

# Indice

## Capitolo 1

Introduzione.....	5
-------------------	---

## Capitolo 2

Inferenza bayesiana e algoritmi di simulazione.....	7
2.1 <i>Che cos'è l'Inferenza bayesiana?</i> .....	7
2.2 <i>Algoritmi di simulazione</i> .....	8
2.2.1 <i>Importance Sampling</i> .....	9
2.2.2 <i>Markov Chain of Montecarlo con passeggiata casuale(MCMC)</i> .....	10
2.2.3 <i>Iterative Importance Sampling (IIS)</i> .....	12

## Capitolo 3

Confronto tra i tre metodi di simulazione nel caso normale.....	15
3.1 <i>Specificazione del modello normale con media ignota e varianza nota</i> .....	15
3.1.1 <i>Importance Sampling</i> .....	16
3.1.2 <i>Markov Chain of Montecarlo</i> .....	19
3.1.3 <i>Iterative Importance Sampling</i> .....	21
3.1.4 <i>Valutazione della bontà degli algoritmi di simulazione tramite EQM</i> .....	27
3.1.5 <i>Conclusioni</i> .....	35
3.2 <i>Specificazione del modello normale con media e varianza ignote</i> .....	36
3.2.1 <i>Importance Sampling</i> .....	38
3.2.2 <i>Markov Chain of Montecarlo</i> .....	40

3.2.3 <i>Iterative Importance Sampling</i> .....	43
3.2.4 <i>Valutazione della bontà degli algoritmi di simulazione tramite EQM</i> .....	45
3.2.5 <i>Conclusioni</i> .....	55

## Capitolo 4

Inferenza bayesiana per valori estremi.....	57
4.1 <i>Richiami sulla teoria dei valori estremi: l'approccio classico</i> .....	58
4.2 <i>Inferenza per la Generalized Extreme Value (GEV)</i> .....	63
4.3 <i>Valori estremi e statistica bayesiana</i> .....	65
4.4 <i>Algoritmi di simulazione per GEV</i> .....	66
4.4.1 <i>Importance Sampling per valori estremi</i> .....	66
4.4.2 <i>MCMC e Valori Estremi</i> .....	69
4.4.3 <i>IIS per valori estremi</i> .....	74
4.5 <i>Qual è l'algoritmo migliore?</i> .....	77

## Capitolo 5

Simulazione e statistica bayesiana per dati temporali.....	81
5.1 <i>Introduzione</i> .....	81
5.2 <i>Simulazione campionaria</i> .....	82
5.3 <i>Un confronto tra MCMC e IIS per filtraggio</i> .....	84
5.3.1 <i>MCMC per filtraggio</i> .....	85
5.3.2 <i>Iterative Importance Sampling per filtraggio</i> .....	89
5.3.4 <i>Un confronto tra MCMC e IIS</i> .....	92
5.4 <i>Un'analisi con dati reali</i> .....	94
5.4.1 <i>Lisciamento con Markov Chain of Montecarlo e IIS</i> .....	95

## Capitolo 6

Conclusioni.....	99
------------------	----

Appendice.....	101
Bibliografia.....	111
Ringraziamenti.....	113



# CAPITOLO 1

## INTRODUZIONE

In ambito bayesiano, introduciamo i concetti di distribuzione a priori ed a posteriori: la simulazione diretta dalla distribuzione a posteriori è, nella maggior parte dei casi, alquanto difficile. La simulazione dalla “a priori” invece, come vedremo, può portare ad una varianza troppo elevata dello stimatore. Dobbiamo quindi trovare dei metodi che permettano di fare simulazioni che garantiscano stime non distorte ed efficienti, ma che, allo stesso tempo, siano semplici da realizzare.

L’obiettivo principale che ci si prefigge in quest’elaborato è appunto quello di confrontare come si comportano (cercandone di misurare, per quanto possibile, precisione ed efficienza) alcuni algoritmi di simulazione per problemi d’inferenza bayesiana (quando oltre all’informazione campionaria, è disponibile un’informazione a priori). I tre metodi che analizzeremo sono: *Importance Sampling* (*IS*, o anche detto campionamento per importanza), *MCMC* (Markov Chain of Montecarlo con passeggiata casuale) e *Iterative Importance Sampling* (*IIS*). Proveremo a valutare, simulando situazioni differenti, quale sia il metodo migliore per precisione e variabilità in ogni situazione prefissata.

Prima di tutto questo, nel Capitolo 2, ripasseremo il concetto d’inferenza Bayesiana e daremo una veloce descrizione dei metodi di simulazione sopradescritti. Nel capitolo 3, valuteremo come si comportano i tre algoritmi di simulazione nel caso normale con media ignota e varianza nota e nel caso con

entrambi i parametri ignoti. Nei capitoli 4 e 5, invece, vedremo un'applicazione di questi metodi in problemi d'inferenza bayesiana per valori estremi e per dati temporali.

Nel capitolo 6, infine, si “tireranno” le conclusioni.

## CAPITOLO 2

### INFERENZA BAYESIANA E ALGORITMI DI SIMULAZIONE

#### *2.1 Che cos'è l'inferenza bayesiana?*

Ci sono alcune situazioni nelle quali sono disponibili delle informazioni a priori sui parametri coinvolti nell'esperimento statistico (quest'informazione è di solito esprimibile nella forma di una distribuzione di probabilità). Queste possono derivare da studi precedenti fatti sullo stesso fenomeno o da indagini pilota fatte sulla popolazione, o anche solo ipotizzate dal ricercatore.... Quando si hanno a disposizione questo genere di informazioni aggiuntive, si entra nell'ambito della statistica bayesiana. Grazie al potente strumento rappresentato dal teorema di Bayes possiamo combinare l'informazione proveniente dai dati (la realizzazione dell'esperimento statistico) e quella disponibile a priori ottenendo una distribuzione a posteriori che dovrebbe (se la scelta della priori ha senso per il problema in questione) rappresentare in maniera chiara e precisa il fenomeno e i parametri della distribuzione d'interesse (potendo fare inferenza, bayesiana appunto, sui parametri).

