Università degli studi di Firenze

Curriculum Data Science

Inferenza statistica Bayesiana

Quaderno degli esercizi

Studente
Filippo Mameli
filippo.mameli@stud.unifi.it

Anno accademico 2017-2018

1 Prima Parte

2 Seconda Parte

Esercizio 10.2 Hoff

Nesting success: younger male sparrows may or may not nest during a mating season, perhaps depending on their physical characteristics. Researchers have recorded the nesting success of 43 young male sparrows of the same age, as well as their wingspan, and the data appear in the file `msparrownest.dat`. Let Y_i be the binary indicator that sparrow i successfully nests, and let x_i denote their wingspan. Our model for Y_i is $logit\theta(Y_i = 1|\alpha, \beta, x_i)) = \alpha + \beta x_i$, where the logit function is given by $logit\theta = log\left[\frac{\theta}{1-\theta}\right]$.

1. Write out the joint sampling distribution $\prod_{i=1}^{n} p(y_i|\alpha, \beta, x_i)$ and simplify as much as possible.

2. Formulate a prior probability distribution over α and β by considering the range of $Pr(Y = 1|\alpha, \beta, x)$ as x ranges over 10 to 15, the approximate range of the observed wingspans.

3. Implement a Metropolis algorithm that approximates $p(\alpha, \beta|\mathbf{y}, \mathbf{x})$. Adjust the proposal distribution to achieve a reasonable acceptance rate, and run the algorithm long enough so that the effective sample size is at least $1,000$ for each parameter.

4. Compare the posterior densities of α and β to their prior densities.

5. Using output from the Metropolis algorithm, come up with a way to make a confidence band for the following function $f_{\alpha\beta}(x)$ of wingspan:

$$f_{\alpha\beta}(x) = \frac{e^{\alpha+\beta x}}{1 + e^{\alpha+\beta x}}$$

   where α and β are the parameters in your sampling model. Make a plot of such a band.

Svolgimento:

L'esercizio ha come obiettivo principale quello di studiare la relazione tra laprobabilità di nidificare e l'ampiezza delle ali per un gruppo di 43 uccellini maschi della stessa età. Il setting del modello è il seguente:

$$Y_i = \begin{cases} 1, & \text{se l'uccellino } i \text{ nidifica} \\ 0, & \text{altrimenti} \end{cases} ; \quad x_i = \text{ampiezza dell'uccellino } i; \quad i = 1, \dots, 43$$

La verosomiglianza per ogni singola osservazione è pertanto una Bernoulli:

$$p(y_i|p_i) = p_i^{y_i}(1-p_i)^{1-y_i}$$

Studiamo la relazione tra la probabilità di nidificare e l'ampiezza delle ali con il modello logistico (siamo quindi nell'ambito dei modelli lineari generalizzati):

$$g(p_i) = logit(p_i) = log\left(\frac{p_i}{1-p_i}\right) = \eta_i = \alpha + \beta x_i$$

pertanto

$$g^{-1}(\eta_i) = p_i = \frac{e^{\eta_i}}{1+e^{\eta_i}} = \frac{e^{\alpha+\beta x_i}}{1+e^{\alpha+\beta x_i}}$$

Parte a

Scriviamo la verosomiglianza secondo il modello appena descritto, quindi come funzione di α e β (ricordiamo l'indipendenza condizionata a tali parametri):

$$\begin{aligned}\mathcal{L}(\alpha;\beta;\mathbf{y};\mathbf{X}) = p(\mathbf{y}|\alpha,\beta,\mathbf{X}) = \prod_{i=1}^{n} p(y_i|\alpha,\beta,\mathbf{x_i}) = \prod_{i=1}^{n}\left[\left(\frac{e^{\eta_i}}{1+e^{\eta_i}}\right)^{y_i}\left(\frac{1}{1+e^{\eta_i}}\right)^{1-y_i}\right] \\ = \prod_{i=1}^{n}\frac{e^{y_i\eta_i}}{1+e^{\eta_i}} = \prod_{i=1}^{n}(e^{y_i\eta_i} - log(1-e^{\eta_i})) = e^{\sum_{i=1}^{n}[y_i\eta_i - log(1+e^{\eta_i})]} \\ = e^{\sum_{i=1}^{n}[y_i(\alpha+\beta x_i) - log(1+e^{\alpha+\beta x_i})]}\end{aligned}$$

