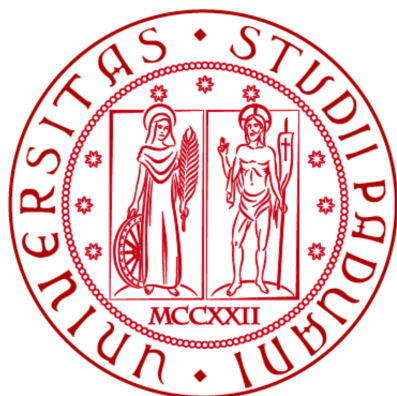


Università degli studi di Padova  
Dipartimento di Scienze Statistiche  
Corso di Laurea Magistrale in  
Statistica



ACCURATEZZA FREQUENTISTA DI STIME BAYESIANE:  
APPLICAZIONE A FAMIGLIE ESPONENZIALI CURVE

Relatore Prof. Nicola Sartori  
Dipartimento di Scienze Statistiche

Laureando: Davide Piatteletti  
Matricola N. 1106734

Anno Accademico 2017/2018



# Indice

<b>Introduzione</b>	<b>5</b>
<b>1 Statistica bayesiana e famiglie esponenziali</b>	<b>7</b>
1.1 Il modello statistico parametrico . . . . .	7
1.2 Impostazione classica e bayesiana . . . . .	8
1.3 Teorema di Bayes e distribuzione a posteriori . . . . .	9
1.3.1 Probabilità a priori e verosimiglianze . . . . .	10
1.4 La distribuzione a priori . . . . .	10
1.4.1 Distribuzioni coniugate . . . . .	11
1.4.2 Distribuzioni non informative . . . . .	11
1.5 Inferenza bayesiana . . . . .	12
1.5.1 Stima puntuale . . . . .	12
1.5.2 Stima intervallare . . . . .	13
1.6 Famiglie esponenziali . . . . .	13
1.6.1 Statistiche sufficienti . . . . .	15
1.6.2 Famiglie esponenziali curve . . . . .	16
1.6.3 Statistica ancillare . . . . .	16
<b>2 Accuratezza frequentista</b>	<b>19</b>
2.1 Formula accuratezza generale . . . . .	19
2.2 Versione bootstrap della formula generale . . . . .	21
<b>3 Famiglie esponenziali piene</b>	<b>25</b>
3.1 Modello normale . . . . .	25
3.2 Modello binomiale con legame logit . . . . .	28
<b>4 Famiglie esponenziali curve</b>	<b>35</b>
4.1 Modello esponenziale tangente . . . . .	35
4.2 Modello parabola normale . . . . .	37
4.3 Modello binomiale con legame probit . . . . .	39

<b>Conclusioni</b>	<b>47</b>
<b>A Funzione accuratezza frequentista</b>	<b>49</b>
<b>B Codici R per il capitolo 3</b>	<b>51</b>
B.1 Modello esponenziale normale . . . . .	51
B.2 Modello esponenziale binomiale . . . . .	54
<b>C Codici R per il capitolo 4</b>	<b>61</b>
C.1 Modello parabola normale . . . . .	61
C.2 Modello binomiale probit . . . . .	65
Riferimenti bibliografici . . . . .	73

# Introduzione

Negli ultimi anni l'uso di tecniche bayesiane è grandemente aumentato nelle applicazioni statistiche dovuto, da un lato, al fatto che la potenza di calcolo dei computer è migliorata, dall'altro, l'utilizzo di strumenti come il metodo Monte Carlo basato sulle catene di Markov (MCMC) ha permesso di ottenere le quantità di interesse indipendentemente dalla forma chiusa della distribuzione a posteriori.

Si supponga di avere dei dati osservati da un dato modello parametrico con relativo parametro e di avere una funzione di suddetto parametro che è la quantità di interesse di riferimento. Applicando la regola di Bayes si ottiene il valore atteso a posteriori. Bisogna verificare, tuttavia, quanto questa stima sia accurata. Una soluzione sarebbe fare inferenza secondo la distribuzione a posteriori bayesiana e, quindi, usare l'errore standard a posteriori.

Scopo di questo lavoro è valutare l'accuratezza di tale stima, a prescindere dalla sua origine bayesiana, da un punto di vista frequentista, calcolandone la sua variabilità considerando  $\hat{\gamma}$  come una funzione dei dati  $y$ . Questo processo viene spiegato nel secondo capitolo in cui si prendono in considerazione due diversi metodi per ottenere la stima dell'accuratezza frequentista: il primo utilizzerà gli algoritmi MCMC, il secondo utilizzerà il bootstrap. Entrambi i metodi sono stati introdotti da Efron (2015).

Nel primo capitolo vengono presentati i concetti base della statistica bayesiana e delle famiglie esponenziali, modelli per cui le formulazioni del capitolo due hanno una versione semplificata.

Nel terzo capitolo vengono applicati i metodi presentati nel secondo capitolo a modelli appartenenti a famiglie esponenziali piene, mentre nel capitolo quattro tale applicazione viene estesa a modelli appartenenti a famiglie esponenziali curve.



# Capitolo 1

## Statistica bayesiana e famiglie esponenziali

In questo capitolo verranno presentati alcuni concetti di base inerenti alla statistica bayesiana e alle famiglie esponenziali: si veda ad esempio Liseo (2008) e Pace e Salvan (1997, Capitolo 5). Tali concetti verranno utilizzati nei successivi capitoli per raggiungere l'obiettivo della nostra tesi.

### 1.1 Il modello statistico parametrico

Sia  $y = (y_1, \dots, y_n)$  realizzazione della variabile casuale  $Y$  con legge di probabilità  $p(y; \theta)$  e sia  $\theta$  il parametro di interesse. Si definisce modello statistico parametrico e si indica con il simbolo  $\Lambda$ , la terna  $\Lambda = (\mathcal{Y}, \mathcal{F}, \Theta)$ , dove:

- lo spazio campionario viene indicato con  $\mathcal{Y}$  e rappresenta l'insieme di tutti i possibili risultati osservabili in un esperimento,
- con  $\mathcal{F}$  si indica la famiglia di tutte le possibili leggi di probabilità  $p(y; \theta)$  che si possono associare alla variabile casuale  $Y$ ,
- lo spazio parametrico per  $\theta$  viene indicato con  $\Theta$  e vale  $\Theta \subseteq \mathbb{R}^p$ .

Scelta la famiglia di appartenenza della legge  $p(y; \theta)$ , il fare inferenza su  $\theta$  sulla base delle osservazioni  $y$  significa identificare con maggiore precisione possibile quale sia il meccanismo generatore dei dati, all'interno del modello  $\Lambda$ .

## 1.2 Impostazione classica e bayesiana

Scopo primario dell'inferenza statistica, almeno nella tradizione classica, è quello di acquisire ulteriore conoscenza su quantità incerte, sulle quali spesso si hanno informazioni parziali, non sufficienti ad eliminare del tutto l'incertezza: questo processo induttivo è spesso la necessaria premessa a un processo decisionale, dove le informazioni in precedenza acquisite, adeguatamente filtrate, vengono utilizzate per scegliere quale decisione prendere, fra diverse possibili.

Ciò che non è stato considerato sono le informazioni a priori a disposizione sui parametri d'interesse. Lo si può fare attraverso il teorema di Bayes. La statistica bayesiana ha come obiettivo il medesimo della statistica classica, tuttavia introduce all'interno del modello statistico parametrico la distribuzione di probabilità *iniziale*. Questo e altro sulle informazioni a priori verrà discusso nel paragrafo dedicato.

Da notare il cambiamento di status dei nostri parametri d'interesse che, mentre nell'impostazione classica sono considerati delle quantità fissate ma incognite, diventano, nell'impostazione bayesiana, delle variabili aleatorie la cui distribuzione iniziale dipende dalle informazioni in nostro possesso in quel dato contesto prima dell'osservazione dei dati. Tale distribuzione è tipicamente soggettiva, ossia varia da individuo a individuo, perchè rappresenta la sintesi delle informazioni che il singolo individuo possiede sul problema specifico.

La prima conclusione che si trae è che le informazioni a priori hanno un ruolo importante nell'inferenza e che tali informazioni introducono nell'analisi una componente soggettiva. Questo aspetto è tuttora fonte di dibattito nell'ambiente scientifico. I sostenitori dell'impostazione bayesiana affermano che il fare uso di informazioni particolari, contingenti, è il modo in cui ogni essere razionale opera in situazioni d'incertezza: del resto persone diverse possono benissimo prendere decisioni differenti anche sulla base di informazioni condivise uguali. Nell'impostazione bayesiana il risultato dell'esperimento statistico viene spesso inteso come non replicabile esattamente e quindi ci si condiziona ad esso. Chi invece critica l'approccio bayesiano sostiene che la statistica, per conservare dignità scientifica e per poter essere usata proficuamente nella pratica, deve garantire una oggettività delle conclusioni che si possono trarre da un esperimento e perciò queste non possono dipendere dalle informazioni di chi conduce l'esperimento. Chi sostiene l'approccio classico, infatti, sostiene il metodo frequentista, cioè la possibilità, almeno ideale, di replicare infinite volte lo stesso esperimento statistico nelle medesi-



me condizioni; l'inferenza si basa quindi sul confronto del risultato osservato con i potenziali risultati ottenibili in queste ipotetiche replicazioni.

Nel seguito del capitolo, si introdurranno i vari elementi formali dell'inferenza bayesiana.

### 1.3 Teorema di Bayes e distribuzione a posteriori

Strumento fondamentale per unire le informazioni a priori sulla quantità di interesse e quelle derivanti dai dati di cui si è in possesso è il teorema di Bayes. Esso rappresenta il modo in cui gli esseri razionali aggiornano le proprie conoscenze sulla base di nuove informazioni disponibili.

**Teorema 1** (di Bayes). *Sia  $E$  un evento contenuto in  $F_1 \cup \dots \cup F_k$ , dove gli eventi  $F_j$ ,  $j=1, \dots, k$  sono a due a due incompatibili e necessari. Allora per ognuno dei suddetti eventi  $F_j$ , per  $j=1, \dots, k$ , vale la seguente formula.*

$$P(F_j|E) = \frac{P(F_j)P(E|F_j)}{\sum_{j=1}^k P(F_j)P(E|F_j)}. \quad (1.1)$$

Tale teorema mette in luce come la probabilità di un evento  $F_j$  dipenda anche dalle informazioni di cui si è in possesso, in questo caso rappresentate dall'evento  $E$ , che sarà il risultato dell'esperimento statistico. Inoltre tale teorema permette l'inversione di probabilità, ossia valutare le probabilità condizionate di varie ipotesi a partire dalle probabilità condizionate delle osservazioni.

Nella formula precedente il membro di sinistra prende il nome di probabilità finale (o a posteriori) dell'evento  $F_j$ : il termine finale sta a significare dopo che è noto che si è verificato l'evento  $E$ . Il denominatore del membro di destra è un semplice fattore di normalizzazione; nel numeratore, invece, compaiono due quantità:  $P(F_j)$  è la probabilità a priori dell'evento  $F_j$  e  $P(E|F_j)$  rappresenta invece la verosimiglianza di  $F_j$ . Perciò l'evento  $F_j$  sarà tanto più plausibile (verosimile) sulla base delle informazioni in nostro possesso tanto maggiore sarà la probabilità che si verifichi l'evento  $E$  condizionatamente al fatto che si sia verificato  $F_j$ .

### 1.3.1 Probabilità a priori e verosimiglianze

Nel contesto di un modello statistico parametrico la formula di Bayes permette di giungere alla distribuzione a posteriori  $\pi(\theta|y)$  per il parametro di interesse  $\theta$ . Indicato con  $\pi(\cdot)$  la distribuzione a priori di  $\theta$  si ottiene che la distribuzione a posteriori ha la seguente espressione generale

$$\pi(\theta|y) = \frac{\pi(\theta)p(y; \theta)}{\int_{\Theta} \pi(\theta)p(y; \theta)d\theta}. \quad (1.2)$$

Generalmente, per  $y$  fissato, si indica con  $L(\theta; y) \propto p(y; \theta)$  la verosimiglianza di  $\theta$ . La Formula (1.2) rende evidente che in ottica bayesiana la quantità di interesse non è più fissata come nell'impostazione classica ma è una variabile aleatoria la cui distribuzione di probabilità è influenzata sia dalle informazioni a priori sia dai dati a disposizione. La distribuzione a posteriori  $\pi(\theta|y)$ , contenendo tutta l'informazione riguardante il parametro  $\theta$ , sarà utilizzata per produrre indicatori sintetici, per la determinazione di stime puntuali o intervallari, e per la verifica d'ipotesi, reperendo da essa, se necessario, le eventuali distribuzioni marginali delle singole componenti di  $\theta$ .

## 1.4 La distribuzione a priori

La scelta della distribuzione iniziale per i parametri presenti nel modello statistico è l'aspetto cruciale dell'impostazione bayesiana. Di fatto, la distribuzione iniziale è il vettore attraverso cui le informazioni extra-sperimentali vengono inserite nel procedimento induttivo, e l'adozione di una distribuzione rende l'analisi statistica, almeno sul piano formale, soggettiva. L'uso di distribuzioni iniziali comporta inoltre problemi di tipo computazionale. A meno di non utilizzare forme analitiche particolari, le cosiddette distribuzioni coniugate al modello statistico utilizzato, risulta spesso difficile ottenere espressioni analitiche in forma esplicita per le distribuzioni finali. Attualmente convivono due diverse filosofie all'interno dell'impostazione bayesiana: quella soggettiva e quella oggettiva.

La filosofia soggettiva implica la scelta della distribuzione a priori che meglio rappresenta le proprie convinzioni iniziali: soggetti diversi potranno specificare sia le stesse distribuzioni di probabilità ma con valori dei parametri che la caratterizzano differenti, sia distribuzioni di probabilità completamente diverse. Da ciò deriva una forte componente soggettiva.

La specificazione di una distribuzione a priori secondo un approccio oggettivo parte dalla debolezza applicativa del metodo soggettivo. Pur riconoscendone la validità teorica, propone la determinazione e l'adozione di di-

istribuzioni a priori di tipo convenzionale, derivabili sulla base delle proprietà matematiche del modello statistico utilizzato. Così, da un lato si cerca sia di conservare la pretesa oggettività delle procedure inferenziali che dipendono in questo modo esclusivamente dal modello statistico prescelto e dal risultato campionario, dall'altro, l'uso di una legge iniziale consente ancora di fare uso del teorema di Bayes per produrre conclusioni inferenziali nel linguaggio probabilistico.

Nel seguito verranno presentate alcune classi di distribuzioni a priori e verranno illustrati brevemente i metodi per l'individuazione di una distribuzione a priori non informativa.

### 1.4.1 Distribuzioni coniugate

Sia  $Y = (Y_1, \dots, Y_n)$  un vettore aleatorio a componenti somiglianti e indipendenti condizionatamente a un vettore di parametri  $\theta \in \Theta$ , con funzione di verosimiglianza

$$L(\theta; y) \propto \prod_{j=1}^n p(y_j; \theta). \quad (1.3)$$

Una distribuzione di probabilità iniziale  $\pi(\theta)$  si dice *coniugata* al modello utilizzato se la forma funzionale della distribuzione iniziale e della distribuzione finale sono uguali. Il risultato sperimentale entrerà in tale meccanismo andando ad aggiornare gli iper-parametri della distribuzione a priori, individuando così i parametri della distribuzione a posteriori. Tali parametri conterranno anche l'informazione disponibile a priori. In generale, l'esistenza di una distribuzione a priori coniugata è garantita quando  $p(y; \theta)$  appartiene alla famiglia esponenziale, descritte nel Paragrafo 1.6.

### 1.4.2 Distribuzioni non informative

Le distribuzioni non informative nascono con l'obiettivo di ottenere delle conclusioni oggettive, pur operando in ambito bayesiano. Segue che tali distribuzioni vengano individuate sulla base del modello statistico scelto per l'esperimento e non sulla base delle informazioni a priori di cui dispone chi conduce l'esperimento. Tali distribuzioni saranno spesso delle distribuzioni improprie, cioè che non rispettano tutte le caratteristiche di una legge di probabilità. Ciò non costituisce un problema fintanto che la distribuzione a posteriori risulta essere una distribuzione propria.

La famiglia delle distribuzioni non informative include, ad esempio, le distribuzioni a priori di Jeffreys o le distribuzioni individuate col metodo delle

*reference prior*.

La distribuzione non informativa di Jefferys è

$$\pi^J(\theta) \propto \sqrt{\det(I(\theta))} \quad (1.4)$$

dove  $I(\theta)$  rappresenta la matrice d'informazione attesa di Fisher relativa ad una singola osservazione. Tali distribuzioni sono invarianti rispetto a cambiamenti di parametrizzazione.

Il metodo delle *reference prior* si pone come il metodo di *default* per l'individuazione di una distribuzione a priori non informativa per un modello statistico. Tale metodo consiste nel massimizzare la distanza di Kullback-Leibler tra la distribuzione a priori e quella a posteriori indotta da essa rispetto alla distribuzione a priori. Si otterrà quindi la distribuzione a priori con minor apporto informativo, cioè quella che massimizza il valore atteso della distanza di Kullback-Leibler, si veda Liseo (2008).

## 1.5 Inferenza bayesiana

L'approccio bayesiano permette di giungere ad una distribuzione finale sintesi della distribuzione a priori arricchita dalla funzione di verosimiglianza dei dati. Come nella statistica classica, anche nell'ambito bayesiano, si riesce a giungere, dalla distribuzione a posteriori, a stime puntuali, intervallari e a condurre delle verifiche d'ipotesi. Nel seguito si darà una breve descrizione dei primi due tipi di inferenza, tralasciando il terzo in quanto non di diretto interesse di questa tesi.

### 1.5.1 Stima puntuale

In ambito bayesiano, una volta determinata la distribuzione a posteriori per  $\theta$ , la sintesi più naturale è fornita da un qualche indicatore sintetico di posizione. La stima puntuale permette di trovare un valore  $\tilde{\theta} = \tilde{\theta}(y) \in \Theta$  che individui quale legge all'interno di  $\mathcal{F}$  abbia agito e generato il risultato sperimentale  $y \in \mathcal{Y}$ . Per semplicità di calcolo e per il suo ruolo anche al di fuori della statistica, il valore atteso  $E(\theta|y)$  è il più usato. A volte il calcolo del valore atteso è complicato, altre volte non esiste. In questi casi è auspicabile ricorrere a stime puntuali alternative come la mediana, la moda o altri quantili della distribuzione a posteriori.

