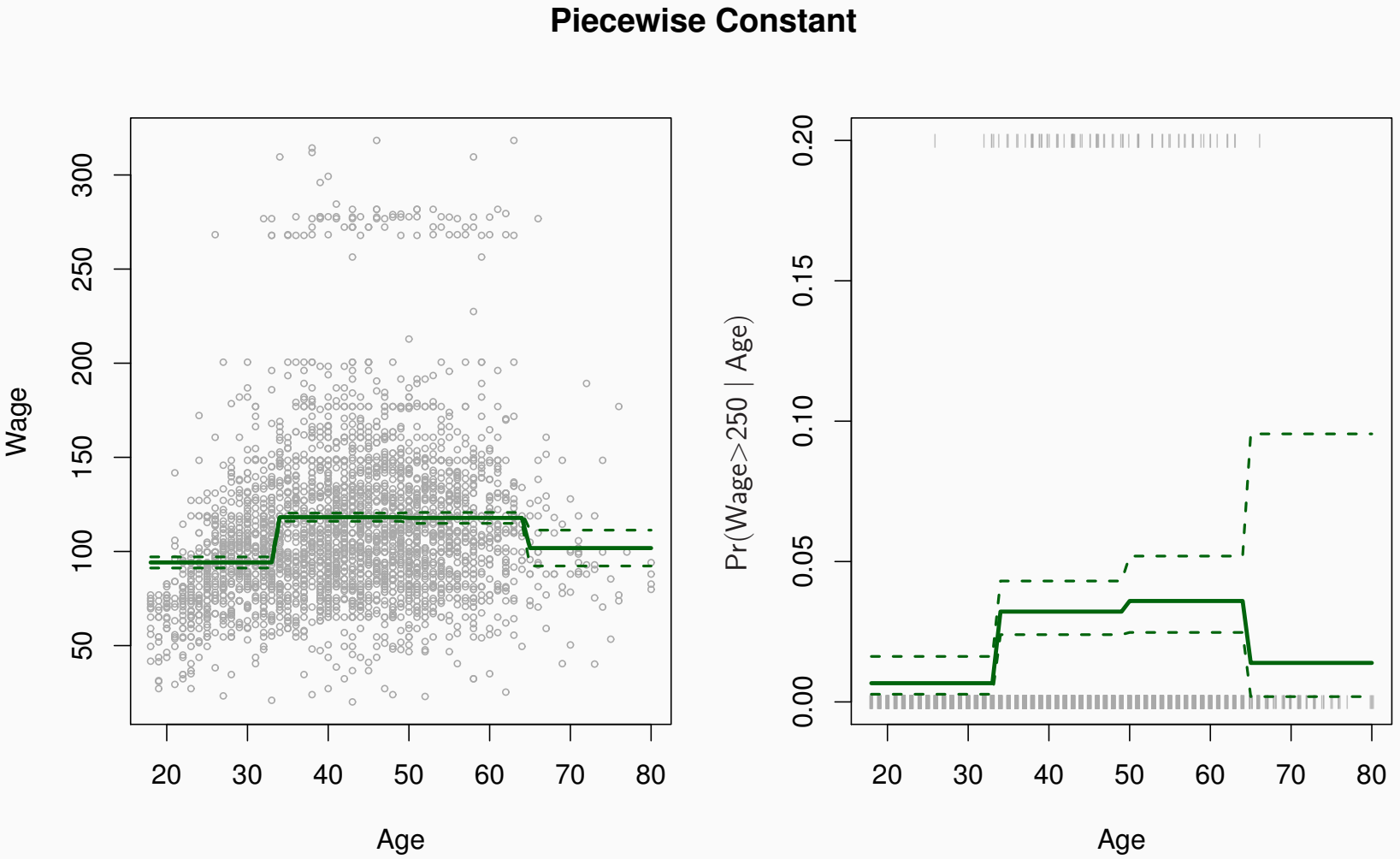
Funzioni a gradini (step functions)

