Il compromesso distorsione-varianza

Data Mining
CLAMSES - University of Milano-Bicocca

Aldo Solari

Riferimenti bibliografici

- AS §3.3, §4.1
- HTF §2.3, §2.9, §7.2, §7.3
- AS Exercises 3.1, 3.2, 3.3

Table of Contents

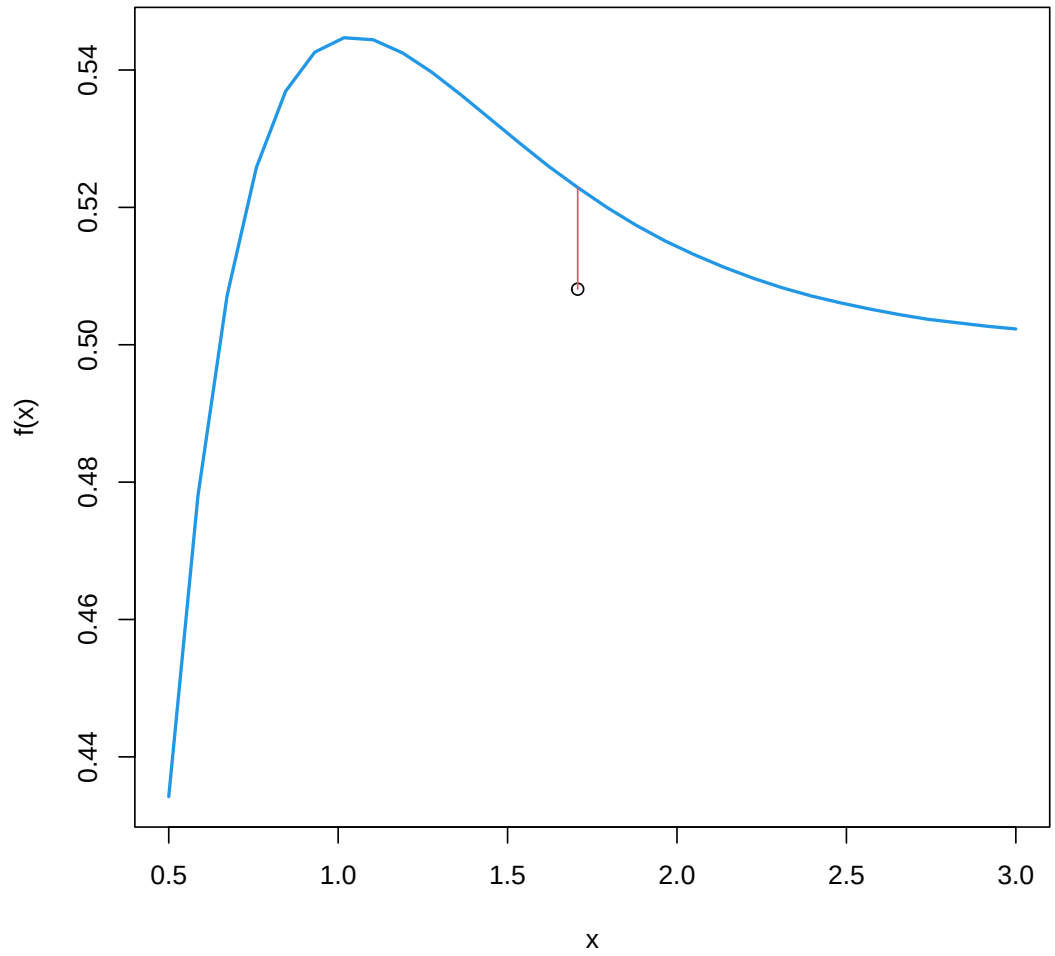
Il segnale e il rumore

Le risposte $y_1, \ldots, y_n$ sono realizzazioni delle v.c.

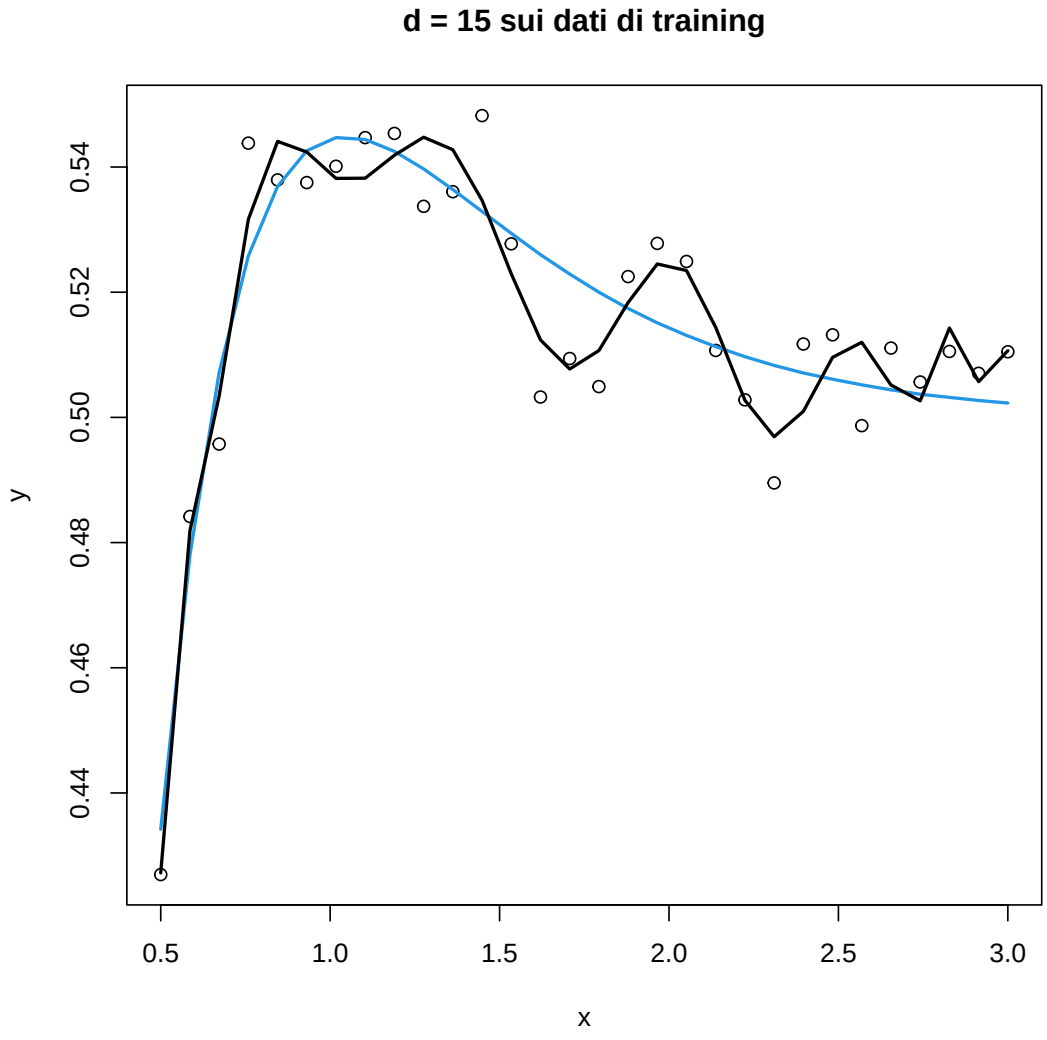
$$Y_i = f(x_i) + \varepsilon_i, \quad i = 1, \ldots, n$$