### 1.5.2 Stima intervallare

La stima intervallare permette di individuare un insieme di valori  $\tilde{\Theta}_{1-\alpha} = \tilde{\Theta}_{1-\alpha} \subseteq \Theta$  per il parametro di interesse  $\theta$ , in modo tale che  $P(\theta \in \tilde{\Theta}_{1-\alpha}|y) = 1 - \alpha$ , dove  $1 - \alpha$  è il livello specificato di credibilità. La regione può essere costruita sfruttando i quantili della distribuzione a posteriori di  $\theta$  oppure inserendo in esso i valori di  $\theta$  più plausibili. Nel primo modo, gli estremi dell'intervallo verranno individuati prendendo il quantile, o i quantili, tali per cui la probabilità della regione così individuata sia pari a  $1 - \alpha$ . Possibili intervalli potrebbero essere:

- $\{\theta \in \Theta : \tilde{\theta}_{\alpha/2} \leq \theta \leq \tilde{\theta}_{1-\alpha/2}\}$
- $\{\theta \in \Theta : \theta \leq \tilde{\theta}_{1-\alpha}\}$
- $\{\theta \in \Theta : \theta \geq \tilde{\theta}_{\alpha}\}$

dove  $\tilde{\theta}_{\alpha}$  rappresenta il quantile tale per cui  $P(\theta \leq \tilde{\theta}_{\alpha}|y) = \alpha$ .

Nel secondo modo, l'insieme  $\tilde{\Theta}_{1-\alpha}$  verrà individuato inserendo in esso i valori  $\theta$  ritenuti più plausibili, cioè quelli che presentano densità a posteriori più elevata, finché non si raggiungerà una probabilità complessiva non inferiore al livello  $1 - \alpha$  stabilito. Tale intervallo è anche detto HPD (*High Posterior Density*). La determinazione di un insieme HPD non è sempre possibile per via analitica; può accadere che la distribuzione finale risulti multimodale cosicché l'insieme HPD potrebbe essere addirittura formato da insiemi disgiunti di valori. Nel caso scalare e nella situazione piuttosto comune in cui la distribuzione finale  $\pi(\theta|y)$  sia unimodale, occorre ricorrere a metodi numerici. A grandi linee la procedura consiste in tre passi:

1. Per  $k$  fissato, minimo del massimo valore di  $\pi(\theta|y)$ , determinare le due radici  $\tilde{\theta}_1(k)$  e  $\tilde{\theta}_2(k)$  dell'equazione  $\pi(\theta|y) = k$ .
2. Si calcola  $P(k) = Pr(\theta \in (\tilde{\theta}_1(k), \tilde{\theta}_2(k))|y)$ .
3. Si determina  $k^*$  che risolve l'equazione  $P(k) = 1 - \alpha$ .

L'intervallo HPD sarà quindi

$$(\tilde{\theta}_1(k^*), \tilde{\theta}_2(k^*)).$$

## 1.6 Famiglie esponenziali

Si consideri una variabile non degenera con supporto  $\mathcal{Y} \subseteq \mathbb{R}$  e densità  $p_0(y)$  rispetto ad una misura  $\mu^*$ . Una famiglia parametrica di densità che include

$p_0(y)$  come caso speciale e i cui elementi hanno lo stesso supporto  $\mathcal{Y}$  possono essere costruiti attraverso *exponential tilting*. Tale termine, introdotto da Efron (1981), è usato perchè le densità nella famiglia sono, come funzione di  $y$ , proporzionali a  $\exp(\theta y)p_0(y)$ .

La famiglia parametrica

$$\mathcal{F}_{ne}^1 = \left\{ p(y; \theta) = \exp\{\theta y - K(\theta)\}p_0(y), y \in \mathcal{Y}, \theta \in \tilde{\Theta} \right\} \quad (1.5)$$

è una famiglia esponenziale naturale di ordine 1 generata da  $p_0(y)$ , dove  $K(\theta) = \log \int_{\mathcal{Y}} \exp(\theta y)p_0(y)d\mu^*$  e  $\tilde{\Theta} = \{\theta \in \mathbb{R} : K(\theta) < +\infty\}$ ;  $\tilde{\Theta}$  è chiamato spazio parametrico naturale.

Tutti gli elementi di  $\mathcal{F}_{ne}^1$  hanno lo stesso supporto. La famiglia esponenziale è definita piena se la dimensione dello spazio campionario  $\mathcal{Y}$  coincide con quella dello spazio parametrico  $\tilde{\Theta}$  e se quest'ultimo è aperto la famiglia viene chiamata regolare.

Si può estendere il discorso al caso in cui la dimensione del parametro sia più grande di uno. La famiglia parametrica

$$\mathcal{F}_{ne}^p = \left\{ p(y; \theta) = \exp\{\theta^T y - K(\theta)\}p_0(y), y \in \mathcal{Y} \subseteq \mathbb{R}^p, \theta \in \tilde{\Theta} \right\} \quad (1.6)$$

dove  $K(\theta) = \log \int_{\mathcal{Y}} \exp(\theta y)p_0(y)d\mu^*$  e  $\tilde{\Theta} = \{\theta \in \mathbb{R}^p : K(\theta) < +\infty\}$ , è chiamata famiglia esponenziale naturale di ordine  $p$  generata da  $p_0(y)$ . Analogamente a quanto detto per la famiglia esponenziale di ordine 1, la famiglia  $\mathcal{F}_{ne}^p$  è detta regolare se  $\tilde{\Theta}$  è un insieme aperto di  $\mathbb{R}^p$ .

Tale estensione ci permette di definire una famiglia esponenziale di ordine  $p$ , anche se la dimensione sia dello spazio campionario che dello spazio parametrico non devono necessariamente coincidere con  $p$ .

Sia  $Y$  un vettore di dimensione  $d$ , con  $d \geq 1$  e densità  $p_0(y)$ ; siano inoltre  $s_1(y), \dots, s_p(y)$  statistiche scalari linearmente indipendenti. Sia  $K_S(\theta)$  la funzione generatrice dei cumulanti di  $(s_1(y), \dots, s_p(y))$  e  $\tilde{\Theta}_S = \{\theta \in \mathbb{R}^p : K_S(\theta) < +\infty\}$ . In aggiunta sia  $\theta$  in funzione di  $\phi$ , per esempio  $\theta = \theta(\phi)$ ,  $\phi \in \Phi$ ,  $\theta(\Phi) \subseteq \tilde{\Theta}_S$ ; pertanto la classe di densità risulta avere la forma:

$$\mathcal{F}_e^p = \{p(y; \theta) = \exp\{\theta(\phi)s(y) - G(\phi)\}h(y), y \in \mathcal{Y} \subseteq \mathbb{R}^d, \phi \in \Phi\} \quad (1.7)$$

ed è chiamata famiglia esponenziale di ordine  $p$ . La notazione  $\mathcal{F}_e^p$  indica la rappresentazione minima, cioè non è presente alcuna dipendenza lineare tra le componenti  $s_1(y), \dots, s_p(y)$ ; inoltre  $\theta$  viene chiamato parametro canonico o naturale e le statistiche  $s_1(y), \dots, s_p(y)$  sono chiamate osservazioni naturali. Se  $\Phi \subseteq \mathbb{R}^q$  ci possono essere varie situazioni in relazione al rapporto tra le dimensioni  $q$  e  $p$  dello spazio parametrico  $\Phi$  e  $\tilde{\Theta}_S$ :

- $q > p$  allora il parametro  $\phi$  non è identificabile,
- $q = p$  e  $\theta(\phi)$  è iniettiva allora  $\theta$  è una riparametrizzazione; se  $\theta(\phi)$  non è iniettiva allora  $\phi$  non è identificabile,
- $q < p$  si ha una famiglia esponenziale curva  $(p, q)$ , che verrà indicata con  $\mathcal{F}_e^{p,q}$ .

### 1.6.1 Statistiche sufficienti

Un'immediata applicazione del teorema di fattorizzazione di Neyman-Fisher mostra che le osservazioni naturali di una  $\mathcal{F}_e^p$  sono una statistica sufficiente di dimensione  $p$ , come si vede in Pace e Salvan (1997, Capitolo 5.5).

**Teorema 2** (di Neyman-Fisher). *Sia  $\mathcal{F} = \{p(y; \theta), \theta \in \Theta\}$  un modello statistico parametrico. Una statistica  $s$  è statistica sufficiente per l'inferenza su  $\theta$  se e solo se la funzione del modello  $p(y; \theta)$  ha forma*

$$p(y; \theta) = h(y)k(s(y); \theta). \quad (1.8)$$

Tale risultato è particolarmente importante perchè la classe di famiglie esponenziali di ordine  $p$  è legato al campionamento casuale di dimensione  $n$ . Siano  $Y_1, \dots, Y_n$  copie indipendenti di una variabile casuale con densità  $p(y; \phi) \in \mathcal{F}_e^p$ . La densità di  $Y = (Y_1, \dots, Y_n)$  è

$$p(y; \phi) = \exp(\theta(\phi) \sum_{i=1}^n s(y_i) - nG(\phi)) \prod_{i=1}^n h(y_i), \quad y \in \mathcal{Y}^n, \phi \in \Phi.$$

Questa è ancora una  $\mathcal{F}_e^p$  e la statistica sufficiente, in quanto  $k(s(y); \phi) = \exp(\theta(\phi)s(y) - nG(\phi))$ ,  $s = (s_1(y), \dots, s_p(y))$ , con  $s_r(y) = \sum_{i=1}^n s_r(y_i)$ , ha dimensione fissata  $p$  indipendentemente dalla dimensione del campione. Poichè la rappresentazione è minimale di  $\mathcal{F}_e^p$ , allora  $s(y)$  è statistica sufficiente minimale se  $n \geq p$ .

### 1.6.2 Famiglie esponenziali curve

In una famiglia esponenziale curva la dimensione  $p$  del vettore delle osservazioni è più grande della dimensione  $q$  dello spazio parametrico  $\Phi$ . Questa classe è particolarmente interessante per la riduzione dei dati e del modello. Inoltre, poichè il vettore delle osservazioni naturali non è necessariamente completo, c'è la possibilità di una riduzione condizionandosi ad una statistica ancillare, così da recuperare l'informazione persa nella riduzione alla stima di massima verosimiglianza. Gli elementi di una famiglia esponenziale curva  $\mathcal{F}_e^{p,q}$  hanno densità:

$$p(y; \phi) = \exp\{\theta(\phi)s(y) - G(\phi)\}h(y), \quad (1.9)$$

dove  $y \in \mathcal{Y} \subseteq \mathbb{R}^d$ ,  $\phi \in \Phi \subseteq \mathbb{R}^q$ ,  $s(y)$  e  $\theta(\phi)$  hanno dimensione  $p > q$  e si assuma che la rappresentazione (1.9) sia minimale. Una famiglia esponenziale curva,  $\mathcal{F}_e^{p,q}$ , può essere pensata come una sotto-famiglia di una famiglia esponenziale piena di ordine  $p$ ,  $\mathcal{F}_e^p$ , con densità

$$p(y; \theta) = \exp\{\theta s(y) - K(\theta)\}h(y), \quad (1.10)$$

con  $\theta \in \tilde{\Theta}$ , dove  $\tilde{\Theta}$  è lo spazio parametrico naturale

$$\tilde{\Theta} = \left\{ \theta \in \mathbb{R}^p : \int_{\mathcal{Y}} \exp(\theta s(y))h(y)d\mu^* < +\infty \right\}.$$

La costante di normalizzazione  $G(\phi)$  in (1.9) è  $G(\phi) = K(\theta(\phi))$ . Da un punto di vista generale, lo studio di famiglie esponenziali curve è interessante perchè una famiglia parametrica sufficientemente regolare può essere localmente approssimata, per un ordine adeguato, da una famiglia esponenziale curva Efron (1975).

### 1.6.3 Statistica ancillare

**Definizione 1.** *Sotto un modello parametrico  $\mathcal{F}$  con statistica sufficiente  $s$ , una statistica  $a$  è chiamata ausiliaria se  $(\hat{\theta}, a)$  è una funzione uno a uno di  $s$ , dove  $\hat{\theta}$  è la stima di massima verosimiglianza di  $\theta$ . Se  $a$  è ausiliaria e costante in distribuzione,  $a$  è chiamata ancillare.*

Tale statistica viene usata per recuperare l'informazione persa quando si riduce una statistica sufficiente  $s$  alla stima di massima verosimiglianza  $\hat{\theta} = \hat{\theta}(s)$  che generalmente non mantiene la sufficienza. Se  $a$  è ancillare, si ottiene la seguente fattorizzazione:

$$p_{\hat{\theta}, A}(\hat{\theta}, a; \theta) = p_A(a)p_{\hat{\theta}|A=a}(\hat{\theta}; a, \theta),$$



quindi  $\hat{\theta}$  è sufficiente minimale nel modello condizionato al valore osservato di  $a$  Fisher (1934). Una statistica ancillare può recuperare l'informazione persa nella riduzione da  $s$  a  $\hat{\theta}$ . Nei casi in cui non ci sia una statistica ancillare esatta o nel caso in cui non si sia in grado di identificarla, si ricorre ad una soluzione approssimata, per esempio una statistica ausiliaria che è asintoticamente costante in distribuzione.

18CAPITOLO 1. STATISTICA BAYESIANA E FAMIGLIE ESPONENZIALI

## Capitolo 2

# Accuratezza frequentista delle stime bayesiane

Negli ultimi vent'anni si sono usati sempre in maniera maggiore le tecniche bayesiane per le applicazioni statistiche. I metodi bayesiani, basati su priori neutrali o non informative, dominano queste applicazioni, portate avanti dalla popolarità degli algoritmi Markow chain Monte Carlo(MCMC). Tale scelta viene spesso presa per convenienza computazionale ma, in realtà, nella fase di analisi successiva si vuole interpretare il risultato da un punto di vista frequentista. Questa argomentazione viene anche affrontata in Efron e Hastie (2016). Pertanto si propone di calcolare l'accuratezza frequentista del valore atteso a posteriori, considerando tale quantità come una funzione dei dati e calcolando la sua variabilità dal punto di vista frequentista. I risultati riportati di seguito sono tratti da Efron (2015).

### 2.1 Formula accuratezza generale

Si vuole stimare l'accuratezza frequentista della stima a posteriori del valore atteso  $\hat{\gamma} = E\{t(\theta)|y\}$  dove  $t(\theta)$  è il parametro di interesse,  $\theta$  è un vettore di parametri sconosciuti che esistono nello spazio parametrico  $\Theta$  con distribuzione a priori  $\pi(\theta)$ ,  $y$  invece indica la statistica sufficiente che assume valori nello spazio  $p$ -dimensionale  $y \in \mathbb{R}^p$ , ottenuti dalla densità  $p_\theta(y)$  da una famiglia parametrica  $\mathcal{F} = \{p_\theta(y), \theta \in \Theta\}$ . Si scrive il valore atteso e la matrice  $p \times p$  di varianza e covarianza di  $y$  condizionato a  $\theta$  come  $y \sim (m_\theta, V_\theta)$ . Si indichi il gradiente di  $\log\{p_\theta(y)\}$  rispetto ad  $y$  come:

$$\alpha_y(\theta) = \nabla_y \log\{p_\theta(y)\} = \left( \frac{\partial}{\partial y_1} \log\{p_\theta(y)\}, \dots, \frac{\partial}{\partial y_n} \log\{p_\theta(y)\} \right)^T. \quad (2.1)$$

**Lemma 1.** *Il gradiente di  $\hat{\gamma} = E\{t(\theta)|y\}$  rispetto ad  $y$  è la covarianza a posteriori di  $t(\theta)$  con  $\alpha_y(\theta)$ :*

$$\nabla_y \hat{\gamma} = \text{cov}\{t(\theta), \alpha_y(\theta)|y\} \quad (2.2)$$

Il Lemma 1 porta direttamente alla *formula di accuratezza generale*; generale nel senso che è applicabile a qualsiasi distribuzione a priori, non solamente a quelle non informative.

**Teorema 3.** *Il metodo delta di approssimazione per la deviazione standard frequentista di  $\hat{\gamma}$  è*

$$\widehat{sd} = [\text{cov}\{t(\theta), \alpha_y(\theta)|y\}^T V_{\hat{\theta}} \text{cov}\{t(\theta), \alpha_y(\theta)|y\}]^{\frac{1}{2}} \quad (2.3)$$

dove  $\hat{\theta}$  è il valore di  $\theta$  avendo  $m_{\hat{\theta}} = y$ .

Di seguito verranno trattati solo alcuni aspetti della formula di accuratezza generale:

- *Implementazione:* si supponga che  $\{\theta_1, \theta_2, \dots, \theta_B\}$  sia un campione di dimensione  $B$  dalla distribuzione a posteriori di  $\theta$  dato  $y$ . Ogni  $\theta_i$  porge il corrispondente valore di  $t(\theta_i)$  e  $\alpha_y(\theta_i)$  che indicheremo rispettivamente con  $t_i$  e con  $\alpha_i$ . Quindi, la quantità  $\bar{t} = \sum t_i/B$  approssima il valore atteso a posteriori  $\hat{\gamma}$ . Inoltre

$$\widehat{cov} = \sum_{i=1}^B (\alpha_i - \bar{\alpha})(t_i - \bar{t})/B, \quad (2.4)$$

con  $\bar{\alpha} = \sum \alpha_i/B$ , indica la stima della covarianza a posteriori (2.2). Le stesse simulazioni che portano a determinare  $\hat{\gamma}$ , provvedono anche all'ottenimento della sua deviazione standard dal punto di vista frequentista.