- I polinomi impongono una struttura *globale* della funzione di X.
- Possiamo usare *funzioni a gradini* per evitare di imporre troppi vincoli: dividiamo il supporto di X in sezioni e stimiamo una costante per ciascuna di queste. Convertiamo una variabile continua in una *variabile categorica ordinale*.
- Creiamo dunque dei "punti di svolta" (cutpoints) $c_1, c_2, \ldots, c_K$ nel dominio di X, e poi costruiamo $K + 1$ nuove variabili:

$$
\begin{aligned}
C_0(X) &= I(X < c_1) \\
C_1(X) &= I(c_1 \leq X < c_2) \\
&\ldots \\
C_{K-1}(X) &= I(c_{K-1} \leq X < c_K) \\
C_K(X) &= I(c_K \leq X),
\end{aligned}
$$

dove $I(\cdot)$ è l'usuale funzione indicatore che restituisce 1 se la condizione è vera, e 0 altrimenti.

Dati sui salari



Dettagli sulla figura precedente

- Dati `Wage` sui salari.
- Sinistra: la curva verde è una stima dei minimi quadrati per la variabile `wage` (in migliaia di dollari) usando funzioni a gradini della variabile dipendente `age`. Le curve tratteggiate denotano un intervallo di confidenza stimato del 95%.
- Destra: modelliamo l'evento dicotomico `wage>250` usando una regressione logistica, ancora con funzioni a gradini di `age`. La probabilità stimata che il salario superi i 250,000$ è mostrata in verde, assieme a un intervallo di confidenza stimato.

Dettagli

- Notare che, per ogni valore di X, $C_0(X) + C_1(X) + \ldots + C_K(X) = 1$, siccome X deve stare esattamente in uno dei $K + 1$ intervalli.
- Usiamo i minimi quadrati ordinari per stimare un modello lineare usando $C_1(X), C_2(X), \ldots, C_K(X)$ come predittori:

$$y_i = \beta_0 + \beta_1 C_1(x_i) + \beta_2 C_2(x_i) + \ldots + \beta_K C_K(x_i) + \epsilon_i.$$

- Quando $X < c_1$, tutti i predittori sono uguali a zero, quindi β_0 può essere interpretato come il valore medio per Y quando $X < c_1$.
- Analogamente, sicome l'equazione sopra prevede una risposta di tipo $\beta_0 + \beta_j$ per $c_j \leq X < c_{j+1}$, allora ogni β_j rappresenta l'incremento medio nella risposta per X quando $c_j \leq X < c_{j+1}$ rispetto a quando $X < c_1$.

Dettagli computazionali

- Si tratta di una metodologia semplice: si creano tante variabili *dummy* per ciascuno dei gruppi.
- Utile modo per creare interazioni facili da interpretare. Per esempio l'effetto interazione tra le variabili `Year` e `Age`:

$$I(\texttt{Year} < 2005) \times \texttt{Age}, \quad I(\texttt{Year} \geq 2005) \times \texttt{Age}$$

  consente l'inclusione di diverse funzioni lineari per ognuna delle categorie dell'età.
- In `R`: `I(Year < 2005)` oppure `cut(age, c(18,25,40,65,90))`.
- La scelta dei cutpoints o dei *nodi* (knots) può essere problematica. Per creare non linearità, anziché avere delle costanti per ogni intervallo, ci sono delle alternative più lisce e regolari come ad esempio le *splines*.