In generale, il teorema di Bayes applicato a problemi di inferenza bayesiana, assume la seguente forma:

$$(i) \quad f(\theta | x) = (f(\theta) * f(x | \theta)) / \int (f(\theta) * f(x | \theta)) d\theta$$

Dove, con  $f(x | \theta)$  intendiamo l'informazione proveniente dai dati (di solito espressa tramite la funzione di verosimiglianza), con  $f(\theta)$  l'informazione

a priori e con  $f(\theta | x)$  la distribuzione a posteriori. La quantità  $\int (f(\theta) * f(x | \theta)) d\theta$  rappresenta, nel caso continuo, la costante di normalizzazione. Dove  $\theta$  è  $k$ -dimensionale (per semplicità e chiarezza espositiva considereremo massimo  $k=2$ ) e  $x$  è un vettore di numerosità  $n$ . Se fossimo nel mondo discreto, invece che l'integrale della (i) avremmo la  $\sum (f(\theta_j) * f(x | \theta_j))$ . Ma, comunque, il denominatore della (i) è funzione della sola  $x$  (dopo che si è integrato rispetto a  $\theta$ ) e può quindi essere trattato come una costante. La (i) diventa quindi:

$$(ii) \quad f(\theta | x) \propto (f(\theta) * f(x | \theta))$$

Che a parole, non significa altro che: la posteriori è proporzionale sia alla priori che alla verosimiglianza. La scelta della “a priori” sarà quindi molto importante, perché come vedremo, una sua “cattiva” scelta, comporterà una distribuzione a posteriori che mal approssima i veri valori del parametro (o dei parametri nel caso  $k=2$ ). Questo si verifica soprattutto con campioni piccoli generati dalla  $f(x|\theta)$ . Con campioni di numerosità elevata (ad esempio 1000) l'effetto della a priori sarà quasi irrilevante.

## 2.2 Algoritmi di simulazione

Come già detto in precedenza, simulare dalla a posteriori  $f(\theta | x)$  è alquanto difficile (vista la presenza dell'integrale). E la simulazione diretta dalla a priori può comportare stime con varianza elevata (visto che, molte volte, non corrisponde affatto alla “a posteriori”). Bisogna trovare, quindi, delle procedure che ci permettano di ottenere buone simulazioni della  $f(\theta | x)$  senza passare direttamente da essa. E per fare questo, ci vengono incontro i metodi di simulazione del tipo IS, MCMC e IIS. Ci sono moltissimi altri algoritmi di

simulazione (tipo l’algoritmo di accettazione rifiuto o il metodo per inversione) ma non tutti permettono di simulare “facilmente” come IS, MCMC e IIS ( ad esempio, molte volte, è alquanto difficile invertire la funzione di ripartizione, soprattutto con due o più parametri d’interesse...). Dal prossimo paragrafo, gli algoritmi di simulazione di riferimento saranno presi in esame uno ad uno sottolineandone le caratteristiche principali, i vantaggi e gli svantaggi.

### 2.2.1 Importance Sampling (IS)

Per la sua semplicità, il campionamento per importanza è uno dei metodi di simulazione più utilizzato. Il concetto che sta alla base di quest’algoritmo è campionare i valori di  $\theta$  tenendo conto del valore della funzione di densità in corrispondenza di ogni valore di  $\theta$ . Più alto è il valore della  $f$  in riferimento ad un certo  $\theta_i$  e maggiore è la probabilità che  $\theta_i$  venga preso come valore della  $f(\theta|x)$ . Il nostro obiettivo, lo ricordiamo, è quello di simulare da  $f(\theta | x)$ . Che, come ricordiamo, è uguale a:

$$f(\theta | x) = (f(\theta) * f(x | \theta)) / \int (f(\theta) * f(x | \theta)) d\theta$$

Possiamo schematizzare l’algoritmo IS nei seguenti 5 passi:

- 1) Simuliamo  $\theta_1, \theta_2, \dots, \theta_m$  da una distribuzione  $g(\theta)$  da cui sia semplice simulare (nei nostri esempi simuleremo sempre dalla a priori ma non vuol dire che un’altra scelta sia sbagliata);

- 2) Calcoliamo i pesi  $w_k$  nel seguente modo:  $w_k = (f(\theta_k) f(x|\theta_k)) / g(\theta_k)$  (nel caso venga scelta come  $g(\theta)$  la “a priori”, la  $f(\theta)$  si semplifica e  $w_k = f(x|\theta_k)$ );
- 3) Normalizziamo  $w_k$  in modo che  $\sum w_k = 1$ ;
- 4) Ricampioniamo  $\theta_1^*, \theta_2^*, \dots, \theta_k^*, \dots, \theta_m^*$  da  $\theta_1, \theta_2, \dots, \theta_k, \dots, \theta_m$  con pesi pari a  $w_1, w_2, \dots, w_k, \dots, w_m$ ;
- 5) Per  $m$  grande, la densità di  $\theta_1^*, \theta_2^*, \dots, \theta_k^*, \dots, \theta_m^* \cong f(\theta | x)$ ;

Per una dimostrazione di questo procedimento si vedano *Smith e Gelfand, 1992*. La scelta della funzione  $g$  da utilizzare diventa, quindi, di fondamentale importanza. Si deve infatti trovare una densità  $g$  da cui sia facile simulare e che permetta di ottenere stime con varianza bassa. Per far questo, importante sarà scegliere una  $g$  tale che rispecchi l'andamento della posteriori. La scelta di una  $g$  con code meno pesanti della a posteriori porterà l'integrale in (i) a divergere. Una scelta di  $g$  con code più pesanti della a posteriori è preferibile, anche se la cosa migliore, se possibile e come già ricordato, è prendere come  $g$  una funzione di densità che rispecchi l'andamento della a posteriori (e che, quindi, faccia corrispondere code con code e punto di massimo con punto di massimo).