- *Campionamento a posteriori:* il campione a posteriori  $\{\theta_1, \theta_2, \dots, \theta_B\}$  si può ottenere attraverso MCMC. La dipendenza dei  $\theta_i$  non invalida l'espressione (2.4) ma suggerisce che sia richiesto un valore di  $B$  grande per l'accuratezza computazionale. Il campionamento a posteriori basato sul bootstrap del Paragrafo 2.2 produce  $\theta_i$  indipendenti.
- *Famiglie esponenziali:* Nel Paragrafo 2.2 si mostra che, in presenza di famiglie esponenziali,  $\alpha_y(\theta)$  ha una forma semplificata, indipendente da  $y$ .

- *Vettore dei parametri di interesse:* Il Lemma 1 e il Teorema 3 sono applicati anche al caso in cui il parametro di interesse  $t(\theta)$  risulta essere un vettore di valori di dimensione  $q$ . Allora  $\nabla_y(\hat{\gamma})$  e  $\text{cov}\{t(\theta), \alpha_y(\theta)|y\}$  nell'equazione (2.2) diventano una matrice  $p \times q$ , rendendo la matrice di covarianza approssimata  $\widehat{\text{var}}$  per  $\hat{\gamma} = E\{t(\theta)|y\}$ ,

$$\widehat{\text{var}} = \text{cov}\{t(\theta), \alpha_y(\theta)|y\}^T V_{\hat{\theta}} \text{cov}\{t(\theta), \alpha_y(\theta)|y\}. \quad (2.5)$$

con  $\alpha_y(\theta)$  e  $V_{\hat{\theta}}$  definiti come in precedenza.

In Efron (2015) vengono considerati anche altri aspetti, come ad esempio: *Fattorizzazione, Sufficienza, Statistica discreta e Correzione della distorsione della deviazione standard.*

## 2.2 Versione bootstrap della formula generale

Un possibile svantaggio del metodo precedente è l'esigenza di avere un campione di  $\{\theta_1, \dots, \theta_B\}$  da  $\pi(\theta|x)$ . In questo paragrafo si discute un approccio che prevede di applicare un bootstrap parametrico alla formula generale (2.3) che, da un lato, elimina il campionamento a posteriori al costo di una perdita di generalità. Infatti, si assumono famiglie esponenziali e distribuzioni a priori  $\pi(\theta)$  non informative; dall'altro il metodo bootstrap rende l'analisi dell'errore computazionale chiara ed aiuta il collegamento tra il punto di vista bayesiano e frequentista.

Nella famiglia esponenziale  $\mathcal{F}$  (1.7)  $\theta$  è il vettore di parametri naturali o canonici e  $s(y)$  è la statistica sufficiente  $p$ -dimensionale. Il valore atteso del parametro  $\beta = E_{\theta}(s(y))$  è una funzione iniettiva di  $\theta$ , del tipo  $\beta = A(\theta)$ . Lo spazio parametrico  $\Theta$  per  $\theta$  è un sottoinsieme di  $\mathbb{R}^p$  come lo è lo spazio per  $\beta$ .

Il valore atteso e la varianza di  $s(y)$  dato  $\theta$  possono essere scritte come  $s(y) \sim (\beta, V_{\theta})$  e sono ottenute differenziando  $K(\theta)$ .

**Teorema 4.** *L'approssimazione del metodo delta per la deviazione standard frequentista di  $\hat{\gamma} = E(t(\theta)|s(y))$  nella famiglia esponenziale (1.7) è*

$$\widehat{\text{sd}} = [\text{cov}\{t(\theta), \theta|s(y)\}^T V_{\hat{\theta}} \{\text{cov}\{t(\theta), \theta|s(y)\}\}]^{\frac{1}{2}} \quad (2.6)$$

dove  $\hat{\theta}$ , il vettore di parametri naturali corrispondenti a  $s(y)$ , è la stima di massima verosimiglianza di  $\theta$ .

Nel Teorema 4, i valori  $\theta$ , ottenuti dal Lemma 1, hanno una versione più semplice quando la nostra distribuzione appartiene alla famiglia esponenziale:

$$\begin{aligned}\alpha_y(\theta) &= \nabla_{s(y)} \log\{p_\theta(s(y))\} \\ &= \nabla_{s(y)} \{\theta s(y) - K(\theta) + \log\{p_0(s(y))\}\} \\ &= \theta + \nabla_{s(y)} \log\{p_0(s(y))\}.\end{aligned}\tag{2.7}$$

Come si vede nella (2.7),  $\alpha_y(\theta)$  diventa il parametro canonico  $\theta$ , in quanto l'ultimo termine a destra non dipende da  $\theta$ ; di conseguenza la Formula (2.3) si semplifica nella (2.6).

Un bootstrap parametrico può essere utilizzato per calcolare  $\hat{\gamma}$  e la deviazione standard, come suggerito in Efron (2012). Ricampioniamo indipendentemente  $B$  volte da  $\mathcal{F}$  avente vettore dei parametri  $\theta$  uguale a  $\hat{\theta}$ :

$$f_{\hat{\theta}}(\cdot) \rightarrow \{\beta_1^*, \beta_2^*, \dots, \beta_i^*, \dots, \beta_B^*\}.\tag{2.8}$$

Ogni  $\beta_i^*$  porge un vettore corrispondente di parametri naturali  $\theta_i = A^{-1}(\beta_i^*)$ . Sia  $\pi_i = \pi(\theta_i)$  e definiamo il fattore di conversione come:

$$R_i = \frac{f_{\theta_i}(s(y))}{f_{\hat{\theta}}(\beta_i^*)}\tag{2.9}$$

il rapporto tra la verosimiglianza e la densità bootstrap.

La distribuzione discreta che pone i pesi

$$p_i = \frac{\pi_i R_i}{\sum_{j=1}^B \pi_j R_j}\tag{2.10}$$

su  $\theta_i$ , per  $i = 1, \dots, B$ , approssima la distribuzione condizionata di  $\theta$  dato  $s(y)$ . Pertanto, assumendo che  $t_i = t(\theta_i)$  e  $\hat{\gamma}_B = \sum_{i=1}^B t_i p_i$ , si avrà:

$$\hat{\gamma}_B = \frac{\sum_{i=1}^B t_i \pi_i f_{\theta_i}(s(y)) / f_{\hat{\theta}}(\beta_i^*)}{\sum_{i=1}^B \pi_i f_{\theta_i}(s(y)) / f_{\hat{\theta}}(\beta_i^*)}$$

Dato che  $s_i$  sono stati ottenuti dalla densità bootstrap  $f_{\hat{\theta}}(\cdot)$ , l'equazione 2.2 rappresenta un *importance sampling* di:

$$\frac{\int_{\Theta} t(\theta) \pi(\theta) f_{\theta}(s(y)) d\theta}{\int_{\Theta} \pi(\theta) f_{\theta}(s(y)) d\theta},\tag{2.11}$$

che è uguale a  $E\{t(\theta)|s(y)\}$ .

Lo stesso argomento può essere applicato a qualsiasi calcolo a posteriori. In particolare,  $cov\{t(\theta), \theta|s(y)\}$  nell'espressione (2.6) è approssimata da

$$\widehat{cov} = \sum_{i=1}^B p_i(\theta_i - \bar{\theta})(t_i - \hat{\gamma}), \quad \bar{\theta} = \sum p_i\theta_i, \quad \hat{\gamma} = \sum p_it_i. \quad (2.12)$$

Si può implementare il risultato Teorema 4 con le seguenti tre fasi:

1. Generare un campione tramite bootstrap parametrico  $\beta_1^*, \dots, \beta_B^*$ , come nella (2.8).
2. Per ogni  $\beta_i^*$  calcolare  $\theta_i$ ,  $t_i = t(\theta_i)$  e  $p_i$ , come nella (2.10).
3. Calcolare  $\widehat{cov}$  usando la (2.12).

Pertanto  $\hat{\gamma}_B = \sum p_it_i$  approssima  $\hat{\gamma} = E\{t(\theta)|s(y)\}$  e ha deviazione standard dal punto di vista frequentista tramite metodo delta pari a

$$\hat{sd} = (\widehat{cov}^T V_{\hat{\theta}} \widehat{cov})^{\frac{1}{2}} \quad (2.13)$$

La matrice  $V_{\hat{\theta}}$  può essere sostituita dalla matrice di covarianza empirica di  $\beta_1, \dots, \beta_B$  o dall'inversa della matrice di covarianza di  $\theta_1, \dots, \theta_B$ .

Un campione ottenuto tramite MCMC  $\{\theta_i, i = 1, \dots, B\}$  approssima una distribuzione a posteriori multidimensionale attraverso una distribuzione ugualmente pesata su  $B$  punti dipendenti. L'implementazione bootstrap (2.8)-(2.10) pone pesi diversi su  $B$  punti indipendenti e identicamente distribuiti.

Il fattore di conversione (2.9) può essere definito in qualsiasi famiglia  $\{f_{\theta}(s(y))\}$ , ma ha una semplice espressione per le famiglie esponenziali, ossia

$$R_i = \xi(\theta_i) \exp\{\Delta(\theta_i)\} \quad (2.14)$$

dove  $\Delta(\theta)$  è la *half-deviance difference*

$$\Delta(\theta) = (\theta - \hat{\theta})^T(\beta + s(y)) - 2\{K(\theta) - K(\hat{\theta})\} \quad (2.15)$$

e

$$\xi(\theta) = \frac{1}{\pi^{Jeff}(\theta)} \quad (2.16)$$

con  $\pi^{Jeff}(\theta) = |V_{\theta}|^{\frac{1}{2}}$ , priori di Jeffreys per  $\theta$ . Se la distribuzione a priori  $\pi(\theta)$  è  $\pi^{Jeff}(\theta)$  allora

$$\pi_i R_i = \exp\{\Delta(\theta_i)\}. \quad (2.17)$$

La distribuzione bootstrap  $f_{\hat{\theta}}(\cdot)$  pone i  $\theta_i$  ricampionati vicino alla stima di massima verosimiglianza  $\hat{\theta}$ . Una priori informativa  $\pi(\theta)$  potrebbe essere una priori che porta a valori distanti da  $\hat{\theta}$ . In tale caso  $R_i$  assume valori enormi destabilizzando l'*importance sampling*.



# Capitolo 3

## Famiglie esponenziali piene

Obiettivo di questo capitolo è quello di applicare i metodi dei Paragrafi 2.1 e 2.2 a famiglie esponenziali piene. In particolare si vuole confrontare la stima frequentista della deviazione standard bayesiana con la deviazione standard bayesiana. Inoltre si vuole valutare l'accuratezza di tale stima frequentista attraverso uno studio di simulazione. In Efron (2015) è presente il modello normale con algoritmo MCMC e il modello binomiale con la versione bootstrap.

### 3.1 Modello normale

Si prendono in considerazione i dati su pazienti affetti da diabete di Efron *et al.* (2004) in cui si hanno 442 soggetti con il diabete. La variabile risposta  $y$  indica la misura della progressione della malattia dopo un anno dall'entrata nello studio. Per ognuno di questi soggetti si è osservato un vettore  $x$  di 10 variabili: età, sesso, indice di massa corporea, pressione del sangue e sei misure del siero sanguigno.

Si assume un modello lineare normale:

$$y = X\beta + \epsilon \quad \epsilon \sim N_n(0, I), \quad (3.1)$$

dove  $X$  è la matrice dei predittori, precedentemente standardizzata, di dimensione  $442 \times 10$ . Park e Casella (2008) hanno applicato una versione bayesiana del lasso (Tibshirani, 2011) ai dati presi in esame. Rispetto al modello (3.1) Park e Casella (2008) hanno preso come distribuzione a priori per  $\beta$

$$\pi(\beta) = \exp\{-\lambda L_1(\beta)\} \quad (3.2)$$

dove  $L_1(\beta)$  indica la norma prima  $\sum_{j=1}^{10} |\beta_j|$  e  $\lambda$  è stato fissato pari a 0.37.

Tramite MCMC, utilizzando il Metropolitan-Hastings adattivo (pacchetto *MHadaptive*), si è prodotto un campione di  $B = 100000$  valori di  $\beta_i$  dalla distribuzione a posteriori  $\pi(\beta|y)$  sotto le assunzioni (3.1) e (3.2). Attraverso tali valori è possibile approssimare il valore atteso bayesiano  $\hat{\gamma} = E(t(\beta)|y)$  per qualsiasi parametro di interesse  $t(\beta)$  con

$$\hat{\gamma} = \sum_{i=1}^B t(\beta_i)/B. \quad (3.3)$$

Si può quindi applicare il Teorema 3 per stimare la deviazione standard dal punto di vista frequentista di  $\hat{\gamma}$  attraverso

$$\hat{sd} = (\widehat{cov} V \widehat{cov})^{0.5}. \quad (3.4)$$

Per calcolare la covarianza si è utilizzata la formula (2.4), in cui gli  $\alpha_i$  sono stati ottenuti dal Lemma 1 mentre i  $t_i$  indicano il parametro d'interesse, infine  $V$  indica la stima della varianza dei parametri di interesse. In questo caso gli  $\alpha_i$  saranno pari a:

$$\alpha = y - X\beta. \quad (3.5)$$

Poichè il modello è una famiglia esponenziale piena con densità:

$$p(y_i; \beta) = \exp\left\{\sum (y_i x_i) \beta - \sum (x_i \beta)\right\} p_o(y_i),$$

gli  $\alpha$  sono pari al parametro canonico  $\beta$ . Gli  $\alpha$  e i  $\beta$  hanno dimensione differente: i primi sono un vettore lungo 15000 mentre i secondi sono una matrice di valori simulati  $15000 \times 10$ . Quindi per il calcolo della covarianza si possono usare indistintamente sia gli  $\alpha$  ottenuti da (3.5) che i valori simulati  $\beta$ . Il valore della deviazione standard risulta essere invariato, questo è dovuto al fatto che la covarianza viene moltiplicata per la matrice di varianza  $V$ .

Come parametro di interesse si è considerato

$$\gamma_{125} = x_{125}\beta \quad (3.6)$$

cioè la progressione del diabete per il paziente 125. I valori ottenuti dall'algoritmo MCMC sono 15000, in quanto 25000 sono stati utilizzati per il *burn-in* e poi si è tenuto un valore ogni 5 per ridurre la correlazione dei valori ottenuti. Tali valori sembrano ben approssimati da una  $N(0.2156, 0.0758^2)$ , come si vede nella Figura 3.1a. Nella prima riga della Tabella 3.1 sono riportate la stima del valore atteso e le stime delle deviazioni standard dal punto di

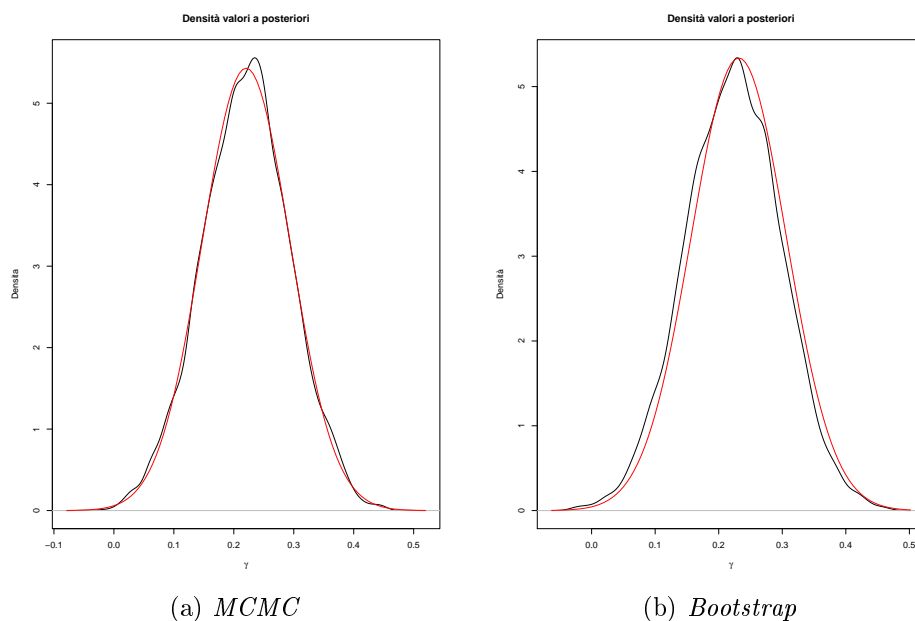


Figura 3.1: Densità a posteriori dei valori simulati  $\gamma_{125}$ .

vista frequentista e bayesiano.

Si è applicato, anche, la versione bootstrap della formula di accuratezza bayesiana. Si sono generati 8000 campioni bootstrap di  $y$  partendo da un modello normale che ha come media la stima di massima verosimiglianza e come varianza il valore unitario. Quindi si sono calcolate le nuove stime di massima verosimiglianza a partire dalle nuove  $y$  e si sono calcolati i pesi che andranno a trasformare la verosimiglianza in una distribuzione a posteriori. Per ottenere i pesi (2.10) si è prima andati a calcolare il fattore di conversione, (2.9), in cui si è fatto il rapporto tra la verosimiglianza calcolata per i nuovi  $\beta_i$  e il vettore  $y$  originale e la verosimiglianza calcolata con la stima di massima verosimiglianza originale e la nuova  $y$ ; una volta fatto questo si sono moltiplicate le quantità  $R_i$  appena ottenute con i  $\pi_i$ , che si ottengono inserendo i  $\beta_i$  nella priori (3.2). Le stime a posteriori di  $\hat{\gamma}$ , della deviazione standard dal punto di vista frequentista e quella bayesiana, ottenuta come  $\{\sum_i p_i (\gamma_i - \hat{\gamma})^2\}^{0.5}$ , sono riportate nella seconda riga della Tabella 3.1.

Infine si è fatto uno studio di simulazione Monte Carlo. Si sono simulati 1000 vettori  $y$  partendo da una normale con media la stima di massima verosimiglianza e varianza unitaria, quindi, per ogni vettore  $y$ , si è applicato l'algoritmo MCMC, con Metropolitan-Hastings adattivo, e si sono ottenute,

Tabella 3.1: Confronto stime ottenute con formula generale, versione bootstrap e con il principio del campionamento ripetuto.