dove

- f è la funzione di regressione (il segnale)
- ε è il termine di errore (il rumore) dove $\varepsilon_1, \ldots, \varepsilon_n$ sono i.i.d. con $\mathbb{E}(\varepsilon_i) = 0$ e $\mathbb{V}\text{ar}(\varepsilon_i) = \sigma^2$.


f(x)
0.44
0.46
0.48
0.50
0.52
0.54
0.5
1.0
1.5
2.0
2.5
3.0
x

Sovra-adattamento (overfitting)



Confondere il rumore per il segnale

Table of Contents

Errore di previsione

Funzione di perdita: errore quadratico medio (MSE)

Obiettivo: minimizzare l'errore di previsione atteso (nel setting Fixed-X)

$$\text{ErrF} = \mathbb{E}(\text{MSE}_{\text{Te}}) = \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}(Y_i^* - \hat{f}(x_i))^2\right] = \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left[(Y_i^* - \hat{f}(x_i))^2\right]$$

dove il valore atteso è rispetto alle v.c. $Y_1, \ldots, Y_n$ e $Y_1^*, \ldots, Y_n^*$

Fonti di errore

- *Errore irriducibile*
  Possiamo fare previsioni senza commettere errori ?
  No, neppure se conoscessimo la vera f, per la presenza del termine di errore ε
- *Distorsione*
  Quanto è lontano (in media) lo stimatore $\hat{f}$ dalla vera f ?
  Ad esempio, se stimiamo una retta di regressione quando la vera relazione è quadratica
- *Varianza*
  Quanto è variabile lo stimatore $\hat{f}$?
  In altre parole, quanto variano le nostre stime se le calcoliamo su training set diversi?

Errore riducibile ed irriducibile

Supponiamo di prevedere y_i^* con $\hat{y}_i^* = \hat{f}(x_i)$ dove $\hat{f}$ è la stima di f ottenuta con il training set

L'accuratezza di $\hat{y}_i^*$ dipende da due quantità, l'errore riducibile ed l'errore irriducibile

Abbiamo

$$\begin{aligned}
\mathbb{E}[(Y_i^* - \hat{Y}_i^*)^2] &= \mathbb{E}[(f(x_i) + \varepsilon_i^* - \hat{f}(x_i))^2] \\
&= \mathbb{E}[\{f(x_i) - \hat{f}(x_i)\}^2 + \{\varepsilon_i^*\}^2 + 2\{[f(x_i) - \hat{f}(x_i)]\varepsilon_i^*\}] \\
&= \underbrace{\mathbb{E}[\{f(x_i) - \hat{f}(x_i)\}^2]}_{\text{RIDUCIBILE}} + \underbrace{\mathbb{V}\text{ar}(\varepsilon_i^*)}_{\text{IRRIDUCIBILE}}
\end{aligned}$$

dove $\mathbb{V}\text{ar}(\varepsilon_i^*) = \sigma^2$

Errore riducibile

L'errore riducibile può essere ulteriormente scomposto in distorsione (al quadrato) e varianza dello stimatore $\hat{f}$