### 2.2.2 Markov Chain of Montecarlo con passeggiata casuale (MCMC)

Markov chain of Montecarlo con passeggiata casuale è un metodo generico per simulare da distribuzioni ( nel nostro caso  $f(\theta | x)$  ) da cui sarebbe difficile simulare altrimenti. Il trucco, è quello di costruire una catena markoviana (con determinate caratteristiche su cui qui non ci soffermiamo, per approfondimenti si veda *Coles, 2004*) che permetta di simulare facilmente e,

allo stesso tempo, abbia  $f(\theta | x)$  come distribuzione limite. L'algoritmo è descritto, approssimativamente, nelle seguenti fasi (per informazioni aggiuntive vi rimandiamo a *Robert C. e Casella G., 2004*):

- 1) Si decide il punto di partenza della catena ( $\theta_0$ );
- 2) Si calcola la quantità  $\theta^* = \theta_i + u[-\epsilon, +\epsilon]$  (dove alla prima iterazione  $\theta_i = \theta_0$ );
- 3) Viene calcolata la probabilità  $\alpha$  come il  $\min\{1, (f(\theta^*)f(x|\theta^*)) / (f(\theta_i)f(x|\theta_i))\}$ ;
- 4) Campioniamo  $\theta_{i+1}$  come  $\theta_{i+1} = \theta^*$  con la probabilità  $\alpha$  o come  $\theta_{i+1} = \theta_i$  con probabilità  $1 - \alpha$  ( $\alpha$  è quello calcolato al punto 3);
- 5) Ripetere  $M$  volte i punti dal 2 al 4;

Si dimostra che, per  $M$  grande, la distribuzione ottenuta è approssimativamente  $f(\theta | x)$ . Ma attenzione, i primi  $m$  valori (con  $m$  molto più piccolo di  $M$ ) corrispondono al tempo di convergenza della catena (anche detto periodo di burn-in) e devono essere eliminati perché non approssimano bene la a posteriori. Una delle cose più complicate di quest'algoritmo, è appunto capire quanto grande deve essere  $m$ , visto che la velocità di convergenza varia di volta in volta e in base alle caratteristiche specificate all'inizio dell'algoritmo. Anche i restanti valori  $\theta_{m+1}, \theta_{m+2}, \dots, \theta_M$  continuano ad essere dipendenti. Ma, a differenza di  $\theta_1, \theta_2, \dots, \theta_{m-1}$  la loro distribuzione marginale  $f(\theta | x)$  rimane valida per ogni termine, approssimando (e bene) la vera distribuzione a posteriori. Quindi  $\theta_1, \theta_2, \dots, \theta_{m-1}$  rappresentano un campione (non casuale semplice) della distribuzione a posteriori  $f(\theta | x)$ . Se poi, si vuole limitare la dipendenza nel campione ottenuto, si può filtrare la serie: ad esempio, dopo aver tolto il burn-in si può tenere un valore ogni 10 (con questo

metodo rimane solo una dipendenza leggera). Anche la scelta del valore di  $\epsilon$  è di fondamentale importanza: maggiore è  $\epsilon$  e più veloce è la convergenza dell'algoritmo e la dipendenza della catena markoviana creata..

Il principale limite di questo metodo, è dato dal fatto che, il valore iniziale ( $\theta_0$ ) e il valore di  $\epsilon$ , sono del tutto arbitrari.

### 2.2.3 Iterative Importance Sampling (IIS)

Il campionamento per importanza iterativo (traduzione italiana di IIS) è stato costruito cercando di “raccoliere” quanto di buono IS e MCMC permettevano di fare, eliminandone, per quanto possibile, i difetti. IIS (*Andrieu, Doucet e Robert, 2004*) lavora in 2 stadi: al primo, fa un Importance Sampling “classico” come quello descritto nel paragrafo 2.2.1; al secondo una sorta di MCMC dove, per ricampionare i  $\theta$ , costruisce dei pesi  $w$  simili a quelli creati con IS. Schematizzando IIS lavora così:

1° stadio (facciamo IS):

- 1) Simuliamo  $\theta_1, \theta_2, \dots, \theta_k, \dots, \theta_m$  da una distribuzione  $g(\theta)$  da cui sia semplice simulare (nei nostri esempi simuleremo sempre dalla a priori ma non vuol dire che un'altra scelta sia sbagliata);
- 2) Calcoliamo i pesi  $w_k$  nel seguente modo:  $w_k = (f(\theta_k) f(x | \theta_k)) / g(\theta_k)$  (nel caso venga scelta come  $g(\theta)$  la a priori, la  $f(\theta)$  si semplifica e  $w_k = f(x | \theta_k)$ );
- 3) Normalizziamo  $w_k$  in modo che  $\sum w_k = 1$  ;

- 4) Ricampioniamo  $\theta_1^*, \theta_2^*, \dots, \theta_k^*, \dots, \theta_m^*$  da  $\theta_1, \theta_2, \dots, \theta_k, \dots, \theta_m$  con pesi pari a  $w_1, w_2, \dots, w_k, \dots, w_m$ ;
- 5) Per  $m$  grande, la densità di  $\theta_1^*, \theta_2^*, \dots, \theta_k^*, \dots, \theta_m^* \cong f(\theta | x)$ ;

2° stadio:

- 1) Cambiamo  $\theta_k^*$  con  $\theta_k^{**} = \theta_k^* + u [-\varepsilon, +\varepsilon]$ ;
- 2) Calcoliamo i nuovi pesi  $w_k = f(\theta_k^{**}) f(x | \theta_k^{**})$ ;
- 3) Normalizziamo  $w_k$  in modo che  $\sum w_k = 1$ ;
- 4) Ricampioniamo  $\theta_1^{***}, \theta_2^{***}, \dots, \theta_k^{***}, \dots, \theta_m^{***}$  da  $\theta_1^{**}, \theta_2^{**}, \dots, \theta_k^{**}, \dots, \theta_m^{**}$  con pesi pari a  $w_1, w_2, \dots, w_k, \dots, w_m$ ; Si dimostra che, anche i  $\theta_k^{***}$  così creati, approssimano  $f(\theta | x)$ ;
- 5) Si riparte dal passo 1 del 2° stadio (in tutto, si rifà la procedura del 2° stadio N-1 volte);