Regressione locale

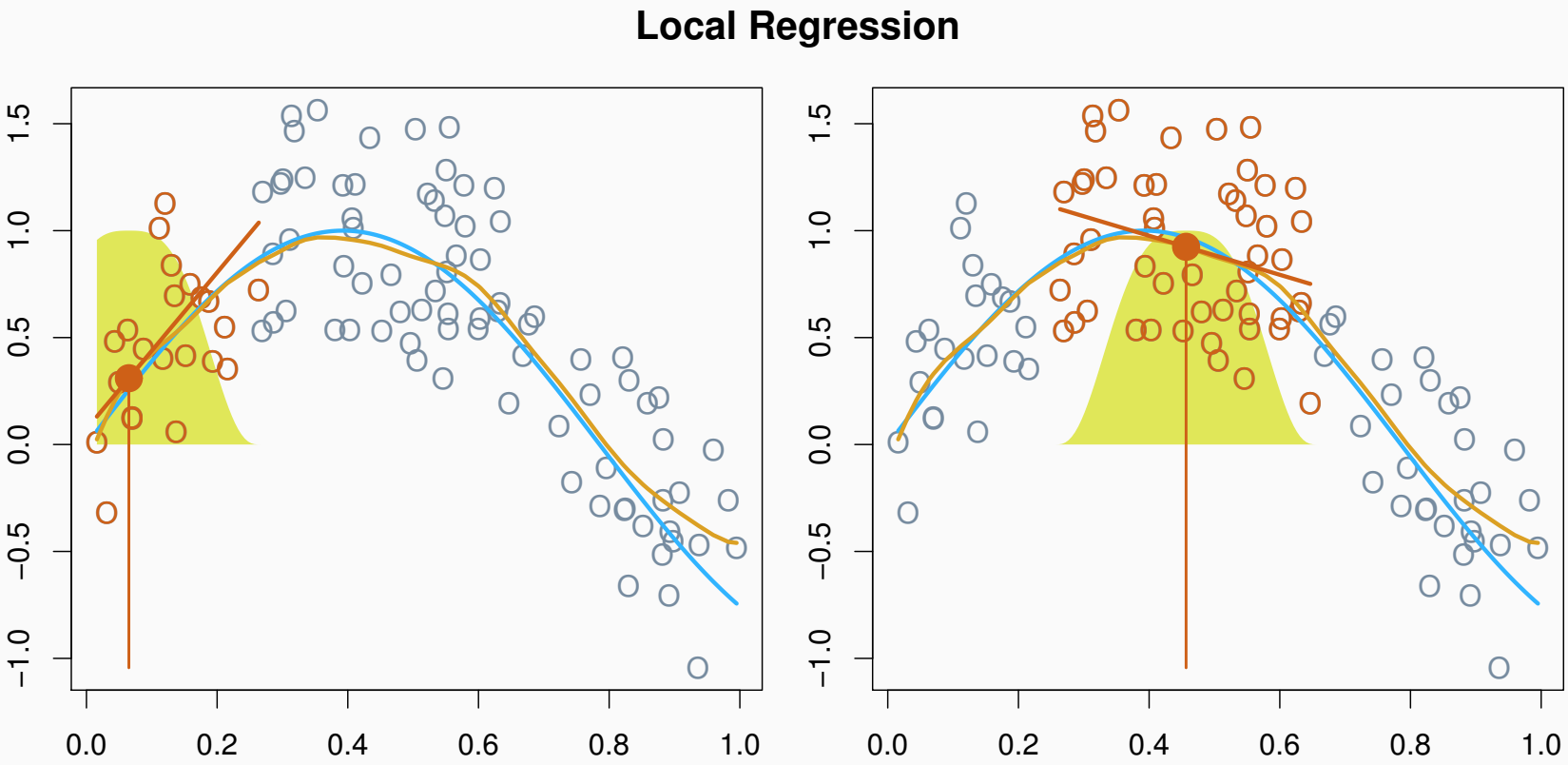
Regressione locale

- La *regressione locale* è un approccio differente per stimare in modo flessibile funzioni non lineari. Consiste nel calcolare la stima di un punto *target* x_0 usando solamente le osservazioni di training nelle *vicinanze*.
- Prima di continuare con la naturale prosecuzione della regressione polinomiale classica, quindi la regressione polinomiale *a pezzi* o *splines*, facciamo un interludio con questo metodo che appartiene ai cosiddetti algoritmi di stima della densità (*kernel density estimation*). Si tratta appunto di metodi locali, che condividono con le splines l'idea di *dividere* lo spazio delle covariate in sezioni convenienti per la stima.
- Per maggiori e approfonditi dettagli consultare il libro *The Elements of Statistical Inference* by Hastie, Tibshirani and Friedman).