Potevamo in maniera analoga procedere passando direttamente attraverso la scrittura delle singole verosomiglianze nella forma della famiglia esponenziale in questo modo:

$$p(y_i|p_i) = p_i^{y_i}(1-p_i)^{1-y_i} = e^{y_i log(\frac{p_i}{1-p_i} + log(1-p_i))}$$

pertanto

$$\prod_{i=1}^{n} p(y_i|p_i) = \prod_{i=1}^{n} e^{\sum_{i=1}^{n}[y_i\eta_i + log(\frac{1}{1+\eta_i})]} = e^{\sum_{i=1}^{n}[y_i\eta_i + log(1+\eta_i)]}$$
$$e^{\sum_{i=1}^{n}[y_i(\alpha+\beta x_i) - log(1+e^{\alpha+\beta x_i})]}$$

Parte b

Possiamo formulare la a priori per α e β in molti modi, due dei quali sono i seguenti:

1. **soggettivamente:** a priori pensiamo che la probabilità di nidificare sia alta e che vari tra $[0.5, 0.9]$; sapendo inoltre che il campo di variazione della covariata è $[10, 15]$, troviamo il range di α e β che sia compatibile conq uello della probabilità e in base ad esso formuliamo la prior sui parametri. In dettaglio: Pensiaom che $p = Pr(Y = 1|\alpha, \beta, \mathbf{x}) \in [0.5, 0.9]$ e quindi che $logit(p) = log\left(\frac{p}{1-p}\right) = \alpha + \beta x \in [0, 2.2]$. Si ha il seguente sistema di disequazioni:

$$\begin{cases} \alpha + \beta x \geq 0 \\ \alpha + \beta x \leq 2.2 \end{cases}$$

Troviamo il range di α e di β risolvendo il sistema per il valore minimo e per quello massimo di x:

$$\begin{cases} \alpha + 10\beta = 0 \\ \alpha + 15\beta = 2.2 \end{cases} \quad \begin{cases} \beta = 0.44 \\ \alpha = -4.4 \end{cases} \quad \begin{cases} \alpha + 15\beta = 0 \\ \alpha + 10\beta = 2.2 \end{cases} \quad \begin{cases} \beta = -0.44 \\ \alpha = -4.4 \end{cases}$$

Quindi $\beta \in [-0.44, 0.44]$ e $\alpha \in [-4.46.6]$. Ipotizzando come prior per α e β una normale (soluzione più naturale dal momento che in ogni caso non è possibile fare inferenza n+ in forma chiusa nè tramite Gibbs sampler ma con un algoritmo Metropolis-Hastings), specifichiamo come vettore delle medie il centroide $(\alpha_0, \beta_o)^T = (1.1, 0)^T$. Resta da specificare la matrice di varianza e covarianza. Innanzitutto, dal momento che i valori di α e β che sono contemporaneamente massimi e contemporaneamente minimi generano valori del logit fuori dal range a priori, ipotizziamo covarianza nulla tra i due parametri in modo che i valori appena citati siano meno probabili: $\sigma_{\alpha\beta} = 0$. Per specificare le varianze σ^2_α e σ^2_β seguiamo la logica degli intervalli di confidenza: date le distribuzioni normali di α e β, sappiamo che:

$$P(\alpha_0 - 2\sigma_\alpha \leq x \leq \alpha_0 + 2\sigma_\alpha) \simeq 0.95; \qquad P(\beta_0 - 2\sigma_\beta \leq x \leq \beta_0 + 2\sigma_\beta) \simeq 0.95$$

Quindi cerchiamo le deviazioni standard in modo che

$$2\sigma_\alpha = \frac{6.6 - (-4.4)}{2} = 5.5; \qquad 2\sigma_\beta = 0.44$$

e si ha che

$$\sigma_\alpha 2.75; \qquad \sigma_b eta = 0.22$$

Per tutto quanto detto, la prior formulata considerando il range di $P(Y = 1|\alpha, \beta, x)$ al variare di x in $[10, 15]$ è:

$$\begin{pmatrix} \alpha \\ \beta \end{pmatrix} \sim N_2 \left(\begin{pmatrix} \alpha_0 \\ \beta_0 \end{pmatrix}, \begin{pmatrix} \sigma^2_\alpha & \sigma^2_{\alpha\beta} \\ \sigma^2_{\alpha\beta} & \sigma^2_\beta \end{pmatrix} \right) \equiv N_2 \left(\begin{pmatrix} 1.1 \\ 0 \end{pmatrix} \begin{pmatrix} 2.75^2 & 0 \\ 0 & 0.22^2 \end{pmatrix} \right)$$