IIS permette quindi di creare N campioni che approssimano tutti  $f(\theta | x)$ . Già il primo campione (quello creato con IS al primo stadio) approssima abbastanza bene la nostra “a posteriori”. Al 2° passo però, vengono creati campioni che tendono ad essere sempre più precisi (spostandosi approssimativamente di un  $\varepsilon$  di volta in volta) che permettono di stimare sempre più accuratamente la  $f(\theta | x)$ . Più grande è N e maggiore sarà l’approssimazione alla a posteriori. La “potenza” di questo metodo è quindi molto grande, dato che dovrebbe permettere (con l’aumentare di N) di ottenere stime sempre più precise e che permettano di ridurre, e di molto, la variabilità rispetto ad Importance Sampling (il quale, peraltro, dà già stime che sono non distorte ma alquanto variabili). Inoltre, in IIS non è richiesta la specificazione di un valore iniziale per cominciare l’algoritmo. Molte volte in algoritmi del tipo

MCMC, il difficile è proprio capire quale può essere un buon valore  $\theta_0$  che permetta di inizializzare l'algoritmo. La scelta del valore iniziale di MCMC è critica soprattutto perché da questa scelta, dipende anche la velocità di convergenza di tutta la catena markoviana.... Se  $\theta_0$  è lontano dal vero valore di  $\theta$  ci vorranno più simulazioni prima di ottenere stime "buone" che rispecchino la  $f(\theta | x)$ . Evitare di specificare un valore iniziale per IIS può essere quindi una cosa importante.

## CAPITOLO 3

### CONFRONTO TRA I TRE METODI DI SIMULAZIONE NEL CASO NORMALE

In questo capitolo, forniremo due esempi di applicazione degli algoritmi di simulazione studiati nel capitolo due. Nel primo (paragrafo 3.1), studieremo il caso normale-normale (dove, la prima citata è la a priori, la seconda la a posteriori) con media ignota e varianza nota. Nel secondo (paragrafo 3.2), il caso normale con media e varianze ignote. Abbiamo scelto di “utilizzare” questi due esempi perché, di entrambi, conosciamo la distribuzione a posteriori teorica. Potremmo quindi confrontare i risultati empirici, ottenuti simulando, con quelli teorici cercando di valutare la precisione dei 3 metodi discussi nel capitolo precedente. E’ stata semplicemente una scelta, potevamo benissimo fornire altri esempi di cui conosciamo la distribuzione teorica (tipo gli esempi esponenziale - gamma o poisson – gamma). I risultati ottenuti, come vedremo, saranno generalizzabili anche negli altri casi “noti” dove la “a posteriori” è, appunto, conosciuta.

#### *3.1 Specificazione del modello normale con media ignota e varianza nota*

Supponiamo di avere un campione  $X_1, X_2, \dots, X_n$  che condizionatamente al valore del parametro  $\mu$ , sia indipendente e con la stessa distribuzione  $N(\mu, \sigma_0^2)$  con  $\sigma_0^2$  costante nota. La funzione di verosimiglianza (eliminando tutti i fattori che non dipendono dal parametro di interesse) è quindi:

$$f(x | \mu) \propto L(\mu) \propto \exp \left\{ -\frac{n(\bar{x} - \mu)^2}{2\sigma^2} \right\}$$

Assumiamo, poi, che la distribuzione a priori  $f(\mu)$  sia del tipo  $N(\mu_0, \tau^2)$  con  $\mu_0$  e  $\tau^2$  costanti da noi scelte, determinate in base alle specifiche del problema. Risulta così che:

$$f(\mu) \propto \exp \left\{ -\frac{(\mu - \mu_0)^2}{2\tau^2} \right\}$$

e, per il teorema di Bayes, la distribuzione a posteriori sarà quindi:

$$\begin{aligned} f(\mu | x_1, x_2, \dots, x_n) &\propto f(\mu)L(\mu) \\ &\propto \exp \left\{ -\frac{1}{2} \left( \frac{(\mu - \mu_0)^2}{\tau^2} + n \frac{(\bar{x} - \mu)^2}{\sigma^2} \right) \right\} \\ \text{(i)} \quad &\propto \exp \left\{ -\frac{1}{2} \frac{(\mu - \mu^*)^2}{\tau^{*2}} \right\}; \end{aligned}$$

dove  $\mu^* = (\mu_0\sigma^2 + n\tau^2\bar{x})/(\sigma^2 + n\tau^2)$  e  $\tau^{*2} = \sigma^2\tau^2 / (\sigma^2 + n\tau^2)$  e la formula (i) è riconoscibile come il nucleo di una distribuzione normale di media  $\mu^*$  e deviazione standard  $\tau^*$ .

Ora, segue l'implementazione di quanto detto in questo paragrafo. Proveremo a valutare come variano le “simulazioni” a seconda del metodo utilizzato.

