

# Multivariate and Statistical Learning: Metodo della massima verosimiglianza in R

Università degli studi di Firenze

September 30, 2019

# Overview

- 1 Metodo della massima verosimiglianza per una distribuzione di Bernulli
- 2 Metodo della massima verosimiglianza per una distribuzione di Bernulli in R parte 1
- 3 Metodo della massima verosimiglianza per una distribuzione di Bernulli in R parte 2
- 4 Metodo della massima verosimiglianza per una distribuzione di una normale in R
- 5 Metodo della massima verosimiglianza per la regressione logistica
- 6 Metodo della massima verosimiglianza per la regressione logistica in R parte 1
- 7 Metodo della massima verosimiglianza per la regressione logistica in R parte 2

# Metodo della massima verosimiglianza per una distribuzione di Bernulli

- Consideriamo un campione casuale semplice dimensione  $n$  da una variabile dicotomica di  $X \sim B(p)$
- Vogliamo usare un metodo di calcolo per ottenere la stima dei parametri in un modello parametrico dai dati osservati
- Per  $X$  è definita la seguente funzione di massima verosomiglianza e di log verosomiglianza.

$$L(p; x_1, \dots, x_n) = \prod_{i=1}^n p^{x_i} (1 - p)^{1-x_i}$$

- Per semplificare i compiti conviene effettuare una trasformazione logaritmica di  $L$  detta log-verosomiglianza, che è monotona crescente e non cambia il punto di max.

$$l(p; x_1, \dots, x_n) = \sum_{i=1}^n x_i \log(p) + \log(1 - p)(n - \sum_{i=1}^n x_i)$$

# Metodo della massima verosimiglianza per una distribuzione di Bernulli in R

- In R le funzioni di L e l sono state implementate in questo modo

```
likelihood <- function(data, p){  
  sum_xi = 0  
  for (i in 1:length(data)) {  
    sum_xi = sum_xi+data[i]  
  }  
  return ((p^(sum_xi))*(1-p)^(length(data)-sum_xi))  
}  
  
logLikelihood <- function(data, p){  
  sum_xi = 0  
  for (i in 1:length(data)) {  
    sum_xi = sum_xi+data[i]  
  }  
  return (sum_xi*log(p)+log(1-p)*(length(data)-sum_xi))  
}
```

# Metodo della massima verosimiglianza per una distribuzione di Bernulli in R

- Per trovare  $p$  che ha generato i dati, dobbiamo far variare il valore  $p$  in  $[0,1]$  e passarlo allo log-verosomiglianza e cercare poi il massimo della funzione

```
set.seed(42)
sequence <- rbinom(n = 100, size = 1, prob = 0.2)
p <- seq(0, 1, by = 0.001)

likelihoodVal = c()
for (i in 1:length(p)) {
  likelihoodVal[i] = likelihood(sequence,p[i])
}

loglikelihoodVal = c()
for (i in 1:length(p)) {
  loglikelihoodVal[i] = logLikelihood(sequence,p[i])
}

imax <- which.max(loglikelihoodVal) #cerca
p_MLE <- p[imax] # recupera il valore di p
```

# Metodo della massima verosimiglianza per una distribuzione di una normale

- Consideriamo un campione casuale semplice di dimensione  $n$  da una variabile normale di  $X \sim N(\mu, \sigma)$  dove  $\sigma$  è nota e quindi vogliamo stimare la media  $\mu$
- Le funzioni di verosomiglianza e log-verosomiglianza per  $X$  sono definite rispettivamente come:

$$L(\mu, \sigma; x_1, \dots, x_n) = (2\pi\sigma^2)^{\frac{-n}{2}} e^{\frac{-1}{2\sigma^2} \sum_{i=1}^n (x_i - \mu)^2}$$

$$l(\mu, \sigma; x_1, \dots, x_n) = \frac{-n}{2} \log(2\pi\sigma^2) - \frac{1}{2\sigma^2} \sum_{i=1}^n (x_i - \mu)^2$$

# Metodo della massima verosimiglianza per una distribuzione di una normale in R

- Per ricavare  $\mu$  si eseguono gli stessi passi, si fa variare  $\mu$  e la si passa alla log verosomiglianza insieme ai dati e a  $\sigma$

```
likelihood <- function(data, sigma, mu){  
  dev =sum((data-mu)^2)  
  return(((2*pi*sigma^2)^(-length(data)*0.5))*pi^(-0.5*dev*1/sigma^2))  
}  
  
log_likelihood <- function(data, sigma, mu){  
  dev =sum((data-mu)^2)  
  return(-length(data)*0.5*log(2*pi*sigma^2)-0.5*dev*1/sigma^2)  
}
```

# Metodo della massima verosimiglianza per la regressione logistica

Per trovare il vettore dei coefficienti di regressione si usa il metodo di Newton, i passi sono i seguenti:

- 1 Calcola la score function come  $s(\hat{\beta}_{ML}) = \sum_{i=1}^n x_i(y_i - \frac{e^{x_i^T \beta}}{1+e^{x_i^T \beta}})$
- 2 Calcola la matrice dell'informazione osservata  $H$  e la sua inversa
- 3 Calcola la soluzione come  $\hat{\beta}_{t+1} = \hat{\beta}_t + H_t^{-1} s_t(\hat{\beta}_{ML})$ .
- 4 Calcola la log-verosimiglianza usando  $\hat{\beta}_{t+1}$ .
- 5 Ripeti i passi finchè  $t < it_{max} \wedge |\hat{\beta}_t - \hat{\beta}_{t+1}| > \epsilon$

N.B:

- $it_{max}$  e  $\epsilon$  sono rispettivamente il numero di iterazioni massime e la tolleranza  $\epsilon = 0.0004$
- A livello computazionale l'inversa di una matrice non andrebbe mai fatta perchè è un'operazione costosa. Quindi ad ogni passo invece che fare l'inversa si potrebbe utilizzare la fattorizzazione LU pivoting per fattorizzare  $H$  e risolvere quindi una successione di sistemi lineari.



# Metodo della massima verosimiglianza per la regressione logistica in R

```
sigmoid <- function(x){  
  1/(1+exp(-x))  
}  
  
log_likelihood <- function(beta, X, y){  
  return((-1/m)*sum(y*log(sigmoid(X%%beta)) + (1-y)*log(1-sigmoid(X%%beta))))  
}
```

# Metodo della massima verosimiglianza per la regressione logistica in R

```
newton <- function(X, y, beta, num_iter, eps){  
  library(numDeriv)  
  library(MASS)  
  J_hist<-vector()  
  
  for(i in 1:num_iter){  
    score <- (1/n)*(t(X)%*(sigmoid(X%*%beta) - y)) # calcolo della score function  
    H <- hessian(log_likelihood, beta, method = "complex", X = X, y = y)  
    #H contiene le derivate seconde della log-verosimiglianza  
    beta_old = beta  
    beta <- beta_old - ginv(H)%*%score # passo iterativo del metodo di newton  
    J_hist[i] <- log_likelihood(beta, X, y) # calcola la log_likelihood sui dati con il  
    if(norm_vec(beta_old-beta)>eps & i<num_iter){  
      break  
    }  
  }  
  result <- list(beta, J_hist)  
  return(result)  
}
```