	$\hat{\gamma}$	SD frequentista	SD bayesiana
Formula generale	0.2205	0.0734	0.0734
Bootstrap	0.2315	0.0744	0.0747
PCR	0.2186	0.0756	0.0756

per la quantità d'interesse (3.6), le stime del valore atteso, della deviazione standard bayesiana e frequentista. Ottenute tali stime si è fatta la media aritmetica per tutte e tre le quantità. I risultati sono riportati nella terza riga della Tabella 3.1.

Dal confronto dei valori ottenuti con i tre metodi nella Tabella 3.1 si vede che la stima frequentista e bayesiana della deviazione standard sono praticamente identiche.

Come ultima analisi si è voluto andare a vedere che differenza avessero la stima frequentista e bayesiana della deviazione standard in ogni campione che è stato generato nello studio di simulazione. Pertanto si sono rapportate graficamente le stime delle due quantità mettendo quella bayesiana sull'asse delle ascisse e quella frequentista sull'asse delle ordinate.

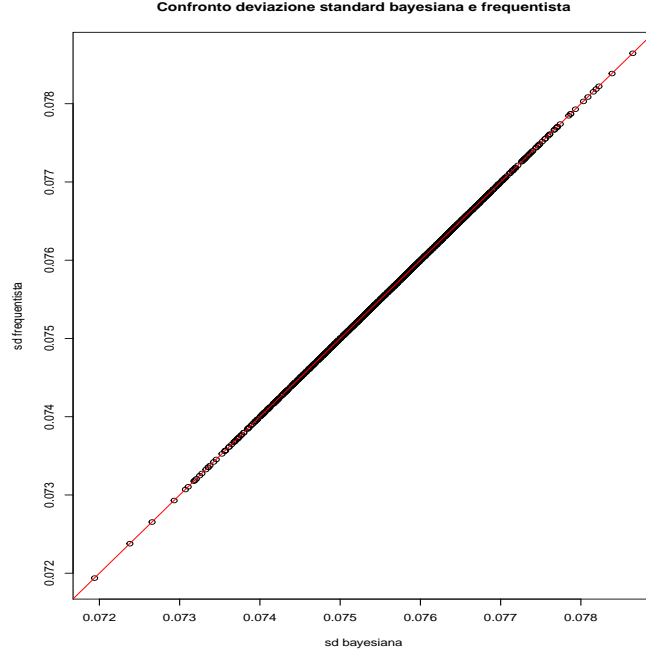
Come si vede nella Figura 3.2, le due deviazioni sono praticamente uguali per tutti i campioni simulati.

Infine si è confrontata la deviazione standard dei 1000 campioni Monte Carlo del valore atteso a posteriori con le medie delle deviazioni standard frequentista e bayesiana. I valori ottenuti sono praticamente identici: 0.0749 per la deviazione standard del valore atteso, 0.0756 la deviazione standard frequentista e 0.0756 quella bayesiana.

## 3.2 Modello binomiale con legame logit

Nella Tabella 3.2 sono riportati i dati relativi all'infusione cellulare tratti da Efron (2015). Le colonie di cellule di muscolo umano sono state instillate con nuclei di topo. Sono state provate cinque proporzioni crescenti di infusione di nuclei di topo, coltivati per periodi di tempo che vanno da uno a cinque giorni e viene osservato se hanno prosperato. La Tabella 3.2 mostra che, per esempio, 52 su 61 colonie hanno prosperato nella categoria proporzione-giorno (5, 5). Siano  $(s_{jk}, n_{jk})$  il numero dei successi e il numero delle colonie

Figura 3.2: Confronto deviazione standard bayesiana e frequentista per i 1000 campioni Monte Carlo.



della  $jk$ -sima cella, si assume che siano indipendenti e abbiano distribuzione binomiale:

$$s_{jk} \sim Bi(n_{jk}, \xi_{jk}) \quad j, k = 1, \dots, 5. \quad (3.7)$$

Il seguente modello logistico si adatta bene i dati

$$\text{logit}(\xi_{jk}) = \alpha_0 + \alpha_1 I_j + \alpha_2 I_j^2 + \alpha_3 D_k + \alpha_4 D_k^2, \quad j, k = 1, \dots, 5 \quad (3.8)$$

dove la matrice  $X$  è di dimensioni  $25 \times 5$  e per ogni riga avremo  $I_j$  che indica le proporzioni di infusione e  $D_k$  che indica i giorni. La log-verosimiglianza per il modello (3.7) con funzione legame (3.8) diventa:

$$l(y; \alpha) \propto \sum_{i=1}^n \{y_i X_i \alpha - \log(1 + \exp(X_i \alpha))\} \quad (3.9)$$

dove  $X_i$  è l' $i$ -esima riga di  $X$ . Come noto il modello (3.9) appartiene ad una famiglia esponenziale in cui la statistica sufficiente è  $\sum_{i=1}^n y_i X_i$ , il parametro canonico è  $\alpha$  e  $K(\alpha) = \sum_{i=1}^n -\log(1 + \exp(X_i \alpha))$ .

Tabella 3.2: Dati infusione cellulare.

Prop. infusione	Risultati per il seguente numero di giorni:				
	1	2	3	4	5
1	5/31	3/28	20/45	24/47	29/35
2	15/77	36/78	43/71	56/71	66/74
3	48/126	68/116	145/171	98/119	114/129
4	29/92	35/52	57/85	38/50	72/77
5	11/53	20/52	20/48	40/55	52/61

Si considera il parametro di interesse  $t(\alpha)$

$$\gamma = \frac{\sum_{j=1}^5 \xi_{j5}}{\sum_{j=1}^5 \xi_{j1}}, \quad (3.10)$$

che è il rapporto di probabilità complessiva di successo del giorno 5 rispetto alla probabilità complessiva del giorno 1. Come distribuzione a priori si è utilizzata la priori di Jeffreys che in questo caso assume la seguente forma

$$\pi(\alpha) \propto \sqrt{\sum_i \frac{\exp(X_i \alpha)}{(1 + \exp(X_i \alpha))^2}}. \quad (3.11)$$

A partire dalla verosimiglianza e dalla priori, si è calcolata la distribuzione a posteriori. Da quest'ultima si sono generati, tramite algoritmo MCMC, utilizzando il Metropolitan-Hastings adattivo (pacchetto *MHadaptive*), 100000 valori di cui 25000 sono stati usati per il *burn-in* e dei rimanenti si è scelto un valore ogni cinque per ridurre la correlazione, pertanto sono stati tenuti 15000  $\alpha_i$ , dove ogni  $\alpha_i$  è un vettore di dimensione 5. Tramite la (2.4) si è calcolata la covarianza, in cui  $\theta$  indica il parametro canonico della famiglia esponenziale, che coincide con  $\alpha$ , e per tale motivo si inseriscono i valori simulati in  $\theta$  e in  $t$  si inseriscono i valori del parametro di interesse  $\gamma$ , calcolato tramite la (3.10). Il valore atteso, la stima frequentista della standard deviation bayesiana e la stima della standard deviation bayesiana, ottenuta come  $\sqrt{\frac{\sum_{i=1}^{15001} (\gamma_i - \bar{\gamma})^2}{15001}}$ , sono riportate nella prima riga della Tabella 3.3. La densità a posteriori del parametro di interesse,  $\gamma_i$ , ottenuti dalla simulazione tramite MCMC è riportata in Figura 3.3a.

Come per l'esempio precedente, anche per questo esempio, si è applicata la versione bootstrap della formula generale.

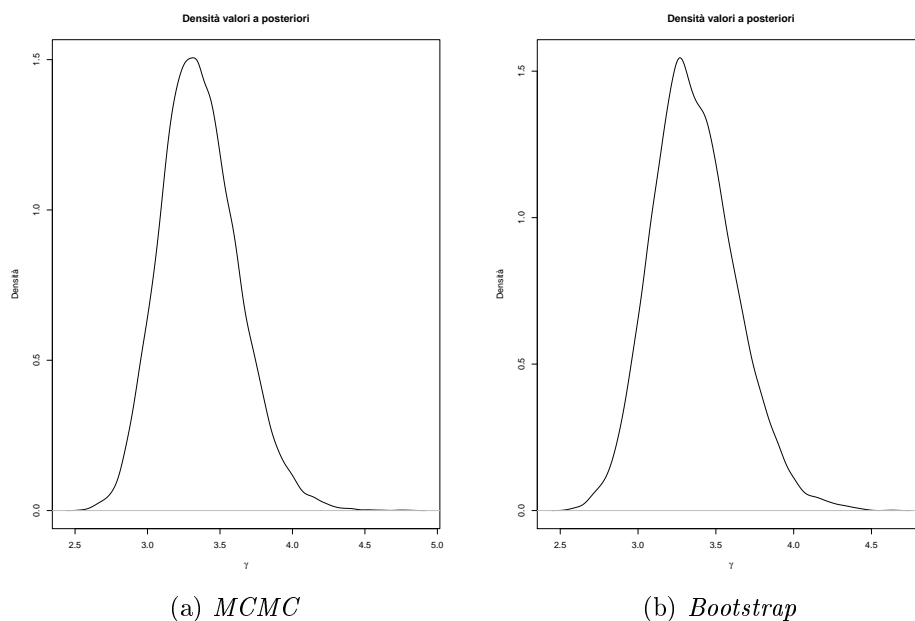


Figura 3.3: Densità a posteriori dei valori simulati.

Si sono generati  $B = 8000$  nuovi campioni  $s_{jk}^*$  da una binomiale di parametri  $n_{jk}$  e  $\hat{\xi}_{jk}$  dove quest'ultima quantità indica la stima di massima verosimiglianza di  $\xi_{jk}$  ottenuta dal modello (3.8). Da questi nuovi campioni  $s_{jk}^*$  si sono ottenute le stime di massima verosimiglianza di  $\alpha_1, \dots, \alpha_{8000}$ . Ottenuti tali valori si sono calcolate le quantità di interesse  $\gamma_i = t(\alpha_i)$  come dalla formula (3.10). I pesi sono stati ottenuti partendo dalla distribuzione a priori di Jeffreys (3.11). Poichè il modello appartiene alla famiglia esponenziale il fattore di conversione  $R_i$  può essere espresso come nella formula (2.14) in cui la *half-deviance difference* della formula (2.15) assume la seguente formula

$$\Delta = \sum_{j,k=1}^5 ((\eta_{jk} - \hat{\eta}_{jk})(\xi_{jk} + \hat{\xi}_{jk}) - 2 \log\{(1 + \exp(\eta_{jk})) / (1 + \exp(\hat{\eta}_{jk}))\}), \quad (3.12)$$

dove  $\eta_{jk} = \log\{\xi_{jk}/(1 - \xi_{jk})\}$ ,  $\hat{\eta}_{jk}$  e  $\hat{\xi}_{jk}$  sono state ottenute a partire dalle stime di massima verosimiglianza di  $\alpha$ . Nella formula (3.12) non è stata utilizzata l'indicizzazione da 1 a  $n$ . Inoltre poichè la distribuzione a priori è quella di Jeffreys, i pesi assumono la forma (2.17), quindi  $p_i = \exp(\Delta(\alpha_i))$ . Ottenuti tali valori per i pesi si sono potute ottenere le stime del valore atteso e delle deviazioni standard sia dal punto di vista frequentista che da quello bayesiano.

Il valore atteso sarà:  $\hat{\beta}_B = \sum_1^B p_i \gamma_i$  mentre la stima della deviazione standard bayesiana si otterrà come  $\{\sum p_i (\gamma_i - \hat{\gamma})^2\}^{0.5}$ . Le stime del valore atteso a posteriori, della deviazione standard frequentista e bayesiana sono riportate nella seconda riga della Tabella 3.3. Nella Figura 3.3b è riportata la densità a posteriori per  $\gamma_i$  ottenuti con quest'ultimo metodo.

Infine si è fatto uno studio di simulazione Monte Carlo. Si sono simulati 1000 valori per ogni  $s_{jk}^*$  partendo da una binomiale con parametri  $n_{jk}$  e  $\hat{\xi}_{jk}$  dove quest'ultima quantità indica la stima di massima verosimiglianza di  $\xi_{jk}$  ottenuta dal modello (3.8). Calcolati tali valori, si è applicato un algoritmo MCMC, Metropolitan-Hastings adattivo, in cui i dati erano una lista che comprendeva la matrice delle  $x$ , gli  $n_{jk}$  e  $s_{jk}^*$ , per simulare i valori del parametro canonico; quindi si sono ottenute le stime della valore atteso a posteriori, della deviazione standard bayesiana e frequentista per la quantità d'interesse (3.10). Ottenute tali stime si è fatta la media aritmetica delle tre quantità sopra citate e i valori sono riportati nella terza riga della Tabella 3.3.

Nella Tabella 3.3 sono riportati i valori per i tre metodi. Come si vede,

Tabella 3.3: Confronto stime ottenute con formule generale, versione bootstrap e con il principio del campionamento ripetuto.

	$\hat{\gamma}$	SD frequentista	SD bayesiana
Formula generale	3.3587	0.2618	0.2627
Bootstrap	3.3540	0.2645	0.2656
PCR	3.3836	0.2677	0.2686

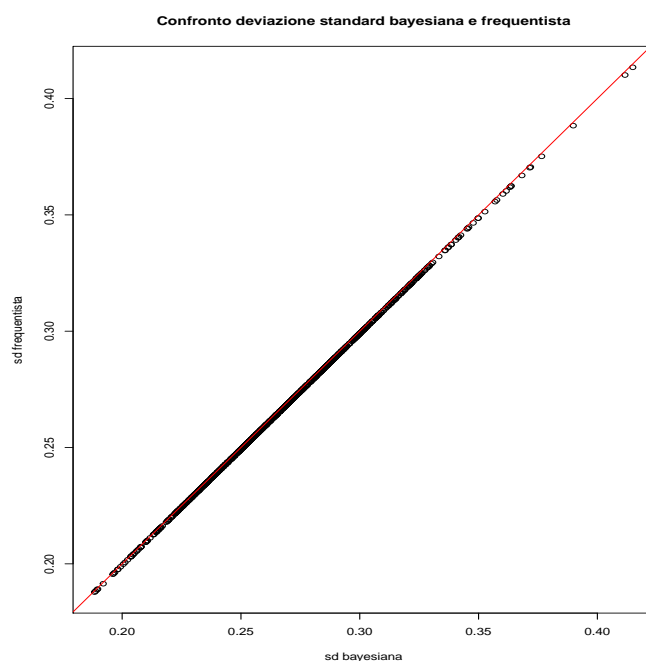
confrontando i risultati, si nota che le stime della deviazione standard sia dal punto di vista frequentista che dal punto di vista bayesiano ottenute con ogni metodo sono molto simili tra di loro.

Analogamente all'esempio precedente, come ultima analisi si è andati a vedere che differenza avessero la stima frequentista e bayesiana della deviazione standard in ogni campione che è stato generato. Pertanto si sono rapportate graficamente le stime delle due quantità mettendo quella bayesiana sull'asse delle ascisse e quella frequentista sull'asse delle ordinate.

Come si vede in Figura 3.4, per valori più piccoli le due deviazioni standard sono praticamente uguali disponendosi lungo la bisettrice, mentre, spostandosi verso destra, all'aumentare dei valori della deviazione standard, la versione frequentista è più piccola, seppur in maniera lieve, della contropar-



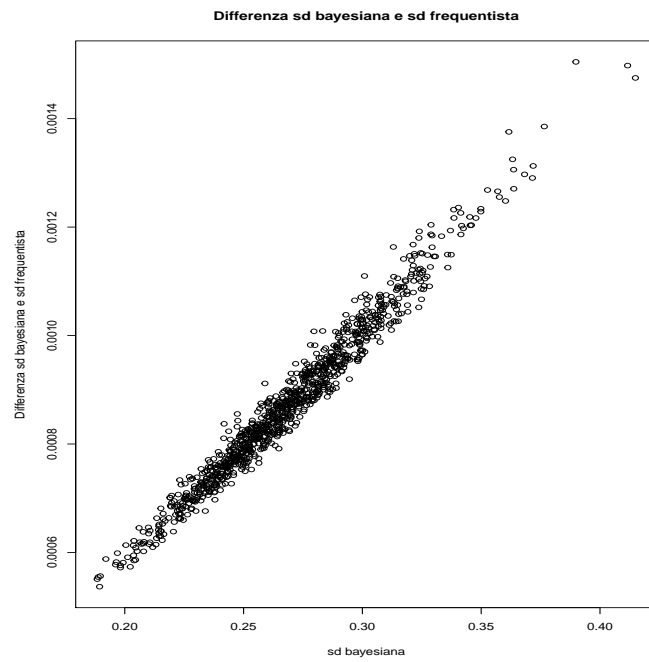
Figura 3.4: Confronto deviazione standard bayesiana e frequentista per i 1000 campioni Monte Carlo.



te bayesiana. Questo andamento si evince anche nella Figura 3.5, in cui sull'asse delle ordinate c'è la differenza fra la deviazione standard bayesiana e frequentista e sull'asse delle ascisse c'è la deviazione standard bayesiana.

Infine si è confrontata la deviazione standard dei 1000 campioni Monte Carlo del valore atteso a posteriori con le medie delle deviazioni standard frequentista e bayesiana. I valori ottenuti sono: 0.2689 per la deviazione standard del valore atteso, 0.2678 la deviazione standard frequentista e 0.2686 quella bayesiana. Si nota che la deviazione standard dei valori attesi è praticamente uguale alla media delle deviazioni standard bayesiane, mentre la media delle deviazioni standard frequentiste risulta essere leggermente inferiore.

Figura 3.5: Differenza tra le stime della standard bayesiana e frequentista per i 1000 campioni Monte Carlo.



# Capitolo 4

## Famiglie esponenziali curve

In questo capitolo si vogliono applicare i metodi presentati nei Paragrafi 2.1 e 2.2; facendo uno studio di simulazione per valutare le differenze tra la deviazione standard frequentista e bayesiana. Inoltre, viene introdotto il modello esponenziale tangente (Reid e Fraser, 2010).

### 4.1 Modello esponenziale tangente

Si consideri un modello di famiglia esponenziale pieno come in (1.7) in cui  $s(y)$  è la statistica sufficiente. Sia  $f(y; \theta)$  che il modello per la distribuzione marginale di  $s(y)$  hanno la stessa funzione di verosimiglianza osservata  $l^0(\theta) = \theta^T s^0 - K(\theta)$  e lo stesso parametro canonico  $\theta$ . Quest'ultimo può essere ottenuto dalla derivata della funzione di verosimiglianza rispetto a qualsiasi punto  $s(y)$ . Si indica con  $s^0$  il valore di  $s(y)$  ottenuto con i dati osservati  $y^0$ .