### 3.1.1 Importance Sampling

Supponiamo di avere un campione di numerosità pari a 10 proveniente da una Normale Standardizzata (campione che manterremo uguale anche negli altri metodi in modo da poterne dare diretta confrontabilità), di conoscere il

valore di  $\sigma_0^2$  costante nota (ipotizzato essere uguale ad uno) e di assumere una distribuzione a priori comunque  $N(0,1)$ . Il campione (di numerosità 100) della  $f(\mu | x)$  ottenuto dalla simulazione con Importance Sampling è “approssimativamente” rappresentativo della distribuzione a posteriori  $N(\mu^*, \tau^{*2})$ . Nella figura 3.1, sono rappresentate le stime ottenute:

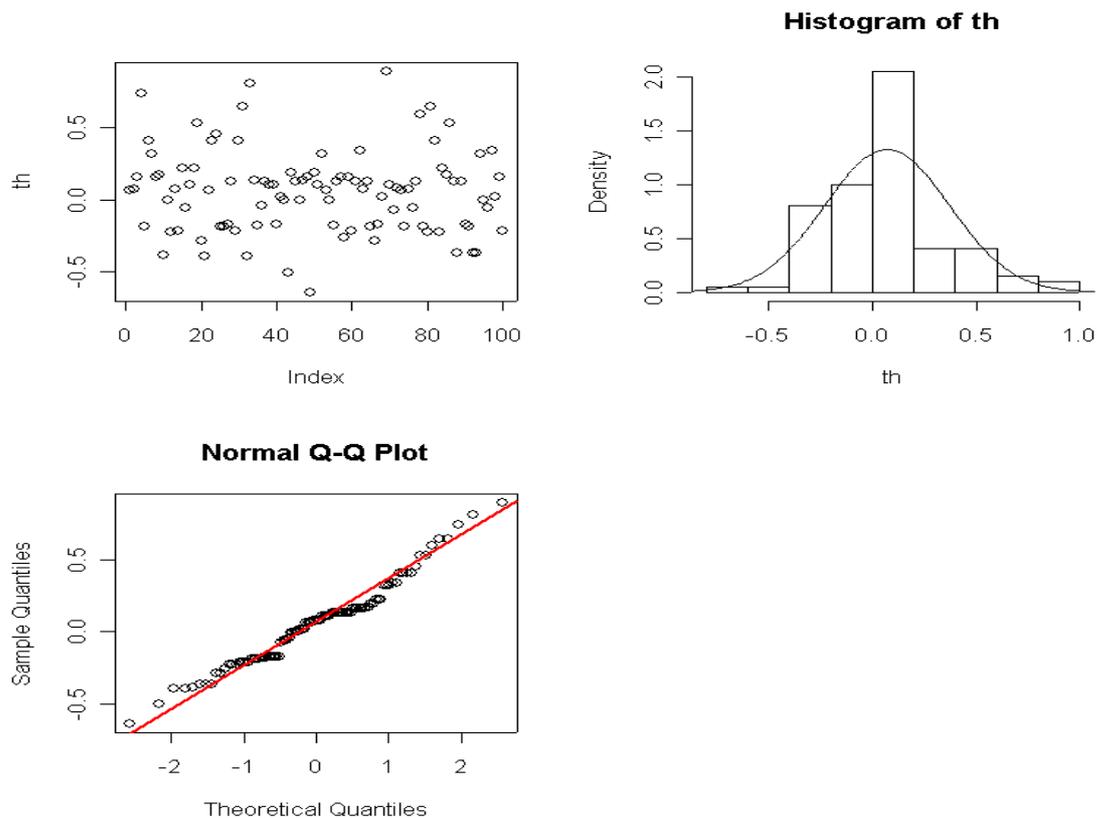


Figura 3.1

Il grafico in alto a sinistra rappresenta il diagramma di dispersione del campione ottenuto dalla distribuzione a posteriori ottenuta dalla simulazione con IS: da esso si evince, come la media delle simulazioni sia circa zero (ce l’aspettavamo dato che sia la distribuzione a priori che, il campione iniziale, hanno media pari a zero). La media è, infatti, un “compromesso” tra la media della a priori ( $\mu_0=0$ ) e la media del campione di partenza (in realtà uguale a 0.2): infatti si vede come questa sia poco più alta di zero. La densità (grafico in

alto a destra) ed il grafico quantile – quantile (a sinistra, in basso) ottenuti, mostrano come, si sia in realtà ancora abbastanza lontani dalla distribuzione teorica delle stime  $N(\mu^*, \tau^{*2})$ .

Come mostra chiaramente la figura 3.2, aumentando la numerosità campionaria della simulazione  $m$  (da 100 a 1000) e mantenendo invariate tutte le altre condizioni (distribuzione a priori, campione di partenza e valore di  $\sigma_0^2=1$ ) le cose cambiano:

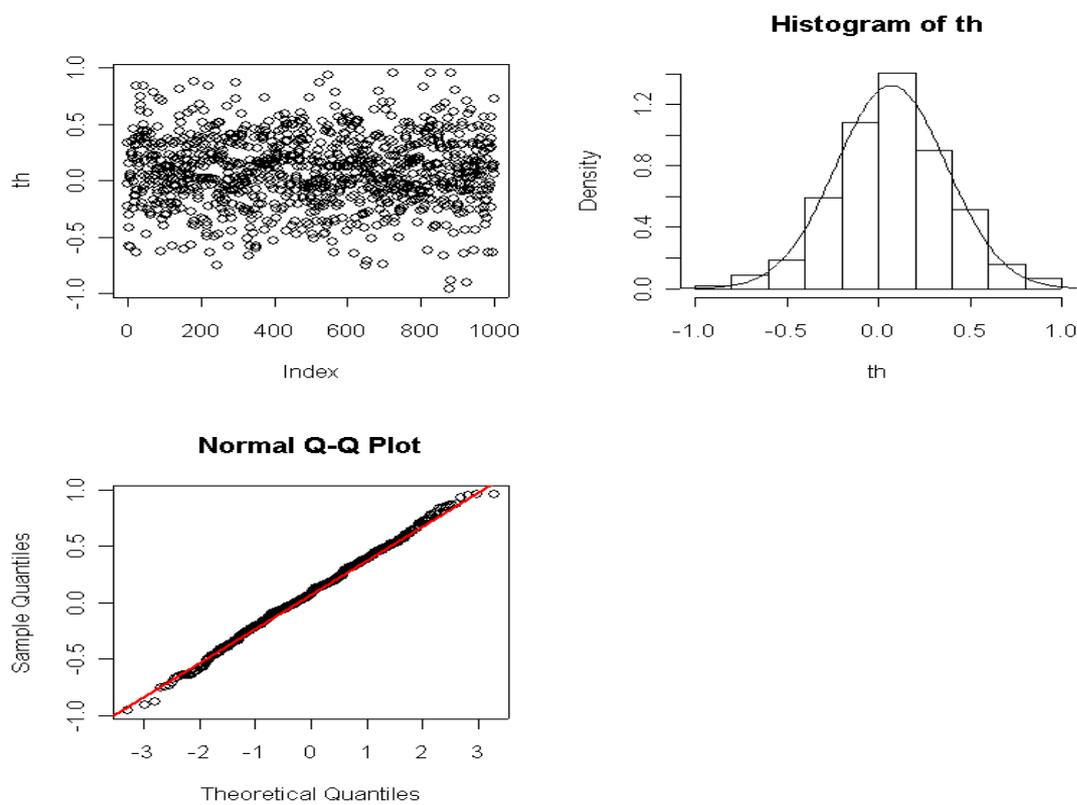


Figura 3.2

L'approssimazione del campione ottenuto alla distribuzione a posteriori nota, è quasi perfetta dato che sia l'istogramma che il qqplot approssimano bene una normale con media pari a  $\mu^*$  e varianza  $\tau^{*2}$ . Proviamo ora a vedere come si comportano MCMC e IIS a parità di condizioni iniziali.

### 3.1.2 Markov Chain of Monte Carlo

In MCMC, oltre a specificare  $f(\mu)$ ,  $\sigma_0^2=1$  e la  $L(\mu)$  come in IS qui sopra, dobbiamo specificare i valori di  $\epsilon$ , il quale, influenza sia la velocità di convergenza dell'algoritmo (più grande è e più rapida ne è la convergenza) che la dipendenza nella catena markoviana creata (maggiore è  $\epsilon$  e più forte è la dipendenza all'interno della catena); inoltre, dobbiamo inizializzare  $\mu_0$  (valore di partenza dell'algoritmo, più vicino è al vero valore di  $\mu^*$  e più rapida è la convergenza). Nel nostro caso li abbiamo posti, rispettivamente, pari a 0.5 e 3. Dobbiamo poi tener conto del burn-in della catena markoviana creata: nel nostro caso abbiamo simulato 120 valori provenienti da  $f(\mu | x)$ , eliminandone i primi 20 (la convergenza dell'algoritmo, è indicata in figura 3.3) ottenendo quindi, un campione "corretto" di numerosità pari a 100 (come rappresentato in figura 3.4):

