

I modelli additivi

Data Mining

CLAMSES - University of Milano-Bicocca

Aldo Solari

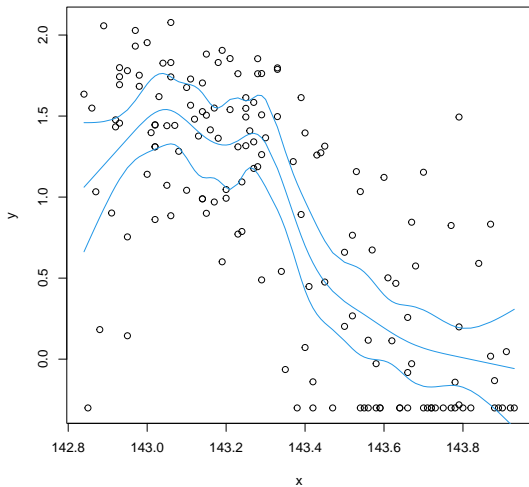
Riferimenti bibliografici

- Bowman, Evers. Lecture Notes on Nonparametric Smoothing. § 4.4
- AS § 4.5
- LKA § 6.3
- HTF § 9.1.1, 9.1.2

Dati della Grande Barriera Corallina

È stata effettuata un'indagine sulla fauna del fondale marino compreso tra la costa del Queensland settentrionale e la Grande Barriera Corallina. La regione di campionamento copriva una zona chiusa alla pesca commerciale, nonché zone limitrofe dove era consentita la pesca.

La relazione tra il punteggio di cattura (Score1) e la longitudine (Longitude) è di particolare interesse perché, in questa posizione geografica, la costa corre all'incirca da nord a sud e quindi la longitudine è un proxy per la distanza al largo. Potremmo quindi ragionevolmente aspettarci che l'abbondanza di vita marina cambia con la longitudine.



Spline cubica naturale con $K = 9$ nodi (interni)

È raro avere problemi che coinvolgono solo una singola covariata. Per i dati della Grande Barriera Corallina l'estensione naturale è esaminare la relazione tra il punteggio di cattura (y) e la latitudine (x_1) e longitudine (x_2), in un modello

$$y_i = f(x_{1i}, x_{2i}) + \varepsilon_i$$

Un modello additivo ha la forma

$$y_i = f_1(x_{1i}) + f_2(x_{2i}) + \varepsilon_i$$

dove le funzioni componenti f_1 e f_2 descrivono gli effetti separati e additivi di le due covariate

Per stimare

$$y_i = \beta_0 + f_1(x_{1i}) + f_2(x_{2i}) + \varepsilon_i$$

ri-arrangiamo i termini:

$$y_i - \beta_0 - f_2(x_{2i}) = f_1(x_{1i}) + \varepsilon_i$$

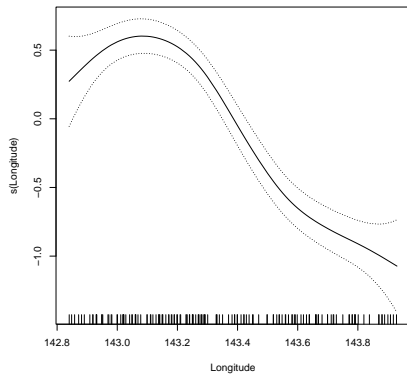
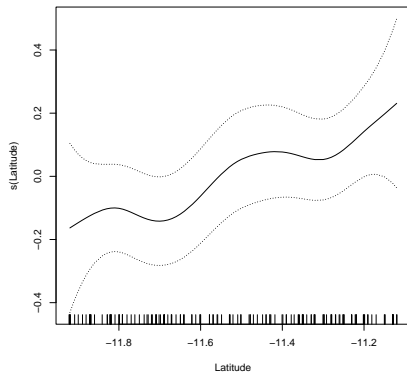
Ciò suggerisce di stimare f_1 con una funzione regolare (*smooth*) s_1 :

$$\hat{f}_1 = s_1(y - \bar{y} - \hat{f}_2 \sim x_1)$$

e analogamente, (ri)-stimare f_2 come

$$\hat{f}_2 = s_2(y - \bar{y} - \hat{f}_1 \sim x_2)$$

La ripetizione di questi passaggi fornisce una forma semplice dell'algoritmo di *backfitting*. La stessa idea si applica quando abbiamo più di due componenti sul modello.