In modelli più generali è possibile trovare un modello che approssima il modello di famiglia esponenziale, il modello esponenziale tangente, che è un modello di famiglia esponenziale le cui funzioni di verosimiglianza osservata e la derivata prima rispetto ai dati osservati sono le stesse del modello originale. Il modello esponenziale tangente nei dati  $s^0$  è definito a partire dal modello  $f(s(y); \theta)$  come

$$f_{TEM}(t; \theta) dt = \exp\{\varphi(\theta)^T t + l(\theta; s^0)\} h(t) dt \quad (4.1)$$

dove  $t$  può essere vista come una variabile punteggio  $t(y) = l_\theta(\hat{\theta}^0; y^0)$ . Le quantità  $l(\theta; y)$  e  $\varphi(\theta; y^0)$  sono definite a partire dal modello originale

$$l(\theta; y^0) = \log f(y^0; \theta), \quad \varphi(\theta; y^0)^T = \frac{\partial}{\partial V} l(\theta; y^0), \quad (4.2)$$

dove  $V$  è descritto in dettaglio nel seguito (formula (4.3)). Segue da (4.1) che  $\log f_{TEM}(\theta; y^0) = l(\theta; y^0)$  e  $(\partial/\partial t) \log f_{TEM}(\theta; t^0) = \varphi(\theta; y^0)^T = l_V(\theta; y^0)$ . Questo dimostra che il modello esponenziale tangente e il modello originale hanno la stessa funzione di log-verosimiglianza e la stessa derivata rispetto al campione della funzione di log-verosimiglianza.

Il modello esponenziale tangente implementa il condizionamento ad una statistica ancillare per costruire un modello approssimato in  $\mathbb{R}^d$  dal modello originale in  $\mathbb{R}^n$  così da dover essere scritto come  $f_{TEM}(t; \theta|a)$ . Questo condizionamento è implementato attraverso la scelta di  $V$ . Indicando con  $z = (z_1, \dots, z_n)$  un vettore di statistiche pivotali  $z_i = z_i(y_i, \theta)$ , che potrebbe essere semplicemente il vettore delle funzioni di ripartizione  $F(y_i; \theta)$ . Si usa tale vettore di quantità pivotali per definire una matrice  $V$  come

$$V = -\left(\frac{\partial z}{\partial y}\right)^{-1} \left(\frac{\partial z}{\partial \theta}\right) \Bigg|_{\theta=\hat{\theta}, y=y^0} = \frac{dy}{d\theta} \Bigg|_{\theta=\hat{\theta}, y=y^0} \quad (4.3)$$

in cui l'espressione finale della (4.3) è la derivata di  $y$  per una fissata quantità pivotale  $z$ . Il parametro canonico della (4.1) è

$$\varphi(\theta)^T = l_V(\theta; y)|_{y=y^0} = \frac{dl(\theta; y)}{dV} \Bigg|_{y=y^0} = \sum_i \frac{\partial l(\theta; y^0)}{\partial y_i} V_i, \quad (4.4)$$

dove  $V_i$  indica la riga della matrice  $V$ .

Un approccio differente di approssimazione fu proposto da Skovgaard (1996), che ha ottenuto le stime delle direzioni dei condizionamenti. Sia  $I(\theta; \theta_0)$  una funzione media di verosimiglianza:

$$I(\theta; \theta_0) = E_{\theta_0}\{l(\theta; y)\} = \int l(\theta; y) f(y; \theta_0) dy. \quad (4.5)$$

Tale funzione è legata alla distanza di Kullback-Leibler, che è  $I(\theta_0; \theta_0) - I(\theta; \theta_0)$  e viene anche chiamata l'informazione di Fraser, Kent (1982). Una nuova versione di  $\varphi$ , chiamiamola  $\bar{\varphi}$ , del parametro canonico per il modello 4.1 è definito differenziando la funzione  $I(\theta; \hat{\theta})$  invece di  $l(\theta; y)$ :

$$\bar{\varphi}(\theta) = \frac{\partial}{\partial \theta_0} I(\theta; \theta_0) \Bigg|_{\theta_0=\hat{\theta}} = \frac{\partial}{\partial \hat{\theta}} I(\theta; \hat{\theta}). \quad (4.6)$$

Il modello di famiglia esponenziale con parametro canonico  $\bar{\varphi}(\theta)$  è un'approssimazione  $O(n^{-1})$  del modello esponenziale tangente (4.1). Facendo la media

della log-verosimiglianza nei calcoli di  $\bar{\varphi}$ , da un lato, elimina la dipendenza dalla statistica ancillare approssimata riducendo l'accuratezza delle approssimazioni dell'area delle code, dall'altro, per molti modelli il calcolo di  $\bar{\varphi}$  è più semplice del calcolo di  $\varphi$  usando la formula (4.4). Sia  $\varphi$  che  $\bar{\varphi}$  sono state ottenute da Reid e Fraser (2010).

## 4.2 Modello parabola normale

In questo paragrafo si prende in considerazione la parabola normale. In particolare, sia  $y = (y_1, \dots, y_n)$  realizzazione di  $n$  variabili casuali indipendenti  $Y_i \sim N(\theta, \theta)$  con  $\theta > 0$  e funzione di densità

$$f(y; \theta) \propto \exp\left\{-\frac{1}{2\theta^2} \sum_{i=1}^n y_i^2 + \frac{1}{\theta} \sum_{i=1}^n y_i - n \log \theta\right\}. \quad (4.7)$$

Si noti che il vettore della statistica sufficiente  $s(y) = (\sum y_i, \sum y_i^2)$  ha dimensione due mentre la dimensione del parametro è uno, pertanto tale modello è una famiglia esponenziale curva. Per tale motivo, per poter utilizzare le formule nei Paragrafi 2.1 e 2.2, si è utilizzato il modello esponenziale tangente. I dati utilizzati sono stati creati generando 200 valori dal modello con  $\theta = 5$ .

Per poter generare un campione a posteriori di  $\theta$  tramite algoritmo MCMC, utilizzando un Metropolitan-Hastings con generazione da una normale con media e varianza calcolati a partire dalla distribuzione a posteriori, si è usata come priori una distribuzione gamma

$$\pi(\theta) = \frac{\lambda^\gamma}{\Gamma(\gamma)} \theta^{\gamma-1} \exp(-\lambda\theta) \quad (4.8)$$

con  $\lambda$  e  $\gamma$  pari a 10. Applicando le Formule (4.3) e (4.4) si è ottenuto il parametro canonico

$$\varphi(\theta) = -\frac{1}{\theta^2 \hat{\theta}} \sum y_i^2 + \frac{1}{\theta \hat{\theta}} \sum y_i \quad (4.9)$$

dove  $\hat{\theta}$  indica la stima di massima verosimiglianza calcolata a partire dal campione osservato. Ottenute le stime a posteriori di  $\theta$  si sono inserite nella formula (4.9) e si sono ottenuti i  $\varphi$ . A partire dai valori di  $\theta$  e di  $\varphi$  si sono ottenute le stime della covarianza (2.4), in cui il parametro di interesse  $t$  è  $\theta$  parametro del modello (4.7) mentre i  $\theta$  sono i  $\varphi$  parametro canonico. La stima della devianzione standard è stata ottenuta con l'espressione del Teorema 3. I valori ottenuti sono riportati nella prima riga della Tabella 4.1.

Analogo procedimento si è applicato con il bootstrap. Si sono generati 8000 campioni da una normale con media e varianza pari alla stima di massima verosimiglianza,  $\hat{\theta}$ . Ottenuti tali campioni, tramite verosimiglianza si sono ottenute le nuove stime di massima verosimiglianza di  $\theta$ . A partire dai  $\theta_i$  si sono ottenuti i  $\varphi_i$  attraverso la formula (4.9).

Per calcolare i pesi che trasformano la verosimiglianza in un campione a posteriori si è utilizzata la formula (2.10). Nel nostro esempio, tuttavia, il parametro di interesse è  $\theta$  mentre il parametro canonico, tramite l'utilizzo del modello esponenziale tangente, è  $\varphi$  pertanto i pesi andranno applicati a quest'ultimo parametro e non a quello di partenza. Nella formula (2.10)  $\pi_\varphi$  sarà il valore della priori calcolata in  $\theta = \theta(\varphi)$  a cui verrà moltiplicato lo Jacobiano dato che è stata applicata la trasformazione  $\varphi = \varphi(\theta)$ . Gli  $R_i$ , invece, poichè nel rapporto tra la verosimiglianza e la densità bootstrap,  $\varphi = \varphi(\theta)$  è una riparametrizzazione, non c'è bisogno di nessuna quantità correttiva. I valori sono riportati nella seconda riga della Tabella 4.1.

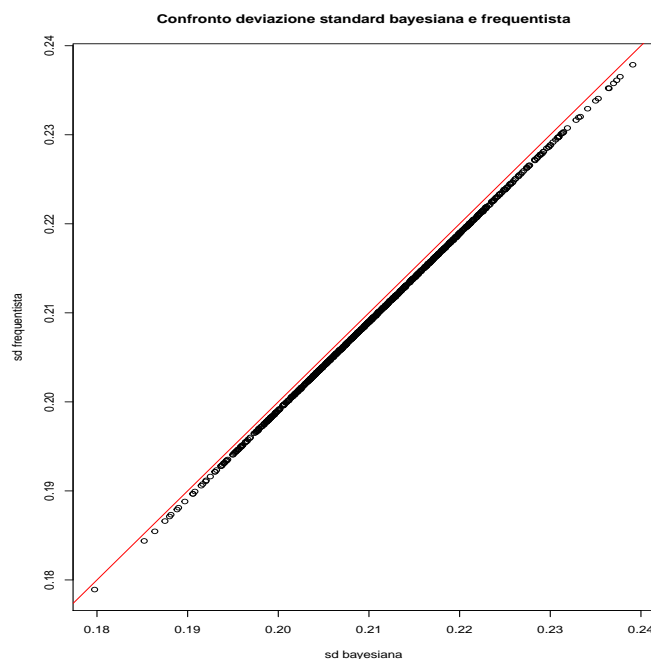
Infine si è fatto uno studio di simulazione Monte Carlo. Si sono simulati 1000 vettori  $y$  partendo da una normale con media e varianza par alla stima di massima verosimiglianza, quindi, per ogni vettore  $y$ , si è applicato l'algoritmo MCMC e si sono ottenute, per la quantità d'interesse  $\theta$  con parametro canonico  $\varphi$ , le stime del valore atteso, della deviazione standard bayesiana e frequentista. Ottenute tali stime si è fatta la media aritmetica per tutte e tre le quantità. I valori sono riportati nella terza riga della Tabella 4.1.

Tabella 4.1: Confronto stime ottenute con formula generale, versione bootstrap e con il principio del campionamento ripetuto.

	$\hat{\theta}$	SD frequentista	SD bayesiana
Formula generale	5.0712	0.2131	0.2140
Bootstrap	4.9854	0.1950	0.1990
PCR	5.0429	0.2099	0.2109

Come ultima analisi si è voluto andare a vedere che differenza avessero la stima frequentista e bayesiana della deviazione standard in ogni campione che è stato generato. Pertanto si sono rapportate graficamente, Figura 4.1, le stime delle due quantità mettendo quella bayesiana sull'asse delle ascisse e quella frequentista sull'asse delle ordinate.

Figura 4.1: Confronto deviazione standard bayesiana e frequentista.



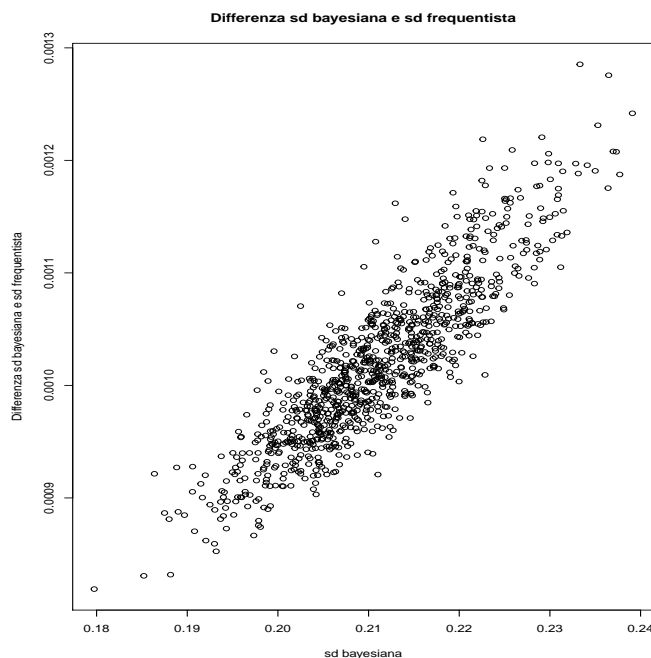
Come si vede in Figura 4.1, la stima della deviazione standard bayesiana sembra sottostimare, in maniera lieve ma sistematica, quella frequentista; infatti i valori nel grafico sono disposti appena sotto la bisettrice. Questo andamento si evince anche dalla Figura 4.2 in cui la differenza, anche se lieve, tra la deviazione standard bayesiana e frequentista sembra avere un trend lineare crescente.

Infine si è confrontata la deviazione standard dei 1000 campioni Monte Carlo del valore atteso a posteriori con le medie delle deviazioni standard frequentista e bayesiana. I valori ottenuti sono: 0.2071 per la deviazione standard del valore atteso, 0.2099 la deviazione standard frequentista e 0.2109 quella bayesiana.

### 4.3 Modello binomiale con legame probit

In questo paragrafo vengono riutilizzati i dati presentati nel Paragrafo 3.2 relativi all'iniezione di nuclei di topo in cellule muscolari umane, Tabella 3.2. Come per il Paragrafo 3.2, si è ipotizzato un modello binomiale in cui  $(s_{jk}, n_{jk})$  sono il numero dei successi e il numero delle colonie della  $jk$ -sima

Figura 4.2: Differenza tra le stime della standard bayesiana e frequentista per i 1000 campioni Monte Carlo.



prova

$$s_{jk} \sim Bi(n_{jk}, \xi_{jk}) \quad j, k = 1, \dots, 5. \quad (4.10)$$

La funzione legame che viene utilizzata in questo esempio è la probit

$$\Phi^{-1}(\xi_{jk}) = \beta_0 + \beta_1 I_j + \beta_2 I_j^2 + \beta_3 D_k + \beta_4 D_k^2 \quad (4.11)$$

in cui  $\Phi^{-1}(\cdot)$  indica la funzione quantile della distribuzione normale e la probabilità del modello (4.10) sarà pertanto  $\xi_i = \Phi(X_i \beta)$ , dove  $X_i$  è l' $i$ -esima riga di  $X$  e  $\xi_i$  è il vettore che mette sequenzialmente i  $\xi_{jk}$  uno sopra l'altro.

Il modello (4.10) con la funzione legame (4.11) non appartiene alle famiglie esponenziali piene ma è una famiglia esponenziale curva. Il parametro di interesse, come per l'esempio nel Paragrafo 3.2, sarà

$$\gamma = \frac{\sum_{j=1}^5 \xi_{j5}}{\sum_{j=1}^5 \xi_{j1}} \quad (4.12)$$

che è il rapporto di probabilità complessiva di successo del giorno 5 rispetto alla probabilità complessiva del giorno 1. Come distribuzione a priori



per  $\beta$  si è scelta una normale multivariata  $\mathcal{N}_5(\mu, \Sigma)$  con  $\mu$  un vettore nullo di dimensione 5 e  $\Sigma$  matrice diagonale con il valore 10 sulla diagonale principale.

A partire dalla verosimiglianza e dalla distribuzione a priori si è calcolata la distribuzione a posteriori. Da quest'ultima si sono generati, tramite MCMC, utilizzando un Metropolitan-Hastings adattivo (pacchetto *MHadaptive*), 100000 valori di cui 25000 sono stati utilizzati per il *burn-in* e dei rimanenti si è scelto un valore ogni cinque per ridurre la correlazione, pertanto sono stati ottenuti 15001  $\beta_i$ , dove ogni  $\beta_i$  ha dimensione cinque.

Il modello (4.10) con funzione legame (4.11) non appartiene ad una famiglia esponenziale piena pertanto per poter applicare le formule viste nei Paragrafi 2.1 e 2.2 ci sono due possibili soluzioni: approssimare il modello (4.10) con il modello esponenziale tangente e calcolare la quantità  $\varphi$  così da poter utilizzare le formule semplificate per una famiglia esponenziale piena, oppure rimanere con il modello (4.10) con funzione legame (4.11) e utilizzare la formula (2.1) derivando la densità per la statistica sufficiente. Per valutare la differenza tra le due strade proposte si eseguiranno entrambe e si metteranno a confronto i risultati. Indipendentemente dal metodo la quantità di interesse  $\gamma$  rimane la medesima, ossia la (4.12).

Per il modello esponenziale tangente, dopo aver utilizzato le formule (4.5) e (4.6) si ottiene la seguente formulazione per  $\bar{\varphi}(\beta)$ :

$$\bar{\varphi}(\beta) = \sum_{i=1}^n \log \frac{\Phi(X_i\beta)}{1 - \Phi(X_i\beta)} \phi(X_i\hat{\beta}) X_i^T \quad (4.13)$$

dove  $\hat{\beta}$  indica la stima di massima verosimiglianza per beta calcolata a partire dal modello (4.10).

I valori di  $\varphi_i$ , ottenuti inserendo i  $\beta_i$  simulati nella formula (4.13), sono stati utilizzati nella formula (2.4) per ottenere la stima della covarianza dove il parametro di interesse  $t$  e il parametro canonico  $\alpha$  sono rispettivamente  $\gamma$  nella (4.12) e  $\varphi$  nella (4.13). La stima della varianza è stata ottenuta invertendo la matrice di varianza dei valori di  $\varphi$ . Quest'ultime, covarianza e varianza, sono state inserite nella formula (2.3) e ci si è procurati la deviazione standard frequentista della stima a posteriori bayesiana.