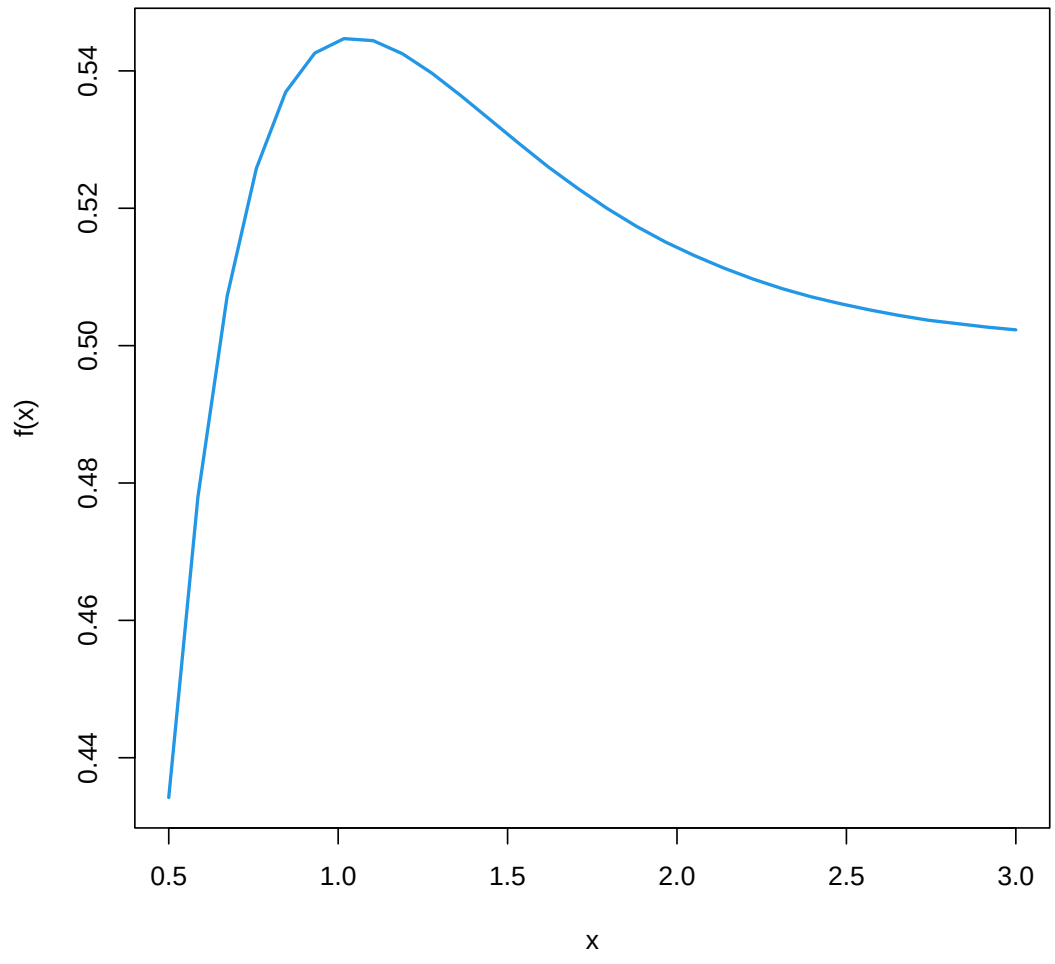
$$\begin{aligned}\mathbb{E}[\{f(x_i) - \hat{f}(x_i))^2] &= \mathbb{E}[(f(x_i) - \mathbb{E}\hat{f}(x_i) + \mathbb{E}\hat{f}(x_i) - \hat{f}(x_i)\}^2] \\ &= \underbrace{[\mathbb{E}\hat{f}(x_i) - f(x_i)]^2}_{[\text{Distorsione}(\hat{f}(x_i))]^2} + \underbrace{\mathbb{V}\text{ar}[\hat{f}(x_i)]}_{\text{Varianza}(\hat{f}(x_i))}\end{aligned}$$

Riassumendo, la scomposizione dell'errore di previsione è data da

$$\text{ErrF} = \sigma^2 + \underbrace{\frac{1}{n}\sum_{i=1}^{n}(\mathbb{E}\hat{f}(x_i) - f(x_i))^2}_{\text{Distorsione}^2} + \underbrace{\frac{1}{n}\sum_{i=1}^{n}\mathbb{V}\text{ar}(\hat{f}(x_i))}_{\text{Varianza}}$$

Distorsione e varianza sono quantità in conflitto e non possiamo minimizzare entrambe contemporaneamente. Dobbiamo quindi scegliere un compromesso (trade-off) tra distorsione e varianza.

Se conoscessimo la vera f...



```
sigmatrue = 0.01
x = seq(.5,3,length=30)
n = length(x)
ftrue = c(0.4342,0.4780,0.5072,0.5258,0.5369,
0.5426,0.5447,0.5444,0.5425,0.5397,
0.5364,0.5329,0.5294,0.5260,0.5229,
0.5200,0.5174,0.5151,0.5131,0.5113,
0.5097,0.5083,0.5071,0.5061,0.5052,
0.5044,0.5037,0.5032,0.5027,0.5023)
```

Distorsione

Per la regressione polinomiale di grado d, la stima del vettore $\underset{p\times 1}{\beta}$ (con $p = d + 1$) risulta pari a $\hat{\beta} = (\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\mathbf{X}^\mathsf{T}\mathbf{y}$ dove $\underset{n\times p}{\mathbf{X}}$ è la matrice del disegno. La previsione per i-sima osservazione è $\hat{f}(x_i) = x_i^\mathsf{T}\hat{\beta}$, dove x_i^T è l' i-sima riga di $\mathbf{X}$.

La distorsione risulta

$$\begin{aligned}\mathbb{E}[\hat{f}(x_i)] - f(x_i) &= \mathbb{E}[x_i^\mathsf{T}(\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\mathbf{X}^\mathsf{T}\mathbf{y}] - f(x_i)\\ &= x_i^\mathsf{T}(\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\mathbf{X}^\mathsf{T}\mathbb{E}[\mathbf{y}] - f(x_i)\\ &= x_i^\mathsf{T}(\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\mathbf{X}^\mathsf{T}\mathbf{f} - f(x_i)\end{aligned}$$

dove $\mathbf{f} = (f(x_1), \ldots, f(x_n))^\mathsf{T}$.