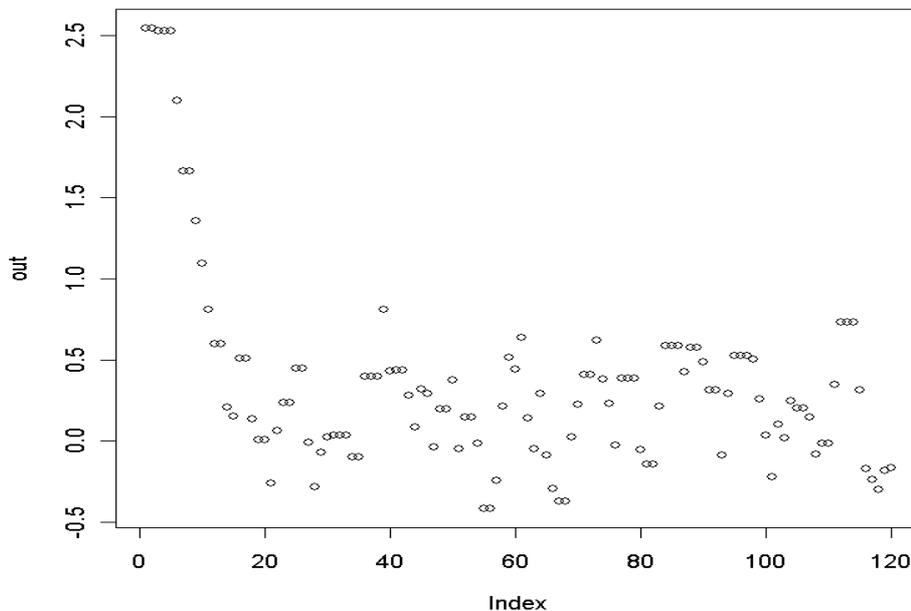


Figura 3.3

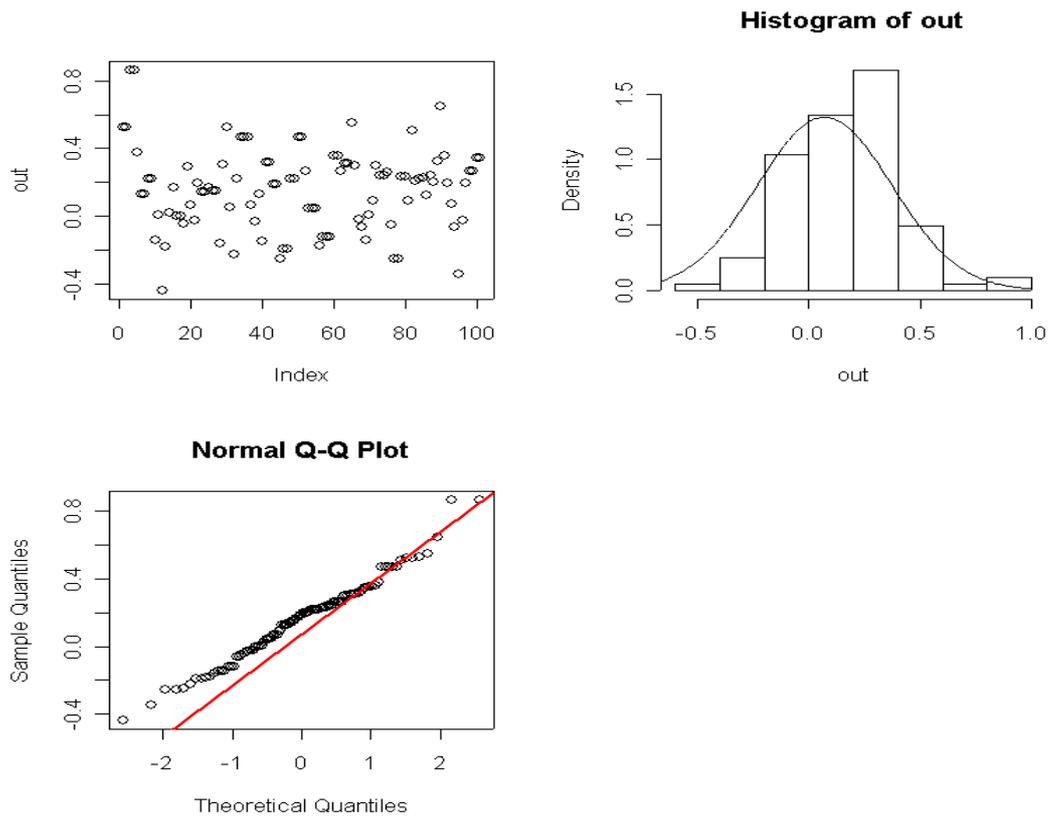


Figura 3.4

Anche in questo caso, l'approssimazione alla distribuzione teorica, non è delle migliori (anche se sembra peggiore di quella ottenuta con IS). Proviamo ad aumentare la numerosità campionaria ( 1100 valori simulati, ne togliamo 100 di burn-in...):

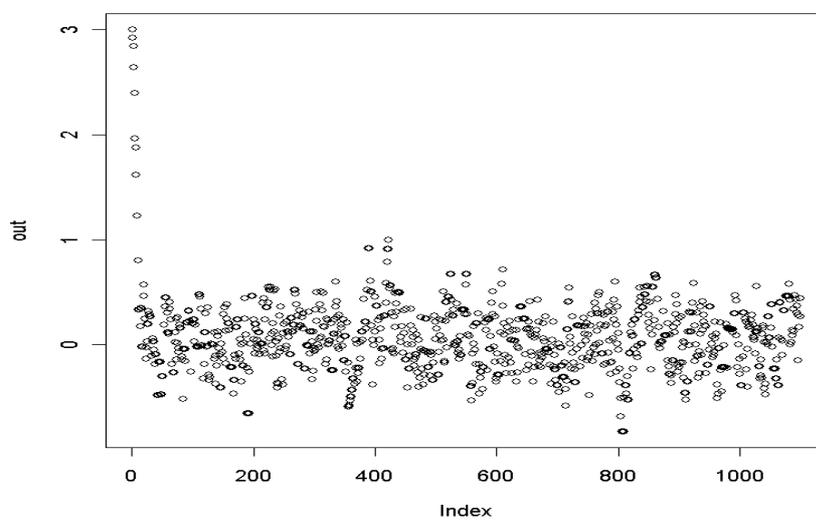


Figura 3.5

e vediamo che l'approssimazione a  $f(\mu | x)$  è molto buona (figure 3.5 e 3.6) anche se, forse, le stime ottenute con Importance Sampling, sembrano migliori.

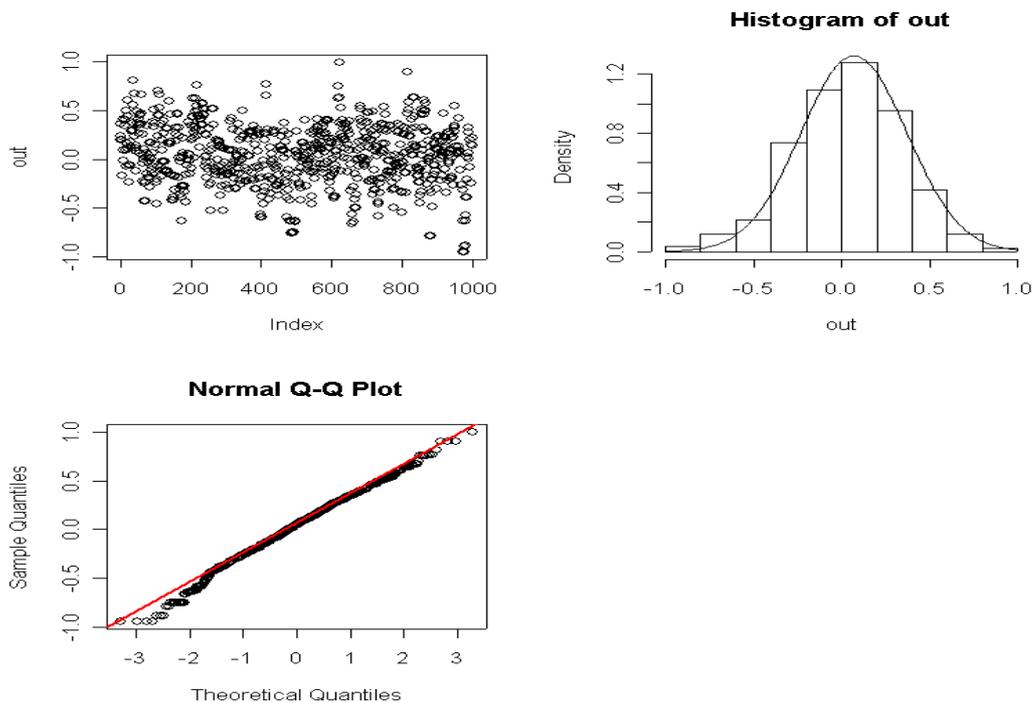


Figura 3.6

Non ci resta, ora, che vedere come si comporta IIS nella stessa situazione.