Gli  $\alpha_y$  definiti nella (2.1), invece, hanno la seguente formulazione:

$$\alpha_y = \log \frac{\Phi(X\beta)}{1 - \Phi(X\beta)}. \quad (4.14)$$

Tali valori vengono inseriti nella formula della covarianza al posto degli  $\alpha$ ; mentre, per calcolare la stima della deviazione standard in un'ottica frequentista, la stima della varianza è stata ottenuta dall'inversa della matrice di varianza degli  $\alpha_y$ .

La deviazione standard bayesiana e frequentista per entrambi i metodi sono riportate nelle prime due righe della Tabella 4.2. La stima bayesiana e frequentista della deviazione standard per entrambi i metodi risulta essere soddisfacente, nel senso che la stima frequentista è simile a quella bayesiana. In particolare, si vede che per  $\alpha_y$  la stima frequentista è praticamente uguale a quella bayesiana e sono rispettivamente 0.2674 e 0.2675; invece per  $\varphi$  le due stime differiscono leggermente, 0.2667 quella frequentista e 0.2675 quella bayesiana. Tale discrepanza è dovuta probabilmente al fatto che si è approssimato la famiglia esponenziale curva con il modello esponenziale tangente attraverso la formula (4.13). Di conseguenza la differenza si vede nelle stime frequentiste della deviazione standard calcolate sia per  $\alpha_y$  che per  $\varphi$ .

Come per il metodo MCMC anche per il bootstrap si sono seguite due strade diverse per ottenere le due stime della deviazione standard: la prima utilizza il modello esponenziale tangente e la seconda utilizza la formula (2.1).

Indipendentemente dal metodo scelto, si sono generati 8000 campioni a partire da una binomiale  $s_{jk}^* \sim Bi(n_{jk}, \hat{\xi}_{jk})$  dove  $\hat{\xi}$  indica la stima di massima verosimiglianza a partire dal modello (4.10) con funzione legame (4.11). Dai nuovi campioni  $s_{jk}^*$  si sono calcolate le stime di massima verosimiglianza di  $\beta$ . A partire da tali stime si è proceduto con le due strade, dalla formula (4.13) si sono calcolati i  $\varphi$ , mentre dalla formula (4.14) si sono calcolati gli  $\alpha_y$ . Nel momento in cui vengono applicati i pesi (2.10) nella parametrizzazione  $\varphi$ , questi si devono calcolare per il parametro canonico del modello esponenziale tangente e non per i  $\beta$ , pertanto il valore della priori andrà moltiplicato per lo Jacobiano della trasformazione  $\beta(\bar{\varphi})$ , in quanto quest'ultima nella distribuzione a priori è una trasformazione della variabile casuale  $\beta$  e non una riparametrizzazione come nel calcolo del fattore di conversione. Di conseguenza si sono derivate le stime delle deviazioni standard sia bayesiana che dal punto di vista frequentista sia per il primo metodo che per il secondo. I valori sono stati riportati nella terza e quarta riga della Tabella 4.2.

Confrontando i risultati ottenuti per la stima frequentista e bayesiana della deviazione standard si hanno delle differenze tra le due quantità. Con il metodo che utilizzava gli  $\alpha_y$  la differenza è praticamente nulla infatti si ha 0.2658 per la stima frequentista e 0.2661 per la stima bayesiana, mentre

con i  $\varphi$  la discrepanza, anche se leggermente maggiore rispetto alla precedente, risulta essere minima: 0.2603 frequentista contro 0.2634 bayesiana. Con il bootstrap ci sono delle differenze, anche se molto piccole, sia nella stima frequentista che in quella bayesiana della deviazione standard ottenute con le due possibili formulazioni di  $\alpha$  (2.1). Tale disparità è presente in quanto nella formula per calcolare la deviazione standard bayesiana entrano in gioco i pesi (2.10) che in  $\alpha_y$  sono ottenuti direttamente inserendo i valori simulati nella distribuzione a priori, mentre in  $\varphi$ , essendo una trasformazione, sono ottenuti moltiplicando il valore della distribuzione a priori per lo Jacobiano della trasformata. Tuttavia, nonostante queste differenze, i quattro valori ottenuti sono molto simili tra di loro.

Infine si è fatto uno studio di simulazione Monte Carlo. Si sono simulati 1000 valori per ogni  $s_{jk}^*$  partendo da una binomiale con parametri  $n_{jk}$  e  $\xi_{jk}$  dove quest'ultima quantità indica la stima di massima verosimiglianza di  $\xi_{jk}$  ottenuta dal modello (4.10). Calcolati tali valori, si è applicato un algoritmo MCMC, Metropolitan-Hastings adattivo, in cui i nostri dati erano una lista che comprendeva la matrice  $X$ , gli  $n_{jk}$  e  $s_{jk}^*$  per simulare i valori del nostro parametro  $\beta$ . Anche per questa procedura si sono calcolati i  $\varphi$  e gli  $\alpha_y$  a partire dai valori simulati quindi si sono ottenute le stime della deviazione standard. Quest'ultime sono riportate nelle ultime due righe della Tabella 4.2.

Anche con questo metodo si sono confrontate le stime ottenute. Utilizzando gli  $\alpha_y$  si vede che la differenza è praticamente nulla: 0.2719 la frequentista rispetto a quella bayesiana di 0.2720; mentre utilizzando  $\varphi$  la differenza è leggermente maggiore: 0.2711 per la deviazione standard frequentista e 0.2720 per quella bayesiana. Si vede quindi come i valori siano comunque molto simili tra di loro.

Tabella 4.2: Confronto stime ottenute con formula generale, versione bootstrap e con il principio del campionamento ripetuto.

	$\hat{\gamma}$	SD frequentista	SD bayesiana
Formula generale $\varphi$	3.3484	0.2666	0.2675
Formula generale $\alpha_y$	3.3484	0.2674	0.2675
Bootstrap $\varphi$	3.3442	0.2603	0.2634
Bootstrap $\alpha_y$	3.3534	0.2658	0.2661
PCR $\varphi$	3.3728	0.2711	0.2720
PCR $\alpha_y$	3.3728	0.2719	0.2720

Come ultima analisi si è voluto andare a vedere che differenza avessero la stima frequentista e bayesiana della deviazione standard in ogni campione che è stato generato per tutte e due le quantità  $\varphi$  e per  $\alpha_y$ . Pertanto, per ognuna delle due formulazioni di  $\alpha$  della formula (2.4), si sono rapportate graficamente le stime delle due quantità mettendo quella bayesiana sull'asse delle ascisse e quella frequentista sull'asse delle ordinate.

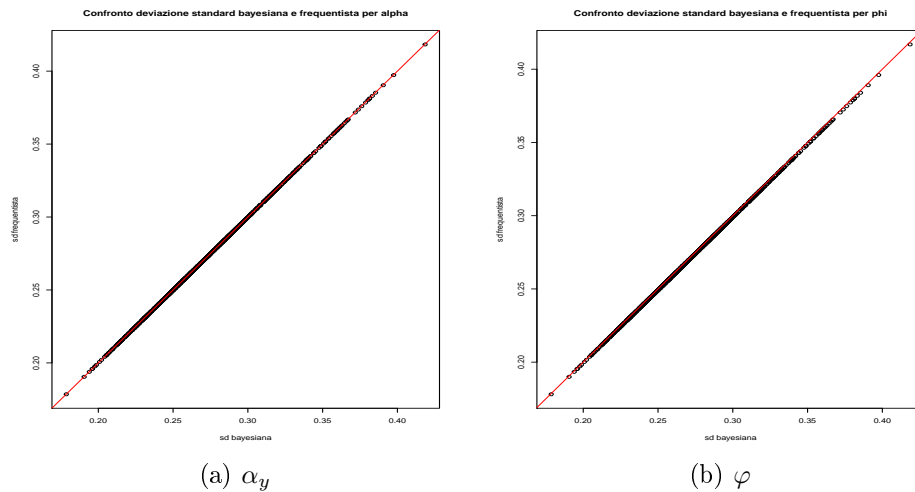


Figura 4.3: Confronto deviazione standard frequentista e bayesiana per i 1000 campioni Monte Carlo.

Nella Figura 4.3a vediamo che per tutti i campioni generati la deviazione standard frequentista è praticamente uguale a quella bayesiana, infatti i valori si dispongono lungo la bisettrice del quadrante. Infatti, come si vede in Figura 4.4a, le differenze tra la deviazione standard bayesiana e frequentista sono minime anche se si vede un trend lineare crescente all'aumentare della deviazione standard bayesiana. Per  $\varphi$ , Figura 4.3b, anche se i valori sono quasi simili, la deviazione frequentista risulta essere costantemente un poco più piccola di quella bayesiana e pertanto i valori si dispongono appena sotto la bisettrice del quadrante. Questo si evince anche dalla Figura 4.4b in cui le differenze, all'aumentare della deviazione standard bayesiana, tra le due deviazioni standard sono più marcate rispetto a quelle ottenute con  $\alpha_y$ . Confrontando, infine, la deviazione standard delle stime del valore atteso a posteriori con le medie delle deviazioni standard frequentista e bayesiana, si nota che sono simili tra loro sia per  $\varphi$  che per  $\alpha_y$ . Infatti si sono ottenute per le tre quantità sopra citate rispettivamente: 0.2809, 0.2719 e 0.2721 per  $\alpha_y$  e 0.2809, 0.2711 e 0.2720 per  $\varphi$ .

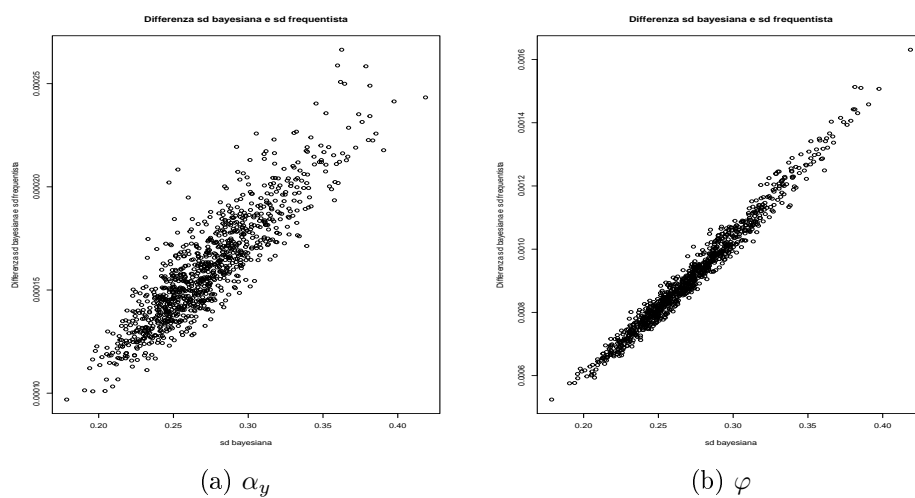


Figura 4.4: Differenza deviazione standard frequentista e bayesiana per i 1000 campioni Monte Carlo.

Avendo utilizzato due differenti quantità,  $\alpha_y$  e  $\varphi$ , per ottenere le stime della deviazione standard, si è voluto confrontare le due quantità prima rapportando la deviazione standard frequentista, Figura 4.5a, e poi quella bayesiana, Figura 4.5b, inserendo nelle ordinate e nelle ascisse rispettivamente  $\alpha_y$  e  $\varphi$ .

Dalla Figura 4.5a si nota che i valori della deviazione standard frequentista calcolati per  $\alpha_y$  e per  $\varphi$  sono praticamente uguali in quanto si dispongono sulla bisettrice. Analogo discorso per la deviazione standard bayesiana.

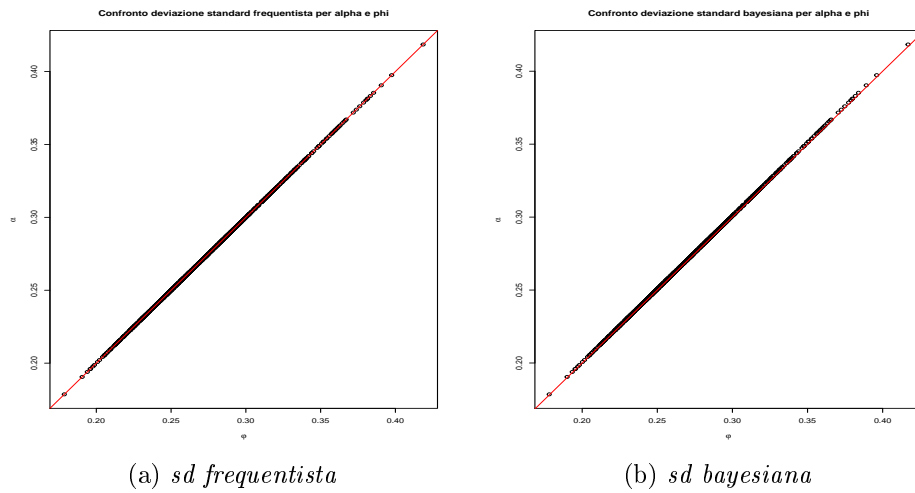


Figura 4.5: Confronto deviazione standard frequentista e bayesiana rispetto  $\alpha_y$  e  $\varphi$  per i 1000 campioni Monte Carlo.

# Conclusioni

Nello sviluppo della tesi si è voluto estendere il lavoro di Efron (2015) a famiglie esponenziali curve. In particolare si voleva valutare l'accuratezza della stima a posteriori bayesiana da un punto di vista frequentista applicando il metodo a famiglie esponenziali curve e non solo a quelle piene.

Si sono applicati due differenti metodi: il primo utilizza i modelli esponenziali tangenti per ricondurre la famiglia esponenziale curva ad una piena e poter utilizzare la versione semplificata. Il secondo, invece, calcola le quantità, che ci permettono di ottenere l'accuratezza frequentista, direttamente dal modello di partenza.

Nonostante si siano considerati solo due esempi molto semplici, si è visto nel Capitolo 4 che i due metodi, utilizzati nel Paragrafo 4.3, sono efficaci e portano a stime molto simili sia tra di loro che a quella bayesiana. Ulteriori approfondimenti e studi di simulazione su larga scala, come per esempio con diverse numerosità campionarie, sono necessari per poter eventualmente convalidare e generalizzare queste prime indicazioni.

Un ulteriore possibile sviluppo futuro è applicare le metodologie presenti nel Capitolo 2 a modelli che non appartengono a famiglie esponenziali.





# Appendice A

## Funzione accuratezza frequentista

```
1 f_accuracy <- function(tt, aa, pp=rep(1,B), V) {
2   #tt -> quantita' d'interesse t(theta)
3   #aa -> parametro canonico o alpha_y
4   #pp -> pesi per il bootstrap per mcmc sara' un vettore
       di uni
5   # V -> matrice di varianza calcolabile a partire da aa
6   if (missing(V)) V <- solve(var(aa))
7   B <- length(tt)
8
9   # i pesi
10  pp <- pp / sum(pp)
11
12  # stima a posteriori bayesiana del valore atteso
13  Ebayes <- sum(pp * tt)
14
15  abar <- as.vector(pp %*% aa)
16  ttt <- tt - Ebayes
17  aaa <- t(t(aa) - abar)
18
19  #deviazione standard bayesiana
20  sdbayes <- sum(pp * ttt^2)^.5
21
22  # covarianza
23  covhat <- ttt %*% (pp * aaa)
24
25  #deviazione standard frequentista
26  sdfreq <- sqrt(covhat %*% V %*% t(covhat))
27
28  if (var(pp) > 0) {# calcolo coefficiente di variazione
```

```
      interna
29     qq <- tt * pp
30     ss <- qq / mean(qq) - pp / mean(pp)
31     cv <- sqrt(sum(ss^2)) / B
32     v <- c(Ebayes, sdfreq, cv, sdbayes)
33     names(v) <- c("Ebayes", "sdfreq", "cv", "sdbayes")
34     return(v)
35   }
36
37   v <- c(Ebayes, sdfreq, sdbayes)
38   names(v) <- c("Ebayes", "sdfreq", "sdbayes")
39   v
40 }
```

# Appendice B

## Codici R per il capitolo 3

### B.1 Modello esponenziale normale

```
1 rm(list=ls())
2 source('f_accuracy.R')
3
4 diabetes<- read.table(file = "diabetes.dat",header = T)
5
6 names(diabetes)
7
8 # standardizzo
9 diabetes2 <- data.frame(scale(diabetes))
10
11 diab<- list(y=diabetes2$Y,X=model.matrix(Y~.-1,data=
12     diabetes2))
13 # log-verosimiglianza
14 llik<- function(alpha,data){
15     mu<- drop(data$X%%alpha)
16     -sum(dnorm(x = data$y,mean=mu,sd=1,log=T))
17 }
18
19 mle<- nlminb(start = rep(0,10),function(x) llik(x,data=
20     diab))
21 mle$par
22
23 # log-priori
24 lprior<- function(alpha, lambda){
25     -lambda*sum(abs(alpha))
```

```

26 }
27
28 # log-posteriori
29 lprior <- function(alpha, data, lambda){
30   lprior(alpha=alpha, lambda=lambda) - llik(alpha=alpha, data
      =data)
31 }
32
33 library(MHadaptive)
34
35 sim <- Metro_Hastings(lprior, pars=rep(0,10), data=diab
      , lambda=0.37,
36                               iterations=100000, burn_in = 25000)
37
38 sim <- mcmc_thin(sim, thin=5) # un po' di thinning (forse
      non necessario)
39 N <- nrow(sim$trace)
40 par(mfrow=c(2,5))
41 for (i in 1:ncol(sim$trace))
42 {
43   plot(sim$trace[,i], type="l")
44   lines(1:N, cumsum(sim$trace[,i])/(1:N), col=2)
45 }
46 for (i in 1:ncol(sim$trace))
47   acf(sim$trace[,i])
48 for (i in 1:ncol(sim$trace))
49   plot(density((sim$trace[,i])))
50
51 # grafico densita' valori a posteriori
52 plot(density(drop(sim$trace%*%diab$X[125,])), ylab = '
      Densita', xlab = expression(gamma), main = 'Densita_
      valori_a_posteriori')
53 curve(dnorm(x, general_accuracy[1], general_accuracy[3]),
      col=2, add=T)
54
55 plot(density(t_alpha), ylab='Densita', xlab=expression(
      gamma), main = 'Densita_valori_a_posteriori')
56 curve(dnorm(x, bootstrap_accuracy[1], bootstrap_accuracy
      [4]), col=2, add=T)
57
58
59 # parametro canonico della famiglia esponenziale
60 alpha_y <- sim$trace