Il modello additivo per la regressione prevede che il valore atteso della variabile risposta sia una somma di contributi parziali, uno per ciascuna variabile predittiva

$$\mathbb{E}(Y|X = \mathbf{x}) = \beta_0 + \sum_{j=1}^p f_j(x_j)$$

Ciò include il modello lineare come caso speciale, dove

$$f_j(x_j) = \beta_j x_j$$

ma è più generale, perché le f_j possono essere funzioni arbitrarie

Abbiamo bisogno di aggiungere a restrizione per renderlo identificabile; senza perdita di generalità

$$\sum_{i=1}^n f_j(x_{ij}) = 0, \quad j = 1, \dots, p$$

Algoritmo di Gauss-Seidel

- Given:
 - $n \times p$ matrix \mathbf{X} of p predictors
 - $n \times 1$ response vector \mathbf{y}
 - small tolerance $\delta > 0$
- Center \mathbf{y} and each column of \mathbf{X}
- Initialize $\hat{\beta}_j \leftarrow 0$ for $j = 1, \dots, p$
- Until (all $|\hat{\beta}_j - \gamma_j| \leq \delta$)
 - for $k = 1, \dots, p$
 - $r_i^{(k)} = y_i - \sum_{j \neq k} \hat{\beta}_j x_{ij}$
 - $\gamma_k \leftarrow$ regression coefficient of $r^{(k)}$ on x_k
 - $\hat{\beta}_j \leftarrow \gamma_k$
- $\hat{\beta}_0 \leftarrow \bar{y} - \sum_{j=1}^p \hat{\beta}_j \bar{x}_j$ with original data
- Return $\hat{\beta}_0, \hat{\beta}_1, \dots, \hat{\beta}_p$

Esercizio 1

Generare i dati come segue:

```
n = 100
set.seed(1)
x1 = rnorm(n)
x2 = rnorm(n)
x3 = rnorm(n)
y = 30 - 10*x1 + 20*x2 + 30*x3 + rnorm(n)
fit = lm(y ~ x1+x2+x3)
X = model.matrix(fit)
```

Ottenere la stima di β con l'algoritmo di Gauss-Seidel

Algoritmo di backfitting

- Given:
 - $n \times p$ matrix of p predictors
 - $n \times 1$ response vector
 - maxit: maximum number of iterations
 - one-dimensional smoother s
- Initialize $\hat{\beta}_0 \leftarrow \bar{y}$ and $\hat{f}_j \leftarrow 0$ for $j = 1, \dots, p$
- for (b in 1:maxit)
 - for $k = 1, \dots, p$
 - $r_i^{(k)} = y_i - \hat{\beta}_0 - \sum_{j \neq k} \hat{f}_j(x_{ij})$
 - $g_k \leftarrow s(r^{(k)} \sim x_k)$
 - $\bar{g}_k \leftarrow g_k - n^{-1} \sum_{i=1}^n g_k(x_{ik})$
 - $\hat{f}_k \leftarrow g_k$
- Return $\hat{\beta}_0, \hat{f}_1, \dots, \hat{f}_p$

Esercizio 2

Generare i dati come segue:

```
set.seed(123)
n <- 500; p <- 4
X <- matrix(runif(n * p, min = -2, max = 2), ncol = p)
f1 <- cos(X[,1] * 4) + sin(X[,1] * 10) + X[,1]^2
f2 <- -1.5 * X[,2]^2 + (X[,2] > 1) * (X[,2]^3 - 1)
f3 <- 0
f4 <- sign(X[,4]) * 1.5
f1 <- f1 - mean(f1); f2 <- f2 - mean(f2)
f3 <- f3 - mean(f3); f4 <- f4 - mean(f4)
y <- 10 + f1 + f2 + f3 + f4 + rnorm(n, sd = 1.2)
```

Ottenere la stima di f_1, \dots, f_4 con l'algoritmo di backfitting