2. **In maniera non informativa:** ricaviamo il range di p in base alla proproporzione osservata e all'appprossimazione alla distribuzione normale della porporzione campionaria. In questo caso $\hat{p} = 0.55$; poiché $\hat{p} \approx N(p, \frac{p(1-p)}{n})$, ragionando sempre secondo la logica degli intervalli di confidenza, ipotizziamo che il campo di variazione di p sia tra $\hat{p} - 2\sqrt{\frac{\hat{p}(1-\hat{p})}{n}} = 0.55 - 2\sqrt{\frac{0.55 \cdot 0.45}{43}} \approx 0.47$ e $\hat{p} + 2\sqrt{\frac{\hat{p}(1-\hat{p})}{n}} = 0.55 + 2\sqrt{\frac{0.55 \cdot 0.45}{43}} \approx 0.62$. Ponendo quindi $p \in [0.47, 0.62]$, impostiamo la prior per α e β secondo la logica seguita al punto precedente. Non svogliamo tutti i calcoli perché si è scelto di lavorare con la prior individuata al punto 1.

Rispondiamo adesso alle altre richieste dell'esercizio in R. Di seguito il codice con output e commenti.

```
#Funzione per campionare da una distribuzione normale multivariata
rmvnorm <- function(n, mu, Sigma)
{
   #samples form the multivariate normal distribution
   E<-matrix(rnorm(n*length(mu)), n, length(mu))
   t( t(E%*%chol(Sigma)) +c(mu))
}
#Lettura dei dati :
```

```r
dati<-as.matrix(dati<-read.table(
  "C:\\Documents\\Bene\\Desktop\\msparrownest.dat",
  col.names=c("Y" , "X"+)))
head(dati)
```

```
Y  X
[1,]  0  13.03
[2,]  1  13.69
[3,]  1  12.62
[4,]  0  11.70
[5,] 0  12.39
[6,]  0  12.44
```

```r
#Matrice del modello e vettore delle osservazioni :
X = cbind(1, dati[,2])
head(X)
```

```
[,1] [,2]
[1,]  1 13.03
[2,]  1 13.69
[3,]  1 12.62
[4,]  1 11.70
[5,]  1 12.39
[6,]  1 12.44
```

```r
y = dati[,1]
head(y)
```

```
[1] 0 1 1 0 0 0
```

```r
#Numero di osservazioni e numero dei parametri si cui fare inferenza
  :
n<-length(y)
p<-dim(X)[2]
#b)
#Si veda il setting per la formulazione della prior:
pmn.beta<-c(1.1, 0)
psd.beta<-c(2.75, 0.22)
#c)
#Creiamo una funzione che approssima la distribuzione a posteriori
  dei
```

```
#coefficienti di regressione mediante algoritmo di Metropolis-Hastings . E'
#richiesto di aggiustare la distribuzione proposta in modo che il tasso di
#accettazione sia ragionevole e di considerare un numero di iterazioni che
#portano ad una effective sample size di circa 1000: pertanto la funzione
#creata prende in ingresso questi due parametri
#Osservazione :
#abbiamo appena detto che la matrice di varianza e covarianza della
#distribuzione proposal e ' inserita come input della funzione in modo da
#vedere come cambiano i risultati al variare di questa e poterla cosi '
#scegliere in maniera adeguata a rispondere alle rischieste . Ricordiamo
#pero ' che essa e ' fissa per ogni catena , ovvero ogni catena ha la sua . In
#verita ' in alcuni casi si puo ' anche aggiustare durante l ' algoritmo a
#patto che siano soddisfatte certe condizioni : in teoria si puo' quindi
#estrarre la varianza ogni volta ma a patto che la distribuzione da cui
#viene estratta non dipenda dai valori dei parametri estratti nella catena
#(a parte quelli dell 'iterazione precedente).
#Questo succede in situazioni piu' complicate in cui a volte vorremmo poter > #fare piccoli passi e a volte grandi : quindi e ' possibile fare un certo
#numero di iterazioni con varianza piccola e un altro con varianza grande , > #sempre sotto la condizione che queste due varianze siano prespecificate o > #comunque prese random e non dipendere dai valori estratti durante la >#catena). In questo modo la distribuzione proposal e' piu' flessibile e
#riesce ad esplorare piu ' facilmente la distribuzione a posteriori . Come >#esempio concreto si puo' considerare il caso in cui la distribuzione a
#posteriori di interesse e ' bimodale e quindi per approssimarla
```

```
campionando
#secondo una algoritmo di Metropolis c'e' bisogno a volte di grandi apssi
#per passare da una moda all'altra, altre di piccoli passi in modo che il
#tasso di accettazione non sia basso. Si aumenta cosi' la cosiddetta
#capacita' di mixing dell'algoritmo.