```

```
61
62 # alpha_y come da derivata in funzione di y
63 alpha_y<- diab$y[125]-drop(sim$trace%%diab$X[125,])
64
65 general_accuracy<- f_accuracy(tt = drop(sim$trace%%diab$
      X[125,]),pp = rep(1,15001),aa = alpha_y)
66
67 # versione con il bootstrap
68 s<- matrix(NA,ncol=442,nrow = 8000)
69 for(i in 1:8000){
70     out<- rnorm(442,mean=diab$X%%mle$par)
71     s[i,]<- out
72 }
73
74 mle_alpha<- apply(s,1,function(c) nlminb(rep(0,10),
      objective = function(e) llik(e,data=list(y=c,X=diab$X)
      ))$par)
75
76 # quantita' di interesse
77 t_alpha<- drop(diab$X[125,]%%mle_alpha)
78
79 # fattore di conversione e priori
80 r<- lprior_value <- rep(NA,8000)
81 for(j in 1:8000){
82 r[j]<- llik(alpha=mle_alpha[,j],data=list(y=diab$y,X=diab
      $X))/llik(alpha=mle$par,data=list(y=s[j,],X=diab$X))
83 lprior_value[j] <- lprior(alpha=mle_alpha[,j],lambda =
      0.37)
84 }
85
86 # pesi
87 weight<- r*lprior_value
88
89 bootstrap_accuracy<- f_accuracy(tt = t_alpha,aa = t_alpha
      ,pp = weight)
90
91 # principio campionamento ripetuto con approccio bayesiano
92 nsim<- 1000
93 dat<- matrix(NA,ncol=nsim,nrow=442,byrow = F)
94 for (i in 1:nsim) dat[,i]<- rnorm(442,diab$X%%mle$par)
95
96 library(snowfall)
97 library(parallel)
```

```

98 ncpu<- detectCores()-1
99 sfInit(parallel = T,cpus = ncpu)
100 sfLibrary(MHadaptive)
101 sfExport(list=c('nsim','lposterior','diab','mle','dat','
    lprior','llik'))
102 system.time(cp_t_alpha<- sfApply(dat,2,function(d) drop(
    mcmc_thin(mcmc_object = Metro_Hastings(lprior,pars
    =rep(0,10),
103
    data=list(X=diab$X,y=d),
    lambda=0.37,iterations=10^5,
104
    burn_in = 25000),thin = 5)$
    trace%%diab$X[125,])))
105 sfStop()
106 pcr_mean<- mean(apply(cp_t_alpha,2,mean))
107 pcr_sd<- mean(apply(cp_t_alpha,2,sd))
108
109 pcr_mean
110 pcr_sd
111 plot(density(cp_t_alpha))
112 gen_accuracy_cp<- apply(cp_t_alpha,2,function(e) f_
    accuracy(tt = e,aa = diab$y[125]-e,pp = rep(1,15001)))
113 accuracy_cpr <- apply(gen_accuracy_cp,1,mean)
114
115 par(mfrow=c(1,1))
116 plot(x = gen_accuracy_cp[3,],y = gen_accuracy_cp[2,],xlab
    = 'sd_bayesiana',ylab = 'sd_frequentista',
117 main='Confronto deviazione standard bayesiana e
    frequentista')
118 abline(a = 0,b = 1,col=2)
119
120 general_accuracy
121 bootstrap_accuracy
122 accuracy_cpr
123
124 #confronto valori Monte Carlo
125 c(sd(gen_accuracy_cp[1,]),mean(gen_accuracy_cp[2,]),mean(
    gen_accuracy_cp[3,]))

```

## B.2 Modello esponenziale binomiale

```
1 rm(list = ls())
```

```

2 source('f_accuracy.R')
3 # creazione del dataset
4 muscle <- read.csv('muscle.csv', header = T, sep = ',')
5 muscle$d <- c(rep(1, 5), rep(2, 5), rep(3, 5), rep(4, 5),
6             rep(5, 5)) # giorno
7
8 muscle2 <-
9   list(Y = muscle$s,
10        n = muscle$n,
11        X = model.matrix( ~ muscle$i + I(muscle$i ^ 2) +
12                          muscle$d + I(muscle$d ^
13                                    2)))
14 #log-verosimiglianza
15 llik.muscle <- function(alpha, data) {
16   eta <- drop(data$X %*% alpha)
17   csi <- plogis(eta)
18   sum(dbinom(
19     x = data$Y,
20     size = data$n,
21     prob = csi,
22     log = T
23   ))
24 }
25
26 # stima di massima verosimiglianza
27 mle.muscle <-
28   nlminb(rep(0.5, 5), function(x)
29     - llik.muscle(alpha = x, data = muscle2))
30
31 # log-priori
32 lprior <- function(alpha, data) {
33   log(sqrt(sum(exp(data$X %*% alpha) / (1 + exp(
34     drop(data$X %*% alpha)))) ^ 2)))
35 }
36
37 # log-posteriori
38 lposterior <- function(alpha, data) {
39   lprior(alpha = alpha, data = data) + llik.muscle(alpha
40     = alpha, data = data)

```

```
41
42 # MCMC
43 library(MHadaptive)
44 out <- Metro_Hastings(
45   lposterior,
46   pars = rep(0, 5),
47   data = muscle2,
48   iterations = 100000,
49   burn_in = 25000
50 )
51
52 out <- mcmc_thin(out, thin = 5)
53 N <- nrow(out$trace)
54 par(mfrow = c(1, 5))
55 for (i in 1:ncol(out$trace))
56 {
57   plot(out$trace[, i], type = "l")
58   lines(1:N, cumsum(out$trace[, i]) / (1:N), col = 2)
59 }
60 for (i in 1:ncol(out$trace))
61   acf(out$trace[, i])
62 for (i in 1:ncol(out$trace))
63   plot(density((out$trace[, i])))
64
65 csi_5_mcmc <-
66   apply(out$trace, 1, function(g)
67     sum(plogis(drop(muscle2$X[muscle2$X[, 4] == '5', ] %*
68       % g))))
69 csi_1_mcmc <-
70   apply(out$trace, 1, function(g)
71     sum(plogis(drop(muscle2$X[muscle2$X[, 4] == '1', ] %*
72       % g))))
73
74
75 #quantita' d'interesse
76 t_alpha_mcmc <- csi_5_mcmc / csi_1_mcmc
77
78 general_accuracy <-
79   f_accuracy(tt = t_alpha_mcmc,
80             aa = out$trace,
81             pp = rep(1, 15001))
82 general_accuracy
83
84 # metodo bootstrap
```



```
82
83 # genero da binomiale
84 s <- matrix(NA, ncol = 25, nrow = 8000)
85 for (i in 1:8000) {
86   for (j in 1:25) {
87     out <-
88       rbinom(1, size = muscle2$n[j], prob = plogis(drop(
89         muscle2$X[j, ] %*% mle.muscle$par)))
90     s[i, j] <- out
91   }
92 }
93 # stime massima verosimiglianza di alpha
94 mle2 <- apply(s, 1, function(c)
95   nlminb(
96     start = rep(0, 5),
97     objective = function(z)
98       - llik.muscle(
99         alpha = z,
100        data = list(Y = c, n = muscle2$n, X = muscle2$X)
101      )
102   )$par)
103
104 csi_5 <-
105   apply(mle2, 2, function(g)
106     sum(plogis(drop(muscle2$X[muscle2$X[, 4] == '5', ] %*%
107       % g))))
108 csi_1 <-
109   apply(mle2, 2, function(g)
110     sum(plogis(drop(muscle2$X[muscle2$X[, 4] == '1', ] %*%
111       % g))))
112
113 # quantita' interesse
114 t_alpha <- csi_5 / csi_1
115
116 # quantita' per calcolare i pesi
117 eta <- apply(mle2, 2, function(d)
118   plogis(drop(muscle2$X %*% d)))
119 ni <- qlogis(eta)
120
121 #half-deviance difference
122 delta <- apply((ni - drop(muscle2$X %*% mle.muscle$par))
123   *
```

```

121             (eta + plogis(drop(
122                 muscle2$X %*% mle.muscle$par
123             ))) - 2 *
124             log((1 + exp(ni)) / (1 + exp(qlogis(
125                 plogis(drop(muscle2$X %*% mle.muscle$
126                 par)))
127             ))), 2, sum)
128 # pesi
129 p <- exp(delta) / sum(exp(delta))
130
131 bootstrap_accuracy <-
132   f_accuracy(
133     tt = t_alpha,
134     aa = t(mle2),
135     pp = exp(delta),
136     V = solve(var(t(mle2)))
137   )
138
139 # principio campionamento ripetuto
140 nsim <- 1000
141 dat <- matrix(NA,
142               ncol = 25,
143               nrow = nsim,
144               byrow = F)
145 for (i in 1:nsim) {
146   for (j in 1:25) {
147     dat[i,j] <- rbinom(1, size = muscle2$n[j], prob =
148       plogis(drop(muscle2$X[j, ] %*% mle.muscle$par)))
149   }
150 }
151 f_cp <- function(data){
152   sim <- mcmc_thin(mcmc_object = Metro_Hastings(lprior
153     , pars = rep(0,5), iterations = 10^5, burn_in = 25000,
154     data=data)
155     , thin = 5)$trace
156   out <- apply(sim, 1, function(e) sum(plogis(drop(data$X[
157     data$X[, 4] == '5', ] %*% e)))) /
158     sum(plogis(drop(data$X[data$X[, 4] == '1'
159     , ] %*% e))))
160   return(list(out=out, sim=sim))
161 }

```

```
158
159 library(snowfall)
160 library(parallel)
161 ncpu <- detectCores() - 1
162 sfInit(parallel = T, cpus = ncpu)
163 sfLibrary(MHadaptive)
164 sfExport(list = c(
165   'nsim',
166   'lposterior',
167   'muscle2',
168   'mle.muscle',
169   'dat',
170   'lprior',
171   'llik.muscle', 'f_cp'
172 ))
173 system.time(cp_t_alpha <-
174   sfApply(dat, 1, function(d) f_cp(data=list(X=muscle2$X,
175     Y=d,n=muscle2$n))))
175 sfStop()
176 gen_freq_cp<- matrix(NA,nrow=nsim,ncol=3)
177 for (j in 1:nsim)
178   gen_freq_cp[j,]<- f_accuracy(tt = cp_t_alpha[[j]]$out,
179     aa = cp_t_alpha[[j]]$sim,pp = rep(1,15001))
179
180 m_pcr<- mean(gen_freq_cp[,1])
181 sdf_pcr<- mean(gen_freq_cp[,2])
182 sd_pcr<- mean(gen_freq_cp[,3])
183 value_cpr<- c(m_pcr,sdf_pcr,sd_pcr)
184
185 general_accuracy
186 bootstrap_accuracy
187 value_cpr
188
189 par(mfrow=c(1,1))
190 plot(x = gen_freq_cp[,3],y = gen_freq_cp[,2],main = '
191   Confronto deviazione standard bayesiana e frequentista
192   ',
193     xlab = 'sd bayesiana',ylab='sd frequentista')
194 abline(a=0,b=1,col=2)
195
196 # differenza sd bayes e frequentista
197 plot(x=gen_freq_cp[,3],y=gen_freq_cp[,3]-gen_freq_cp[,2],
198     xlab='sd bayesiana',ylab='Differenza sd bayesiana e sd
```

```
    frequentista', main = 'Differenza_sdbayesiana_e_sdbayesiana_
    frequentista')
196
197 c(sd(gen_freq_cp[,1]), sdf_pcr, sd_pcr)
```

# Appendice C

## Codici R per il capitolo 4

### C.1 Modello parabola normale

```
1 rm(list=ls())
2 source('f_accuracy.R')
3
4 # genero valori ottenuti da fam esp curva che utilizzerò
   # come dati
5 # normale con media=theta e varianza= theta^2
6 set.seed(1234)
7 y<- rnorm(200,mean=5,sd = 5)
8 y
9
10 # log-verosimiglianza
11 llik<- function(theta,data){
12   out<- sum(dnorm(data,mean = theta,sd=theta,log=T))
13   return(out)
14 }
15
16 # stima di massima verosimiglianza
17 mle<- nlminb(1,objective = function(x) -llik(theta=x,data
   =y),lower = 1e-8)
18 mle$par
19
20 # calcolandolo a mano
21 (-sum(y)+sqrt(sum(y)^2+4*length(y)*sum(y^2)))/(2*length(y
   ))
22
23 # riparametrizzazione con formula 3 e 4 REID 2010
```

```

24 ps<- function(theta,data,mle) -sum(data^2)/(mle*theta^2)+
    sum(data)/(theta*mle)
25
26 # inversa di ps
27 th<- function(psi,data,mle) {
28   (sum(data)/mle-sqrt((sum(data)/mle)^2-4*sum(data^2)/mle
    *psi))/(2*psi)
29 }
30
31 mle_ps<- ps(theta = mle$par,data = y,mle = mle$par)
32 bond<- (sum(y)/mle$par)^2/(4*sum(y^2)/mle$par)
33
34 # log-priori per theta
35 lprior_th<- function(theta,iper){
36   dgamma(theta,shape = iper[1],scale = iper[2],log=T)
37 }
38
39 # log-posteriori per theta
40 lposterior_theta<- function(theta,iper,data){
41   lprior_th(theta=theta,iper=iper)+llik(theta = theta,
    data = data)
42 }
43
44
45 # genero dalla posteriori
46 mcmc_f2<- function(nsim,theta0,iper,data,eps){
47   out<- numeric(nsim)
48   acc<- 0
49   theta<- theta0
50   for(i in 1:nsim){
51     thetas<- theta+runif(1,-eps,eps)
52     alpha<- min(1,exp(lposterior_theta(theta = thetas,
    iper = iper,data = data)-
53       lposterior_theta(theta = theta,
    iper = iper,data = data)))
54     if(runif(1)<alpha){
55       theta<- thetas
56       acc<- acc+1
57     }
58     out[i]<- theta
59   }
60   return(list(value=out,acc=acc/nsim))
61 }

```

```
62
63 # iperparametri
64 iper<- c(10,10)
65 sim<- mcmc_f2(nsim=10^5,theta0=2,iper=iper,data=y,eps=1)
66
67 plot(sim$value)
68 lines(1:10^4,cumsum(sim$value)/(1:10^4),col=2)
69 acf(sim$value)
70
71 burn_in<- 1:25000
72 plot(sim$value[-burn_in])
73 acf(sample(sim$value[-burn_in],15000))
74
75 # valori a posteriori per theta
76 thval<- sample(sim$value[-burn_in],15000)
77
78 # valori a posteriori per phi
79 phival<- ps(theta = thval,data = y,mle = mle$par)
80
81 general_accuracy<- f_accuracy(tt = thval,aa = phival,pp =
      rep(1,15000))
82 general_accuracy
83
84 # versione bootstrap
85 nrep<- 8000
86 sim_data<- matrix(NA,ncol=200,nrow = nrep,byrow = T)
87 for(i in 1:nrep){
88   out<- rnorm(200,mean = mle$par,sd = mle$par)
89   sim_data[i,]<- out
90 }
91
92 # stime di massima verosimiglianza per theta con i nuovi
      campioni
93 mle_th<- apply(sim_data,1,function(x) nlminb(1,function(c
      ) -llik(theta = c,data = x),lower=1e-8)$par)
94
95 #corrispondenti valori per phi
96 phival_boot<- ps(theta = mle_th,data = y,mle = mle$par)
97
98 # otteniamo i pesi per phi
99 r<- priorval<- weight <- rep(NA,nrep)
100 for(i in 1:nrep){
101   r[i]<- llik(theta=th(psi = phival_boot[i],data = y,mle
```

```

      = mle$par), data = y)/
102   llik(theta=th(psi = mle_ps, data = y, mle = mle$par),
      data = sim_data[i,])
103   priorval[i] <- lprior_th(theta=th(psi = phival_boot[i],
      data = y, mle = mle$par), iper = iper)*
104   (4*(sum(y^2)/mle$par*phival_boot[i])*((sum(y)/mle$par)
      )^2-4*sum(y)/mle$par*phival_boot[i])^(-0.5)-2*
105   (sum(y)/mle$par-sqrt((sum(y)/mle$par)^2-4*sum(y^2)
      /mle$par*phival_boot[i])))/(4*phival_boot[i]^2)
106   weight[i] <- priorval[i]*r[i]
107 }
108
109 bootstrap_accuracy <- f_accuracy(tt = mle_th, aa = phival_
      boot, pp = weight)
110 bootstrap_accuracy; general_accuracy
111
112 # principio campionamento ripetuto
113 nsim <- 1000
114 dat <- matrix(NA, ncol=nsim, nrow=200)
115 for(i in 1:nsim) dat[,i] <- rnorm(200, mean=mle$par, sd=mle$
      par)
116
117 f_freq <- function(data, burn, iper, th0, mle, eps){
118   sim_pcr <- sample(mcmc_f2(nsim=10^5, theta0=th0, iper=iper
      , data=data, eps=eps)$value[-burn], 15001)
119   psi_pcr <- ps(theta=sim_pcr, data=y, mle=mle$par)
120   return(list(sim=sim_pcr, psi=psi_pcr))
121 }
122
123 library(parallel)
124 library(snowfall)
125 ncpu <- detectCores()-1
126 burn <- 1:25000
127 th0 <- 2
128 eps <- 1
129 sfInit(parallel = T, cpus = ncpu)
130 sfExport(list=c('llik', 'lposterior_theta', 'lprior_th', '
      dat', 'burn', 'iper', 'th0', 'mle', 'mcmc_f2', 'y', 'eps', 'f_
      freq', 'ps'))
131 ps_pcr <- sfApply(dat, 2, function(d) f_freq(data=d, burn=
      burn, iper=iper, th0=th0, mle=mle, eps=eps))
132 sfStop()
133 gen_acc <- matrix(NA, ncol=3, nrow=nsim)