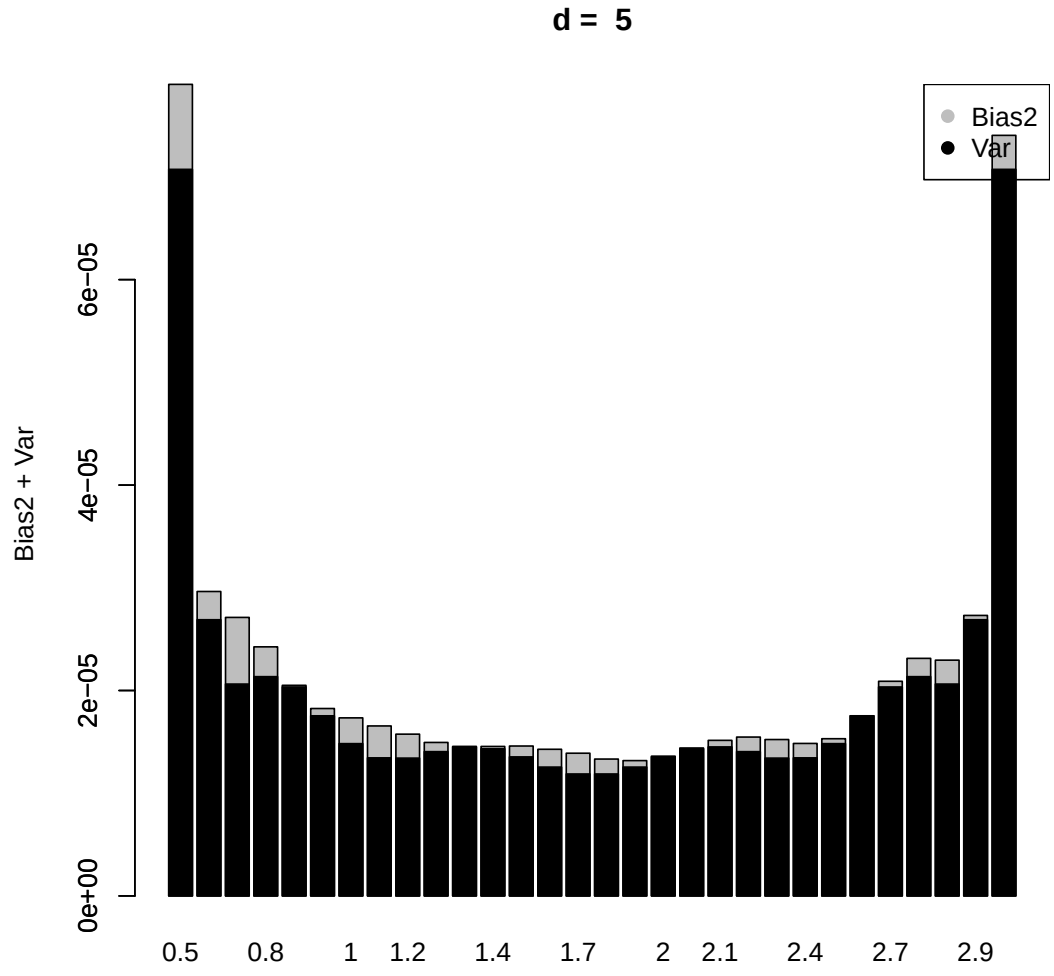
Varianza

Abbiamo inoltre $\mathbb{V}\text{ar}(\hat{\beta}) = \sigma^2(\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}$, quindi la varianza della previsione $\hat{f}(x_i)$ risulta

$$\mathbb{V}\text{ar}(\hat{f}(x_i)) = x_i^\mathsf{T}\mathbb{V}\text{ar}(\hat{\beta})x_i$$

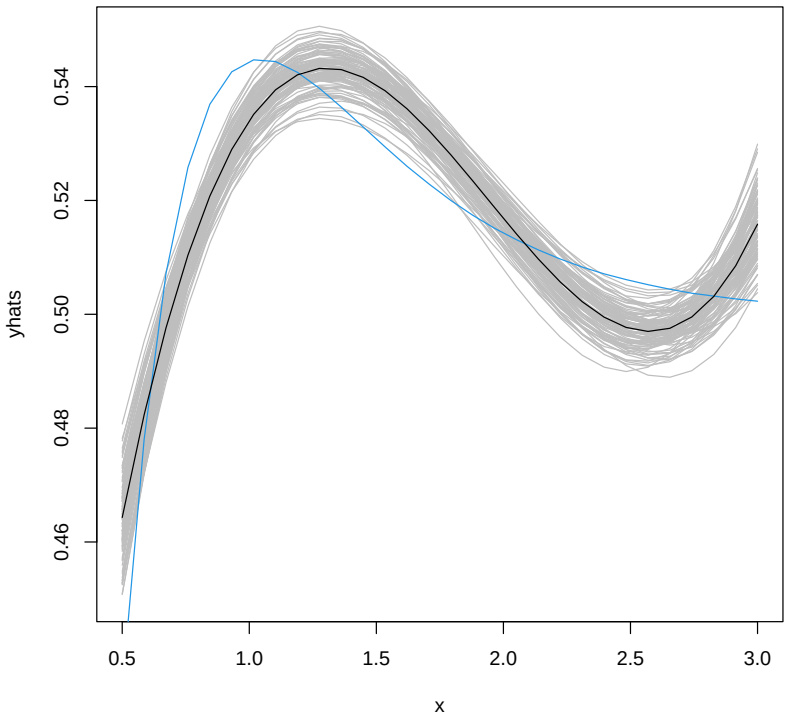
e sommando rispetto alle n osservazioni

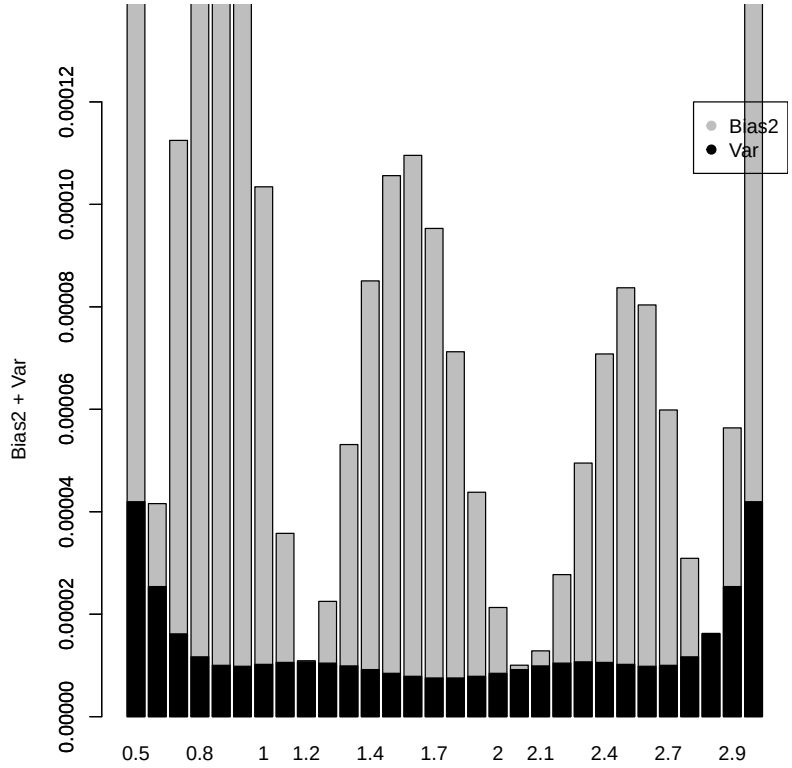
$$\sum_{i=1}^{n}\mathbb{V}\text{ar}(\hat{f}(x_i)) = \text{tr}(\mathbf{X}\mathbb{V}\text{ar}(\hat{\beta})\mathbf{X}^\mathsf{T}) = \sigma^2\text{tr}(\mathbf{X}^\mathsf{T}\mathbf{X}(\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}) = \sigma^2 p$$


d = 5
Bias2
Var
Bias2 + Var
0e+00
2e−05
4e−05
6e−05
0.5
0.8
1
1.2
1.4
1.7
2
2.1
2.4
2.7
2.9