metropolis<-function(tuning, nsimul) {
  #setting della distribuzione a priori:
  pmn.beta<-c(1.1, 0)
  psd.beta<-c(2.75, 0.22)
  #setting della distribuzione proposal: si sceglie come spesso si fa una
  #distribuzione normale multivariata con media vettore nullo e matrice
  #di varianza e covarianza quella specificata in ingresso nella funzione.
  var.prop<-tuning
  #valore inizial del vettore dei coefficenti:
  beta<-rep(0, p)
  #numero di simulazioni
  S<-nsimul
  #Vettore in cui immagazzino i valori campionati della distribuzione a posteriori:
  BETA<-matrix(0, nrow=S, ncol=p)
  #contatore del numero di accetazioni:
  ac<-0
  set.seed(1)
  library(coda)
  #algoritmo metropolis (dal momento che la proposal e' simmetrica siamo in
  #questo caso pasrticolare di Metropolis-Hastings):
   for(s in 1:S){
    #proposta dei coefficenti di regressione:
    beta.p<-t(rmvnorm(1, beta, var.prop))
    #Rapporto di metropolis:
    #puo' essere calcolato in piu' modi tra cui usare la forma funzionale
    #della verosomiglianza trovata al punto a (e' scritto per completezza in
```

```
    #commento) o usando la funzione dbinom di R. Facciamo relativamente a
    #questa alcune considerazioni:
    #-) la funzione prende in ingresso tre elementi: il vettore y delle
    #osservazioni, la dimensione n e il vettore della probabilita' p. In
    #questo caso, dal momento che la dimensione specificata e' 1, la funzione
    #restituisce un vettore della stessa dimensione del vettore risposta il cui
    #i-mo elemento e' il valore della densita' Bernoulliana calcolata
    #nell'elemento i-mo di y con probabiita' pari all'i-mo elemento del
    #vettore delle probabilita' p. La somma dei logaritmi di tali densita' e' la
    #log- verosomiglianza.
    #-) Il vettore di probabilita' specificato nella funzione dbinom e'
    #calcolato secondo la formulazione del modello logistico come xpit del
    #predittore lineare. Il rapporto di metropolis contiene due
    #log-verosomiglianze: una calcolata considerando come vettore di
    #probababilita' quello calcolato con il vettore dei coefficenti
    #all'iterazione corrente e una considerando quello calcolato
    #all'iterazione precedente.
    #-) Il rapporto di metropolis, oltre al rapporto tra verosomiglianze,
    #contiena nche il rapporto tra la densita' della prior specificata per
    #i coefficenti di regressione valutata all'iterazione corrente e quella
    #valutata all'iterazione precedente. Usiamo l'opzione log= TRUE perche'
    #lavoriamo con log-verosomiglizne.
    #-) Nulla viete di lavorare con le verosogiglianze, ma usiamo le
    #log-verosogmilianze perche' migliori da pun punto di vista
```

```
    #computazionale.
    lhr<- sum(log(dbinom(y,1,exp(X%*%beta.p)/(1+exp(X%*%beta.p))))) + dnorm(beta.p[1],pmn.beta[1],psd.beta[1],log=TRUE) + dnorm(beta.p[2],pmn.beta[2],psd.beta[2],log=TRUE)-sum(log(dbinom(y,1,exp(X%*%beta)/(1+exp(X%*%beta)))))-dnorm(beta[1],pmn.beta[1],psd.beta[1],log=TRUE)-dnorm(beta[2],pmn.beta[2],psd.beta[2],log=TRUE)
    }
    if(log(runif(1))<lhr){beta<-beta.p; ac<-ac+1}BETA[s,]<-beta
    Ef<-effectiveSize(BETA)
    #Tasso di accettazione ed effective sample size:
    cat("acceptance rate=", ac/S, "\n")
    cat("effective sample size=", Ef, "\n")
    return (BETA)
  }
  #cerchiamo adesso un numero di iterazioni una matrice di varianze e
  #covarianza dalla distribuzione proposal in modo da avere un buon tasso di
  #accettazione ed una effective sample size di circa 1000 come richiesto
  #dall'esercizio
  #cominciamo con 1000 iterazioni (sicuramente un numero troppo ottimistico)
  #e con una matrice di varianza e covarianza simile a quella della g-prior:
  nsimul<-1000
  var.prop <-var(log(y+1/2))*solve(t(X)%*%X)
  var.prop


}
```

```
[,1] [,2]
[1,] 1.11413865 -0.085406852
[2,] -0.085406852 0.006588971
```

```
beta.post <- metropolis(var.prop, nsimul)
```

```
Acceptance rate = 0.764
```

```
Effective sample size = 51.65653
```