```



```

134 for(j in 1:nsim) gen_acc[j,]<- f_accuracy(tt = ps_pcr[[j
      ]]$sim,aa = ps_pcr[[j]]$psi,pp = rep(1,15001))
135 par(mfrow=c(1,1))
136 plot(x = gen_acc[,3],y = gen_acc[,2],xlab = 'sd_bayesiana
      ',ylab = 'sd_frequentista',
137       main='Confronto deviazione standard bayesiana e
      frequentista')
138 abline(a = 0,b = 1,col=2)
139 sd_pcr<- mean(gen_acc[,3])
140 sdf_pcr<- mean(gen_acc[,2])
141 mean_pcr<- mean(gen_acc[,1])
142 pcr_accuracy<- c(mean_pcr,sdf_pcr,sd_pcr)
143 bootstrap_accuracy
144 general_accuracy
145 pcr_accuracy
146
147 #grafico differenza sd bayes e frequentista
148 plot(x=gen_acc[,3],y=gen_acc[,3]-gen_acc[,2],xlab='sd_
      bayesiana',ylab='Differenza sd_bayesiana e sd_
      frequentista',main = 'Differenza sd_bayesiana e sd_
      frequentista')
149
150 # confronto valori Monte Carlo
151 c(sd(gen_acc[,1]),sdf_pcr,sd_pcr)

```

## C.2 Modello binomiale probit

```

1 rm(list=ls())
2 source('f_accuracy.R')
3
4 # definizione del dataset
5 muscle<- read.csv('muscle.csv',header = T,sep = ',')
6 muscle$d<- c(rep(1,5),rep(2,5),rep(3,5),rep(4,5),rep(5,5)
      ) # giorno
7 muscle$i<- c(rep(c(1:5),5))
8
9 muscle2<- list(Y=muscle$s,n=muscle$n,X=model.matrix(~
      muscle$i+I(muscle$i^2)+muscle$d+I(muscle$d^2)))
10
11 #log-verosimiglianza
12 llik.muscle<- function(beta,data){
13   eta<- drop(data$X%*%beta)
14   p<- pnorm(eta)

```

```

15  sum(dbinom(x = data$Y,size = data$n,prob=p,log = T))
16  }
17
18  # stima di massima verosimiglianza
19  mle.muscle<- nlminb(rep(0,5),function(x) -llik.muscle(
      beta=x,data=muscle2))
20
21  #log-priori
22  library(mvtnorm)
23  lprior<- function(beta,mean_iper,sigma_iper){
24    dmnorm(beta,mean = mean_iper,sigma = sigma_iper,log=T)
25  }
26
27  #log-posteriori
28  lprior<- function(beta,data,mean_iper,sigma_iper){
29    lprior(beta=beta,mean_iper,sigma_iper)+llik.muscle(beta
      = beta,data=data)
30  }
31
32  # simulo dalla posteriori
33  library(MHadaptive)
34  sim<- Metro_Hastings(lprior,rep(0,5),mean_iper=rep
      (0,5),sigma_iper=diag(100,5),data=muscle2,
35    iterations=100000,burn_in = 25000)
36  sim <- mcmc_thin(sim,thin=5)
37
38  par(mfrow=c(1,5))
39  for(j in 1:5){
40    plot(sim$trace[,j])
41    lines(1:15001,cumsum(sim$trace[,j])/(1:15001),col=2)
42    acf(sim$trace[,j])
43    plot(density((sim$trace[,j])))
44
45  }
46
47  # funzione per ottenere parametro canonico in modello
      esponenziale tangente
48
49  ps<- function(beta,data,mle){
50    out<- apply(data,1,function(c) t(qlogis(pnorm(drop(c%%
      beta))))%%dnorm(drop(c%%mle))%%t(c))
51    out<- apply(out,1,sum)
52    return(out)

```

```

53 }
54
55 # inversa della funzione con pseudo inverse
56 library(matlib)
57 beta_inv<- function(psi,data,mle){
58   A<- apply(data,1,function(d) dnorm(drop(d**mle))**t(d
59   ))
59   AAI<- Ginv(t(A)**A)
60   AI<- AAI**t(A)
61   pn<- qnorm(plogis(AI**psi))
62   XI<- Ginv(t(data)**data)**t(data)
63   out<- XI**pn
64   return(drop(out))
65 }
66
67 psival<- apply(sim$trace,1,function(d) ps(beta = d,data =
68   muscle2$X,mle = mle.muscle$par))
69
70 # parametro d'interesse
71 csi_5<- apply(pnorm(muscle2$X[muscle2$X[,4]== '5',]**t(
72   sim$trace)),2,sum)
73 csi_1<- apply(pnorm(muscle2$X[muscle2$X[,4]== '1',]**t(
74   sim$trace)),2,sum)
75
76 gamma_val<- csi_5/csi_1
77
78 # alpha_y ottenuti derivando per y il modello formula 2.1
79 # della tesi
80 alpha_y<- apply(muscle2$X,1,function(e) qlogis(pnorm(drop
81   (e**t(sim$trace)))))
82
83 # accuratezza generale
84 general_accuracy_psi<- f_accuracy(tt = gamma_val,aa = t(
85   psival),pp = rep(1,nrow(sim$trace))) # con psi
86 general_accuracy_alpha<- f_accuracy(tt = gamma_val,aa =
87   alpha_y,pp = rep(1,nrow(sim$trace))) # con alpha_y
88
89 confronto<- matrix(rbind(general_accuracy_psi, general_
90   accuracy_alpha),nrow=2,
91   dimnames = list(c('psi','alpha_y'),c('
92   Ebayes','sdFreq','sdBayes'))))
93
94 # versione bootstrap

```

```

86 # genero i dati
87 s <- matrix(NA, ncol = 25, nrow = 8000)
88 for (i in 1:8000) {
89   for (j in 1:25) {
90     s[i, j] <- rbinom(1, size = muscle2$n[j], prob =
91       pnorm(drop(muscle2$X[j, ] %*% mle.muscle$par)))
92   }
93 }
94
95 # stime mle con i nuovi dati
96 mle_beta<- apply(s,1,function(d) nlminb(start = rep(0,5),
97   objective = function(f)
98   -llik.muscle(beta = f,data = list(X=muscle2$X,n=muscle2
99   $n,Y=d)))$par)
100
101 xi_5_boot <-
102   apply(mle_beta, 2, function(g)
103     sum(pnorm(drop(muscle2$X[muscle2$X[, 4] == '5', ] %*%
104       g))))
105 xi_1_boot <-
106   apply(mle_beta, 2, function(g)
107     sum(pnorm(drop(muscle2$X[muscle2$X[, 4] == '1', ] %*%
108       g))))
109
110 gamma_val_boot <- xi_5_boot / xi_1_boot
111
112 # valori psi partendo dai valori di beta
113 mle_psi<- apply(mle_beta,2,function(e) ps(beta = e,data =
114   muscle2$X,mle = mle.muscle$par))
115
116 # valori per alpha_y
117 alpha_y_boot<- apply(muscle2$X,1,function(e) qlogis(pnorm
118   (drop(e%*%mle_beta))))
119
120 # definisco i pesi per beta
121
122 p_beta<- r_beta<- rep(NA,8000)
123 for(j in 1:8000){
124   r_beta[j]<- llik.muscle(beta = mle_beta[,j],data =
125     muscle2)/
126   llik.muscle(beta = mle.muscle$par,data = list(X=
127     muscle2$X,n=muscle2$n,Y=s[j,]))

```

```

120   p_beta[j]<- lprior(beta = mle_beta[,j],mean_iper = rep
      (0,5),sigma_iper = diag(100,5))
121 }
122
123 w_beta<- p_beta*r_beta
124
125 boot_accuracy_alpha<- f_accuracy(tt= gamma_val_boot,aa=
      alpha_y_boot,pp=w_beta)
126
127 #definisco i pesi per phi
128 # defnisco lo jacobiano
129 library(pracma)
130 Jacob<- function(beta,data,mle){
131   jacobian(function(s) ps(beta=s,data=data,mle=mle),x0 =
      beta)
132 }
133
134 p_psi<- r_psi <- J <- rep(NA,8000)
135 for(j in 1:8000){
136   r_psi[j]<- llik.muscle(beta=beta_inv(psi = mle_psi[,j]
      ],data = muscle2$X,mle = mle.muscle$par),data=muscle2)
      /
137   llik.muscle(beta = mle.muscle$par,data = list(X=
      muscle2$X,n=muscle2$n,Y=s[j,]))
138   p_psi[j]<- lprior(beta = beta_inv(psi = mle_psi[,j],
      data = muscle2$X,mle = mle.muscle$par),
139     mean_iper = rep(0,5),sigma_iper = diag
      (100,5))
140   J[j]<- det(Jacob(beta=mle_beta[,j],mle = mle.muscle$par
      ,data = muscle2$X))(-1)
141
142 }
143
144 w_psi<- r_psi*p_psi*J
145
146 # accuratezza per psi
147 boot_accuracy_psi<- f_accuracy(tt = gamma_val_boot,aa = t
      (mle_psi),pp = w_psi)
148
149 # confronto dei valori
150 confronto_boot<- matrix(rbind(boot_accuracy_psi[-3], boot
      _accuracy_alpha[-3]),nrow=2,
151   dimnames = list(c('psi','alpha_y'),c('

```

```

      Ebayes', 'sdFreq', 'sdBayes'))))
152 confronto_boot
153
154 # principio campionamento ripetuto
155 nsim <- 1000
156 dat <- matrix(NA, ncol = 25, nrow = nsim, byrow = F)
157 for (i in 1:nsim) {
158   for (j in 1:25) {
159     dat[i,j] <- rbinom(1, size = muscle2$n[j], prob =
      pnorm(drop(muscle2$X[j, ] %*% mle.muscle$par)))
160   }
161 }
162
163 f_cp<- function(data){
164   sim<- mcmc_thin(mcmc_object = Metro_Hastings(lposterior
      ,pars = rep(0,5),iterations = 10^5,burn_in = 25000,
165     data=data,
      mean_iper=rep(0,5),sigma_iper=diag(100,5)),thin = 5)$
      trace
166   out<- apply(sim,1,function(e) sum(pnorm(drop(data$X[
      data$X[, 4] == '5', ] %*% e)))/
167     sum(pnorm(drop(data$X[data$X[, 4] == '1',
      ] %*% e))))
168   return(list(out=out, beta=sim))
169 }
170
171 library(snowfall)
172 library(parallel)
173 ncpu <- detectCores() - 1
174 sfInit(parallel = T, cpus = ncpu)
175 sfLibrary(MHadaptive)
176 sfLibrary(mvtnorm)
177 sfExport(list = c(
178   'nsim',
179   'lposterior',
180   'muscle2',
181   'mle.muscle',
182   'dat',
183   'lprior',
184   'llik.muscle', 'f_cp'
185 ))
186 system.time(cp_t_alpha <-
187   sfApply(dat, 1, function(d) f_cp(data=list(

```

```

      X=muscle2$X,Y=d,n=muscle2$n)))
188 sfStop()
189
190 # beta simulati
191 beta<- array(NA,dim = c(15001,5,nsim))
192 for (i in 1:nsim)
193   beta[, ,i]<- cp_t_alpha[[i]]$beta
194
195 # phi ottenuti da beta
196 psi_cp<- array(NA,dim=c(15001,5,nsim))
197 for(i in 1:nsim)
198   for(j in 1:15001)
199     psi_cp[j, ,i]<- ps(beta=beta[j, ,i],data = muscle2$X,
      mle = mle.muscle$par)
200
201 # alpha_y ottenuti da beta
202 alpha_y_cp<- array(NA,dim = c(25,15001,nsim))
203 for (i in 1:nsim)
204   alpha_y_cp[, ,i]<- qlogis(pnorm(drop(muscle2$X%*%t(beta
      [, ,i])))
205
206 gen_acc_cp_alpha<- gen_acc_cp_psi<- matrix(NA,nrow=nsim,
      ncol=3)
207 for (j in 1:nsim) {
208   gen_acc_cp_alpha[j,]<- f_accuracy(tt = cp_t_alpha[[j]]$
      out,aa = t(alpha_y_cp[, ,j]),pp = rep(1,15001))
209   gen_acc_cp_psi[j,]<- f_accuracy(tt = cp_t_alpha[[j]]$
      out,aa = psi_cp[, ,j],pp = rep(1,15001))
210 }
211
212 value_cpr_psi<- apply(gen_acc_cp_psi,2,mean)
213 value_cpr_alpha<- apply(gen_acc_cp_alpha,2,mean)
214
215 plot(x = gen_acc_cp_alpha[,3],y = gen_acc_cp_alpha[,2],
      main = 'Confronto deviazione standard bayesiana e
      frequentista per alpha',
216       xlab = 'sd bayesiana',ylab='sd frequentista')
217 abline(a=0,b=1,col=2)
218
219 plot(x = gen_acc_cp_psi[,3],y = gen_acc_cp_psi[,2],main =
      'Confronto deviazione standard bayesiana e
      frequentista per phi',
220       xlab = 'sd bayesiana',ylab='sd frequentista')

```

```

221 abline(a=0,b=1,col=2)
222
223 #confronto tra sd freq di alpha e psi
224 plot(x = gen_acc_cp_psi[,3],y = gen_acc_cp_alpha[,3],main
      = 'Confronto deviazione standard frequentista per
      alpha e phi',
225       xlab = expression(varphi),ylab=expression(alpha))
226 abline(a=0,b=1,col=2)
227
228 plot(x = gen_acc_cp_psi[,2],y = gen_acc_cp_alpha[,2],main
      = 'Confronto deviazione standard bayesiana per alpha
      e phi',
229       xlab = expression(varphi),ylab=expression(alpha))
230 abline(a=0,b=1,col=2)
231
232
233 # confronto con tutti i metodi
234 confronto_tot<- matrix(rbind(general_accuracy_alpha,
      general_accuracy_psi,
235                               boot_accuracy_alpha[-3],boot
      _accuracy_psi[-3],
236                               value_cpr_alpha,value_cpr_
      psi),nrow=6,
237                               dimnames = list(c('gen_acc_alpha',
      'gen_acc_psi','boot_alpha',
238                                               'boot_psi','pcr_
      alpha','pcr_psi')),
239                                               c('EBayes','sdFreq
      ','sdBayes'))))
240 confronto_tot
241
242 # grafico differenza sd bayes e frequentista
243 plot(x=gen_acc_cp_psi[,3],y=gen_acc_cp_psi[,3]-gen_acc_cp
      _psi[,2],xlab='sd bayesiana',ylab='Differenza sd
      bayesiana e sd frequentista',main = 'Differenza sd
      bayesiana e sd frequentista')
244 plot(x=gen_acc_cp_alpha[,3],y=gen_acc_cp_alpha[,3]-gen_
      acc_cp_alpha[,2],xlab='sd bayesiana',ylab='Differenza
      sd bayesiana e sd frequentista',main = 'Differenza sd
      bayesiana e sd frequentista')
245
246 # confronto valori Monte Carlo
247 c(sd(gen_acc_cp_alpha[,1]),value_cpr_alpha[2:3])

```



```
248 c(sd(gen_acc_cp_psi[,1]),value_cpr_psi[2:3])
```



# Bibliografia

- Brazzale, A. R., A. C. Davison e N. Reid (2007). *Applied asymptotics: case studies in small-sample statistics*. Cambridge University Press.
- Chivers, C. (2012). *MHadaptive: General Markov Chain Monte Carlo for Bayesian Inference using adaptive Metropolis-Hastings sampling*. R package version 1.1-8. URL: <https://CRAN.R-project.org/package=MHadaptive>.
- Efron B. T. Hastie, I. Johnstone e R. Tibshirani (2004). «Least Angle Regression». In: *The Annals of Statistics* 32.2, pp. 407–451.
- Efron, B. (1975). «Defining the Curvature of a Statistical Problem (with Applications to Second Order Efficiency)». In: *The Annals of Statistics* 3.6, pp. 1189–1242.
- (1981). «Nonparametric Standard Errors and Confidence Intervals». In: *The Canadian Journal of Statistics* 9.2, pp. 139–158.
- (2012). «Bayesian inference and the parametric bootstrap». In: *The Annals of Applied Statistics* 6.4, pp. 1971–1997.
- (giu. 2015). «Frequentist accuracy of Bayesian estimates». In: *Journal of the Royal Statistical Society. Series B, Statistical methodology* 77.3, pp. 617–646.
- Efron, B. e T. Hastie (2016). *Computer Age Statistical Inference*. Cambridge University Press.
- Fisher, R. A. (1934). «Two New Properties of Mathematical Likelihood». In: *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character* 144.852, pp. 285–307.
- Fraser, D. A. S. (1990). «Tail Probabilities from Observed Likelihoods». In: *Biometrika* 77.1, pp. 65–76.
- Fraser, D.A.S. e N. Reid (1995). «Ancillaries and Third Order Significance». In: *Utilitas Mathematica* 7, pp. 33–53.
- Kent, John T. (1982). «Robust Properties of Likelihood Ratio Test». In: *Biometrika* 69.1, pp. 19–27.
- Liseo, B. (2008). *Introduzione alla statistica bayesiana*. Dispense didattiche.

- Pace, L. e A. Salvan (1997). *Principles of statistical inference from a neo-fisherian perspective*. World scientific.
- Park, T. e G. Casella (2008). «The Bayesian Lasso». In: *Journal of the American Statistical Association* 103.482, pp. 681–686.
- Reid, N. (2003). «Asymptotics and the Theory of Inference». In: *The Annals of Statistics* 31.6, pp. 1695–1731.
- Reid, N. e D.A.S. Fraser (2010). «Mean loglikelihood and higher-order approximations». In: *Biometrika* 97.1, pp. 159–170.
- Skovgaard, Ib M. (1996). «An Explicit Large-Deviation Approximation to One-Parameter Tests». In: *Bernoulli* 2.2, pp. 145–165.
- Tibshirani, R. (2011). «Regression shrinkage and selection via the lasso: a retrospective». In: *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* 73.3, pp. 273–282.