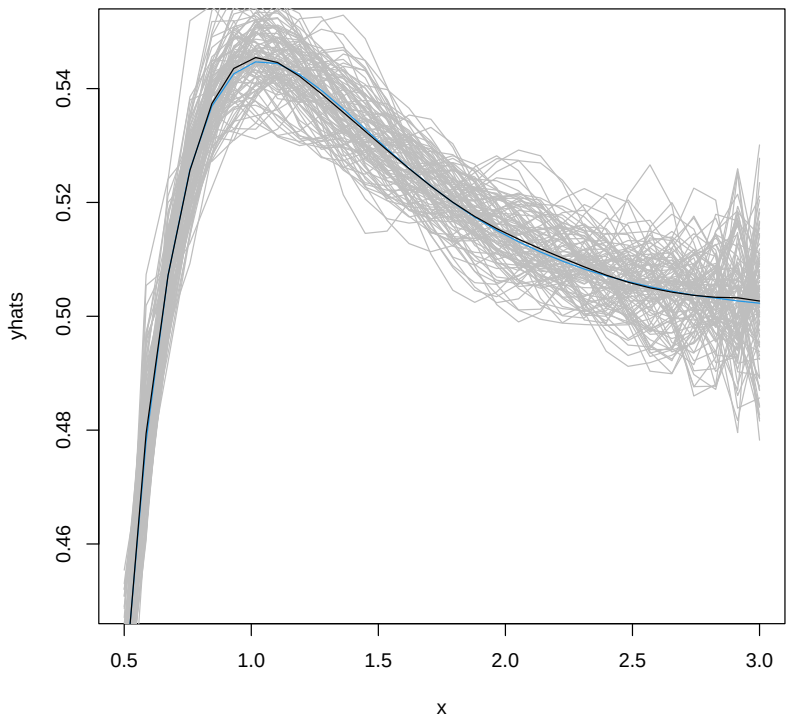
```
# errore di previsione atteso
sigmatrue^2 + mean( Bias2 ) + mean( Var )
0.0001217199

# verifichiamo via simulazione l'errore di previsione attes
ErrF = function(d){
  y = ftrue + rnorm(n,0,sigmatrue)
  fit = lm(y ~ poly(x,degree=d))
  yhat = fitted(fit)
  y_new = ftrue + rnorm(n,0,sigmatrue)
  MSE.te = mean( (yhat - y_new)^2 )
}
B = 1000
set.seed(123)
mean(replicate(B, ErrF(d=5)))
0.0001221817
```