Regressione locale

- L'idea di massima è la seguente: stimare una funzione di regressione $f(X)$ su un dominio $\mathbb{R}^p$ stimando un semplice modello separato per ognuno dei punti di interesse x_0, di modo tale che la risultante funzione stimata $\widehat{f}(X)$ sia opportunamente *liscia* (smooth).
- La stima si ottiene mediante la scelta di una cosiddetta funzione *kernel* $K_{i0}(x_0, x_i)$ che assegni un peso al punto x_i basato sulla sua distanza da x_0.
- I kernel sono solitamente indicizzati mediante un parametro che detta la larghezza degli intervalli di stima da considerare. Si tratta dell'unico parametro da stimare dal set di training.

Esempio su dati simulati



Dettagli figura precedente

- Due punti target utilizzati, vicino a 0.4 e vicino a 0.05.
- La curva blu rappresenta la funzione $f(x)$ *vera* da cui i dati simulati sono stati generati, mentre la curva arancione chiara corrisponde alla stima di regressione locale $\widehat{f}(x)$. I punti arancioni colorati "vuoti" sono locali rispetto al punto target x_0, rappresentato dalla linea verticale arancione.
- L'area gialla a campana indica i pesi assegnati a ciascuno dei punti, che decrescono mano a mano che ci si sposta dal punto target.
- La stima $\widehat{f}(x_0)$ in x_0 è stata ottenuta con un metodo di stima dei minimi quadrati pesati (segmento arancione), usando quindi il valore stimato in x_0 (punto arancione "pieno") come stima $\widehat{f}(x_0)$.

Algoritmo di regressione lineare locale in $X = x_0$

1. Ottenere la frazione $s = k/n$ di punti di training set, secondo cui gli x_i sono i più vicini a x_0.
2. Assegnare un peso $K_{i0} = K(x_i, x_0)$ ad ogni punto in questo intervallo, di modo che il punto più lontano da x_0 abbia peso zero, e il più prossimo a x_0 abbia il peso più grande. Tutti tranne questi k *vicini più vicini* (nearest neighbors) hanno peso zero.
3. Stimare una regressione ai minimi quadrati pesati di y_i rispetto a x_i usando i pesi summenzionati, trovando $\hat{\beta}_0$ e $\hat{\beta}_1$ che minimizzano:

$$\sum_{i=1}^{n} K_{i0}(y_i - \beta_0 - \beta_1 x_i)^2.$$

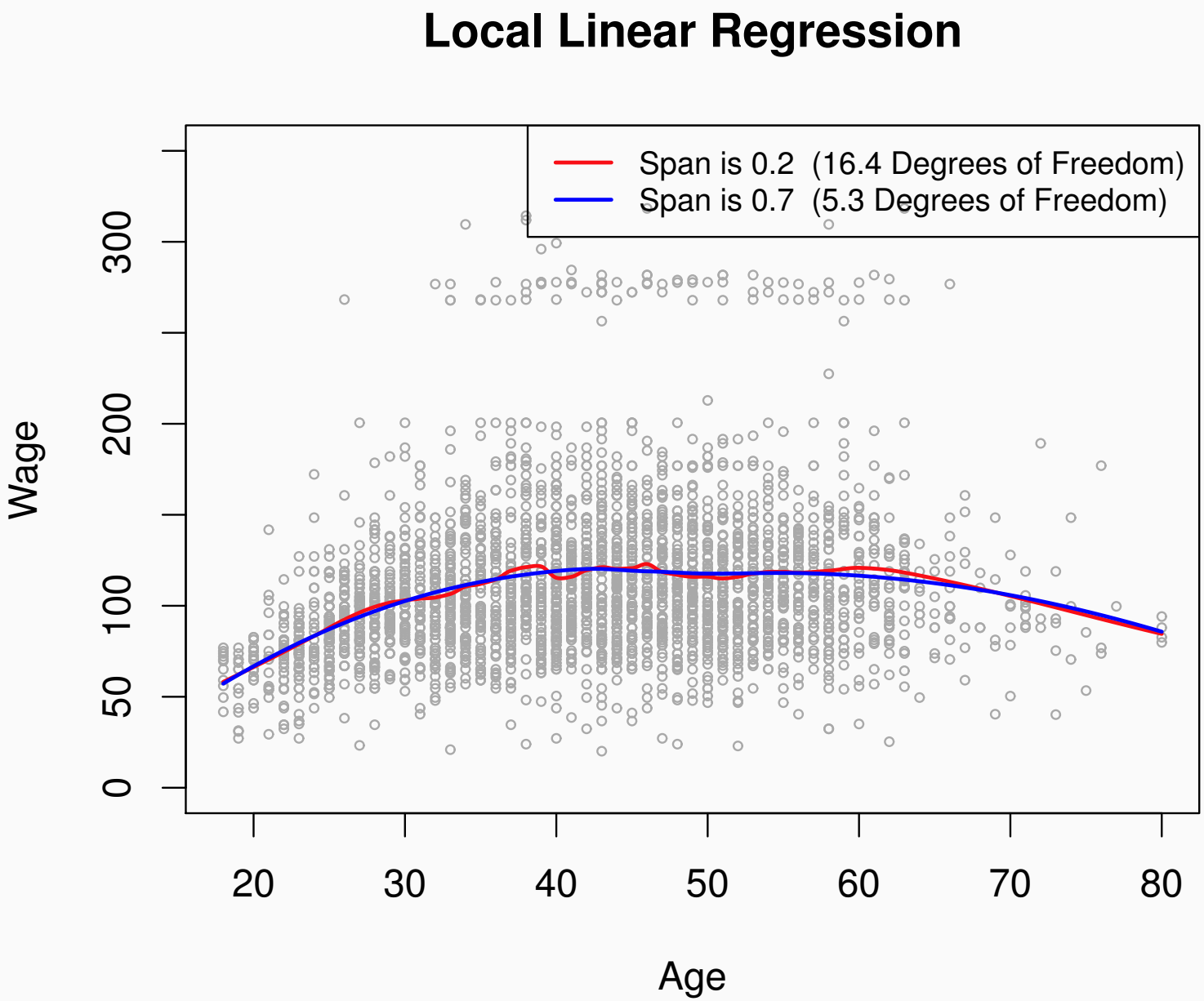
4. Il valore stimato in x_0 è dato da $\widehat{f}(x_0) = \hat{\beta}_0 + \hat{\beta}_1 x_0$.

Scelte di tuning

Per applicare una regressione di tipo locale ci sono alcune scelte obbligate da fare:

- scegliere la funzione kernel K;
- scegliere se per ogni punto target stimare una regressione lineare, costante o quadratica;
- stabilire *l'estensione (span)* s, che controlla la flessibilità della stima non lineare. Più piccolo è s, più *locale* e ondulata (o poco liscia) sarà la nostra stima; viceversa, un s grande condurrà a una stima globale *liscia* usando tutti i dati di training.

Dati sui salari



Dettagli sulla figura precedente

- Regressione lineare locale sui dati `Wage`, che contengono una serie di covariate potenzialmente associate ai salari di un gruppo di maschi residenti nella regione atlantica degli Stati Uniti.
- Due valori per s: quando $s = 0.7$, la curva di regressione è molto più *liscia* rispetto a quando $s = 0.2$.

Estensioni

- La regressione locale può essere estesa in molte direzioni.
- In questa breve trattazione abbiamo dato l'intuizione per la regressione lineare locale, stimando regressioni lineari per ogni punto di interesse e sfruttando l'influenza di punti vicini. Ma perché limitarci alla regressione lineare?
- Potremmo ipotizzare una *regressione polinomiale locale*, assumendo polinomi locali per ogni punto target, il che porta alla seguente minimizzazione della funzione in x_0 (in sostituzione al punto 3 dell'algoritmo di slide 17):

$$\min_{\beta \in \mathbb{R}^{d+1}} \sum_{i=1}^{n} K_{i0}(x_0, x_i) \left[y_i - \beta_0 - \sum_{j=1}^{d} \beta_j x_i^j \right]^2,$$

  con soluzione $\widehat{f}(x_0) = \hat{\beta}_0 + \sum_{j=1}^{d} \hat{\beta}_j x_0^j$.
- Miglioramento: la regressione locale lineare tende a essere distorta nelle regioni di curvatura della funzione vera, mentre quella polinomiale tende a correggere questo *bias* (vedi figura seguente).

Dati simulati: regressione polinomiale locale