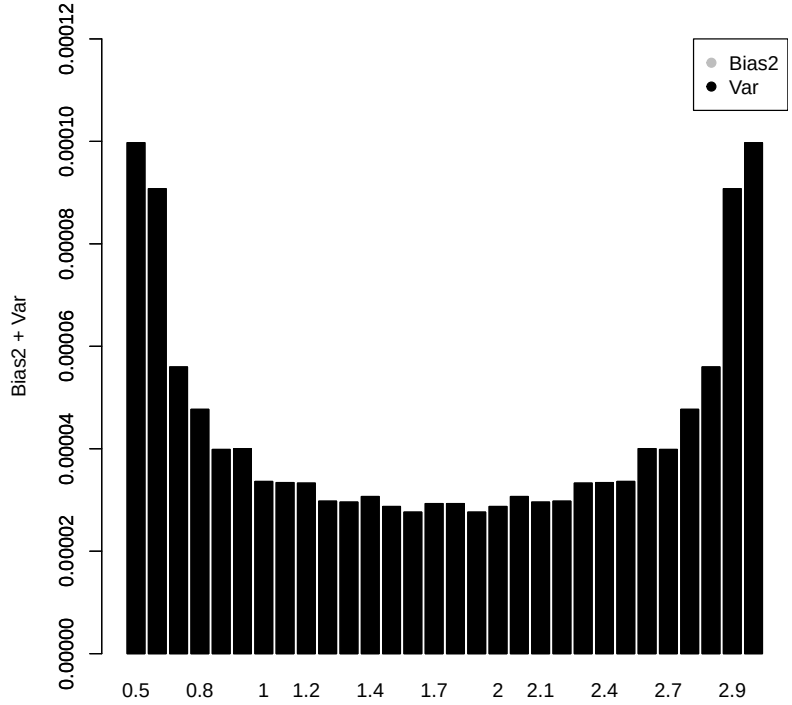
Regressione polinomiale di grado 3



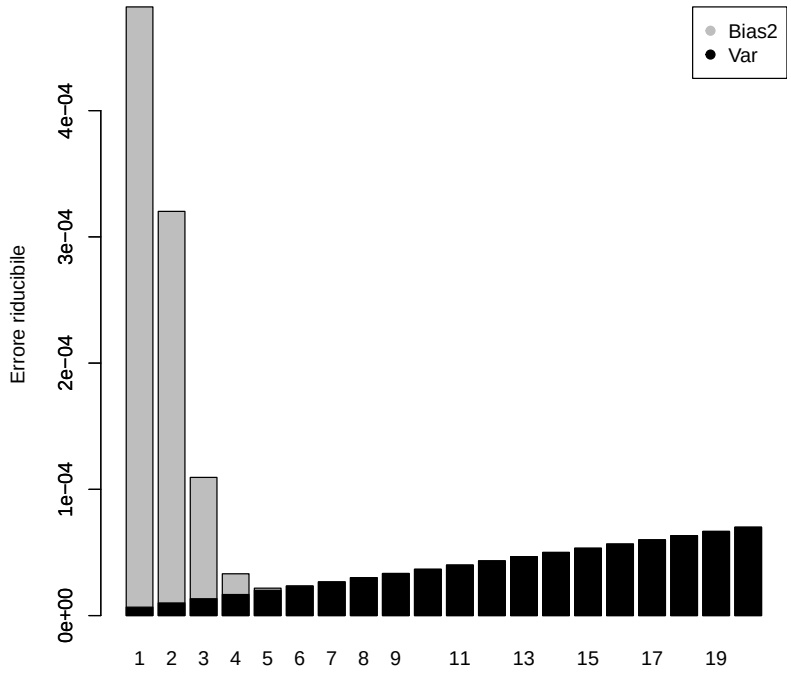


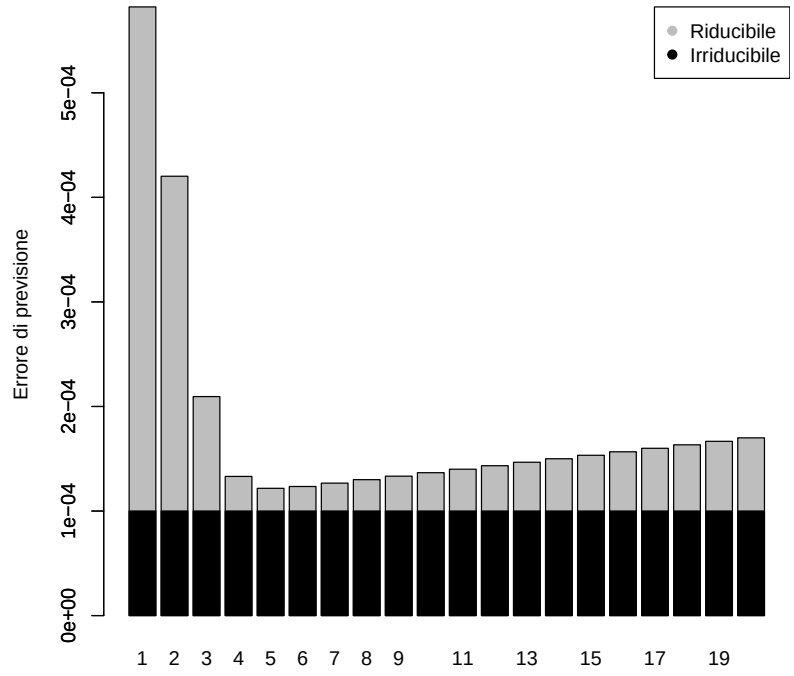
Regressione polinomiale di grado 12



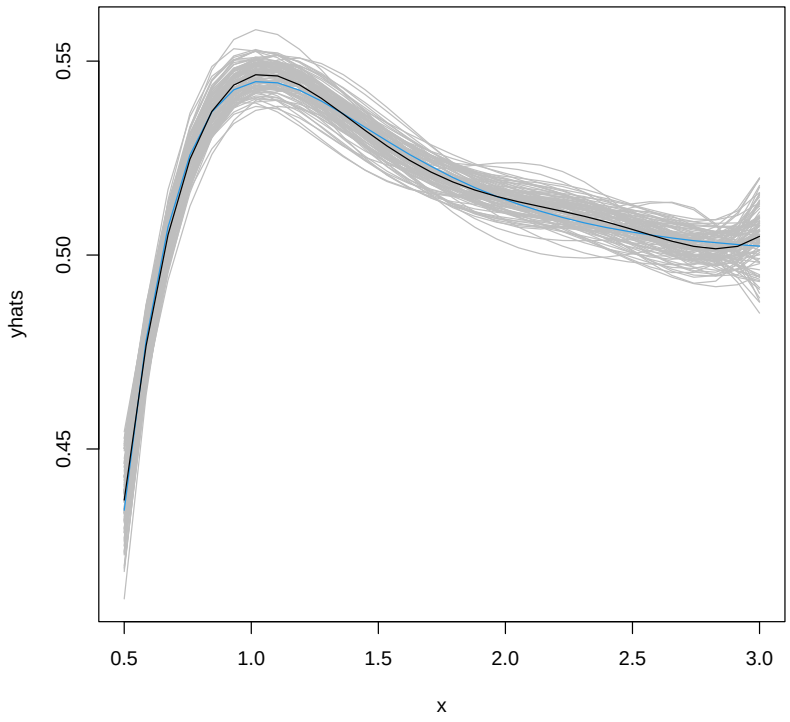


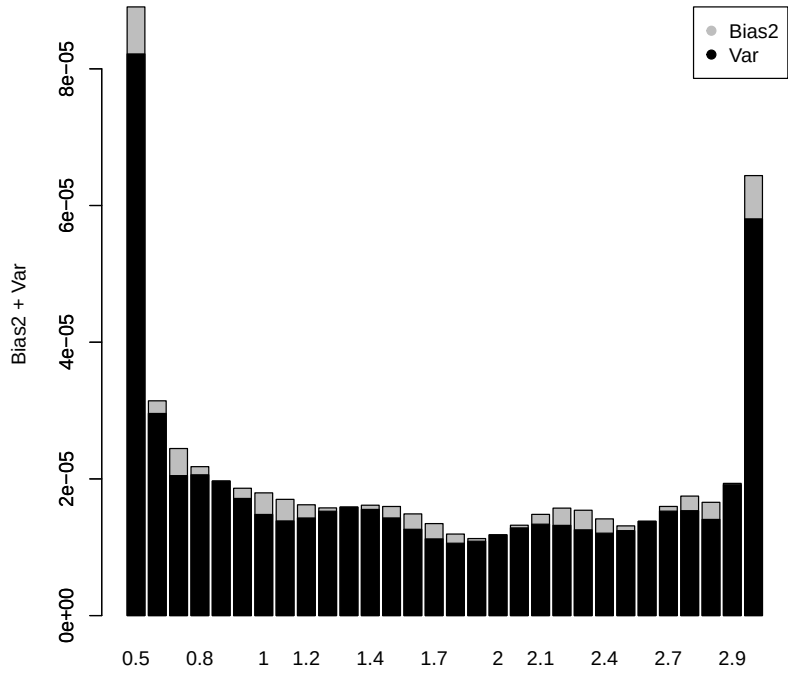
Errore riducibile e atteso per $d = 1, \ldots, 20$





Il miglior modello





Il compromesso distorsione-varianza

- Errore riducibile = Distorsione ^{2} + Varianza
- Modelli con poca distorsione tendono ad avere elevatà variabilità
- Modelli con poca variabilità tendono ad avere elevatà distorsione
- Da un lato, anche se il nostro modello è non distorto ($\mathbb{E}\hat{f}(x_i) = f(x_i)$), l'errore di previsione può essere elevato se il modello è molto variabile
- Dall'altro lato, un modello che prevede una costante (e.g. $\hat{f}(x_i) = 0$) ha varianza nulla ma elevata distorsione
- Per prevedere bene, dobbiamo bilanciare la distorsione e la varianza

Table of Contents

Metodi nonparametrici

I metodi nonparametrici non fanno assunzioni specifiche a proposito della forma funzionale della f (ad esempio, " f è un polinomio "). Si lascia che i dati "parlino da soli"

Vantaggio: evitando assunzioni sulla forma di f, permettiamo qualsiasi forma funzionale per f (anche le più strane e irregolari)

Svantaggio: poichè il problema della stima di f non si riduce più al problema di stimare p parametri (se $p \ll n$), risulta necessario avere tante osservazioni (n elevato)

Il metodo dei k vicini più vicini

Si consideri il metodo dei k vicini più vicini (k-nearest neighbors, abbreviato k-NN) per un problema di regressione

Supponiamo di voler prevedere la risposta y_1^* in corrispondenza ad un certo punto x_1^*. Definiamo il "vicinato" (neighbourhood) di questo punto con $N_k(x_1^*)$: è l'insieme dei k punti del training set più "vicini" a x_1^*. Dobbiamo quindi definire una distanza: si può ad esempio considerare la distanza Euclidea tra x_i e x_1^*

$$\|x_i - x_1^*\|_2 = \sqrt{(x_i - x_1^*)^\mathsf{T}(x_i - x_1^*)} = \sqrt{\sum_{j=1}^{p}(x_{ij} - x_{1j}^*)}$$

dove $\|\cdot\|_2$ indica la norma Euclidea

La previsione è definita dalla media delle k risposte y_i del training set che appartengono al vicinato di x_1^*, i.e. le $x_i \in N_k(x_1^*)$

$$\hat{f}(x_1^*) = \frac{1}{k}\sum_{i \in N_k(x_1^*)} y_i$$

Parametro di regolazione

k può assumere valori da 1 a n, e rappresenta il parametro di regolazione (tuning parameter)

Un k piccolo corrisponde ad una stima più flessibile, vicerversa un k grande ad una stima meno flessibile

Il caso estremo $k = n$ corrisponde alla media delle risposte del training: $\hat{f}(x_1^*) = \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$, mentre $k = 1$ corrisponde a $\hat{f}(x_1^*) = y_i$ per un certo indice i tale che x_i è il punto più vicino a x_1^*

Operativamente, poichè il metodo utilizza una distanza, spesso i valori dei predittori vengono standardizzati

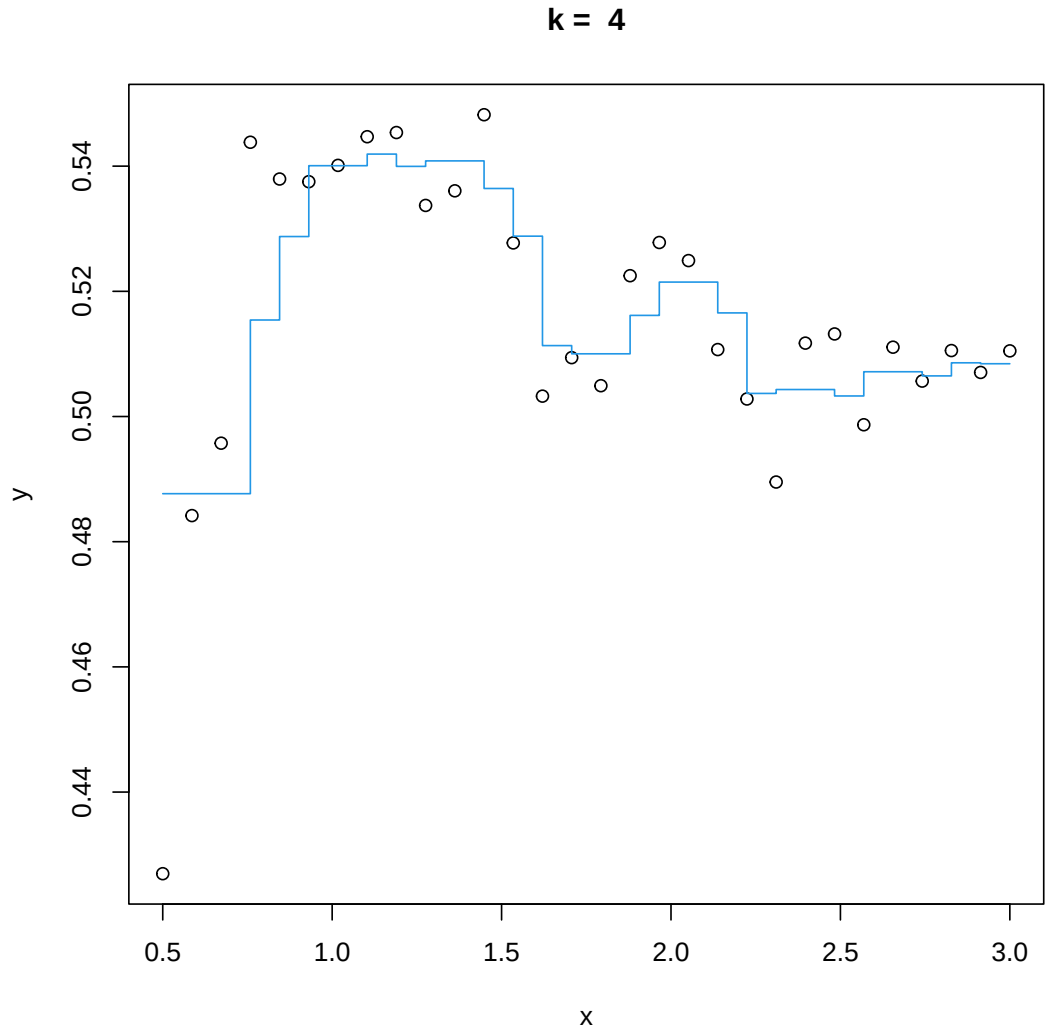
Parametro di regolazione

k può assumere valori da 1 a n, e rappresenta il parametro di regolazione (tuning parameter)

Un k piccolo corrisponde ad una stima più flessibile, vicerversa un k grande ad una stima meno flessibile

Il caso estremo $k = n$ corrisponde alla media delle risposte del training: $\hat{f}(x_1^*) = \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$, mentre $k = 1$ corrisponde a $\hat{f}(x_1^*) = y_i$ per un certo indice i tale che x_i è il punto più vicino a x_1^*

Operativamente, poichè il metodo utilizza una distanza, spesso i valori dei predittori vengono standardizzati


k = 4
y
0.44
0.46
0.48
0.50
0.52
0.54
0.5
1.0
1.5
2.0
2.5
3.0
x

Errore di previsione atteso per kNN

Nel Fixed-X setting, la distorsione di kNN risulta

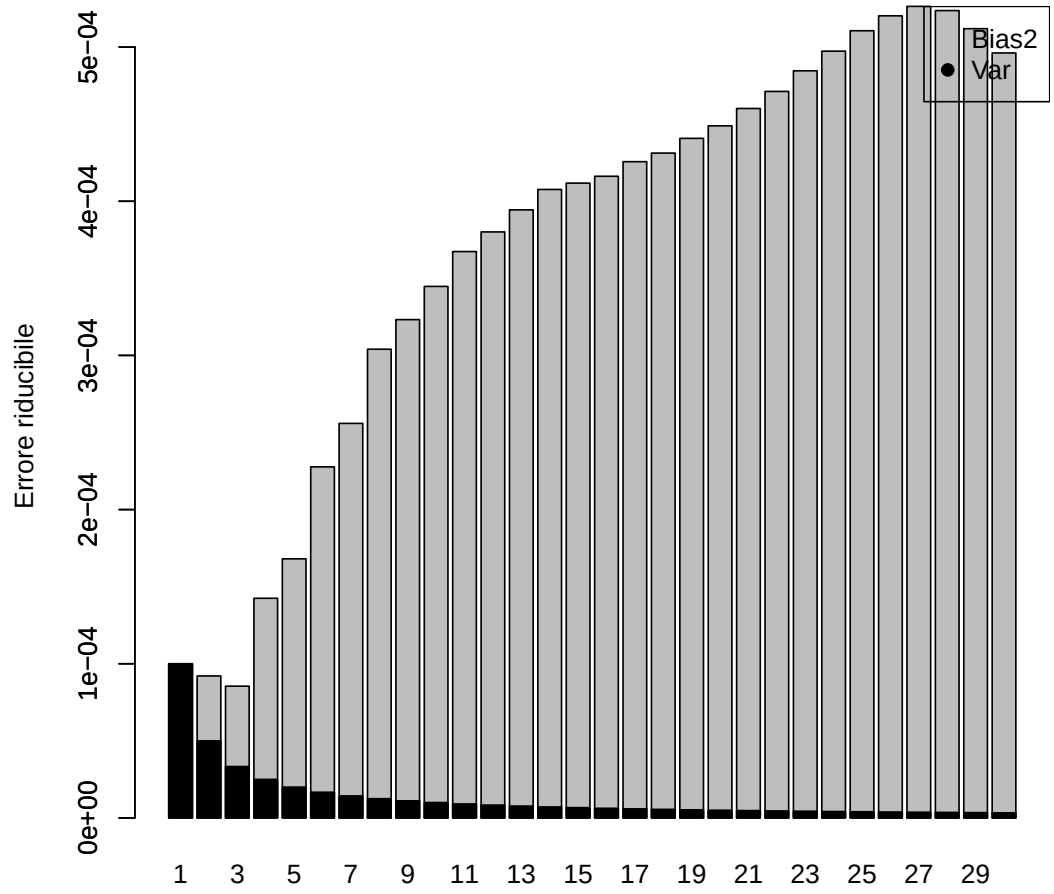
$$\mathbb{E}[\hat{f}(x_i)] - f(x_i) = \frac{1}{k} \sum_{i \in N_k(x_i)} \mathbb{E}[y_i] - f(x_i) = \frac{1}{k} \sum_{i \in N_k(x_i)} f(x_i) - f(x_i)$$

e la varianza di kNN risulta

$$\mathbb{V}\text{ar}(\hat{f}(x_i)) = \mathbb{E}\{(\hat{f}(x_i) - \mathbb{E}[\hat{f}(x_i)])^2\} = \frac{\sigma^2}{k}$$

quindi l'errore di previsione di kNN si può calcolare come

$$\text{ErrF} = \sigma^2 + \frac{1}{n} \sum_{i=1}^{n} (\frac{1}{k} \sum_{i \in N_k(x_i)} f(x_i) - f(x_i))^2 + \frac{\sigma^2}{k}$$


Errore riducibile
0e+00
1e−04
2e−04
3e−04
4e−04
5e−04
1
3
5
7
9
11
13
15
17
19
21
23
25
27
29
Bias2
Var

```
# errore di previsione atteso
sigmatrue^2 + Bias2s[4] +Vars[4]
0.0002424745

# verifichiamo via simulazione l'errore di previsione attes
ErrF = function(k){
  y = ftrue + rnorm(n,0,sigmatrue)
  yhat =  kknn(y ~ x, train, test, distance = 2,
  kernel = "rectangular", k = k)$fitted.values
  y_new = ftrue + rnorm(n,0,sigmatrue)
  MSE.te = mean( (yhat - y_new)^2 )
}
B = 1000
mean(replicate(B, ErrF(k=4)))
0.0002462827
```