### 3.1.3 Iterative Importance Sampling

IIS, invece (per quanto spiegato nel capitolo due), a parità di condizioni iniziali, dovrebbe permettere di ottenere stime più efficienti dato che, prima, campiona una serie di valori per  $\mu_1, \mu_2, \dots, \mu_i, \dots, \mu_m$  (con  $i=1,2,\dots,100$ ) tramite Importance Sampling; e poi per  $N$  volte (il valore di  $N$  è del tutto arbitrario) si ricampiona avvicinandosi di  $\epsilon$  in  $\epsilon$  e ottenendo una stima della  $f(\mu | x)$  sempre più precisa. In Figura 3.7, riportiamo istogramma, qqplot (prima riga) e altri due grafici (seconda riga) che ci forniscono un'informazione importante. Nel primo in basso a sinistra, infatti, calcoliamo una sorta di media “cumulata” per dare un'idea sulla convergenza nelle stime della media (che indicheremo con

$\mu_1^\#, \mu_2^\#, \dots, \mu_{100}^\#$  ipotizzando che  $N=100$ ) alla “vera” media della distribuzione a posteriori ( $\mu^\circ$  indicata con una retta orizzontale e pari circa a 0.07) nel seguente modo: al primo passo avremo  $\mu_1^\#/1$ , al secondo  $\mu_1^\#+\mu_2^\#/2$ , al terzo  $\mu_1^\#+\mu_2^\#+\mu_3^\#/3$  fino al centesimo passo dove abbiamo  $(\sum_{i=1}^{100} \mu_i^\#)/100$ ; in quello in basso a destra riportiamo le medie stimate di campione in campione (per un totale di  $N=100$ ) della  $f(\mu | x)$   $\mu_1^\#, \mu_2^\#, \dots, \mu_{100}^\#$  (indicate con i pallini) confrontate con la “vera” media a posteriori (anche qui indicata con una retta orizzontale). In questo paragrafo utilizzeremo sempre questo tipo di rappresentazione per valutare la bontà delle simulazioni ottenute con IIS.

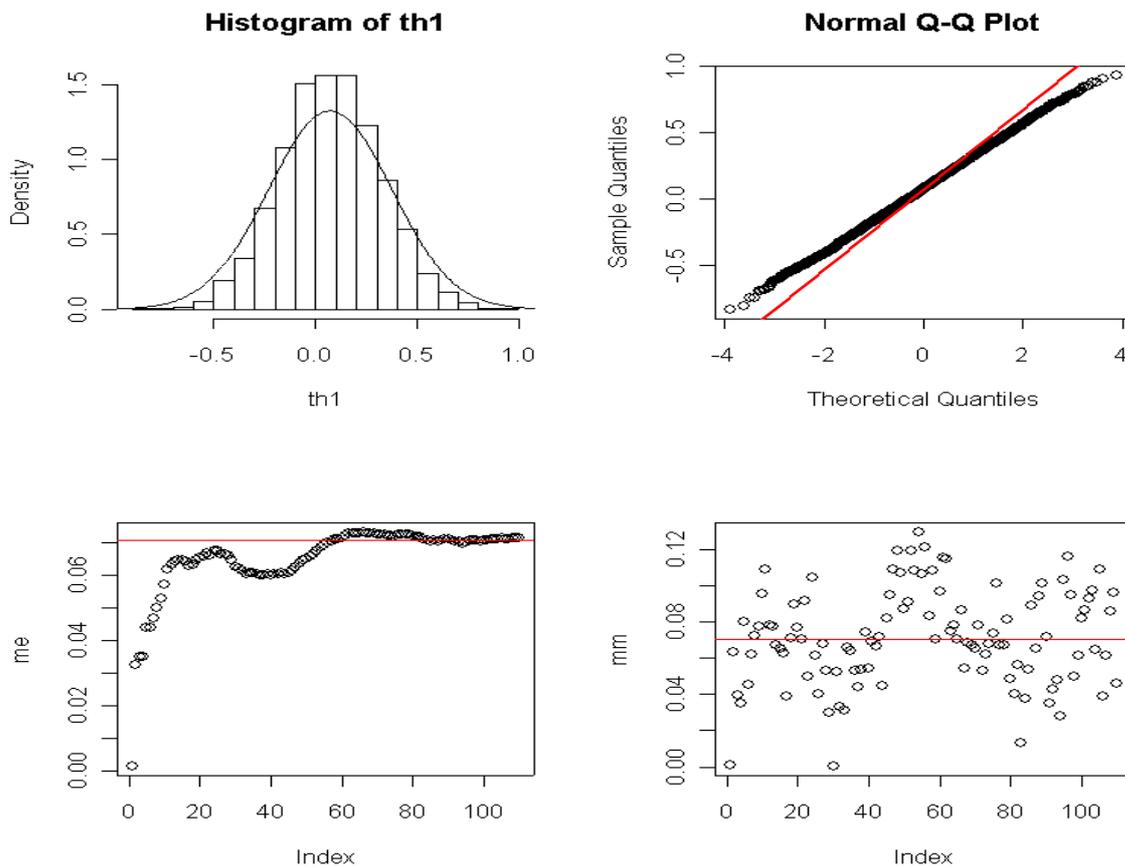


Figura 3.7

Ciò che si vede, è che i campioni, di volta in volta simulati, tendono a discostarsi, e di molto, dalla distribuzione a posteriori: si vede chiaramente nel

qqplot e nell'istogramma come le code non rispecchino per bene la “a posteriori”, e come, la media “cumulata” tenda alla media della posteriori  $\mu^\circ$  solo alla sessantesima iterazione. Per veder la differenza, rispetto ai due casi precedenti (IS e MCMC), abbiamo preso il campione creato all'iterazione  $N=100$  per vederne istogramma e qqplot (figura 3.7, appunto). Aumentando la numerosità campionaria delle simulazioni di volta in volta prodotte (portandola ad  $m=1000$ ) e tenendo costante il numero di iterazioni  $N$  (100), le cose non sembrano cambiare di molto...(migliorano solo i grafici della seconda riga di figura 3.8) . Infatti in figura 3.8, con  $N=100$  abbiamo...

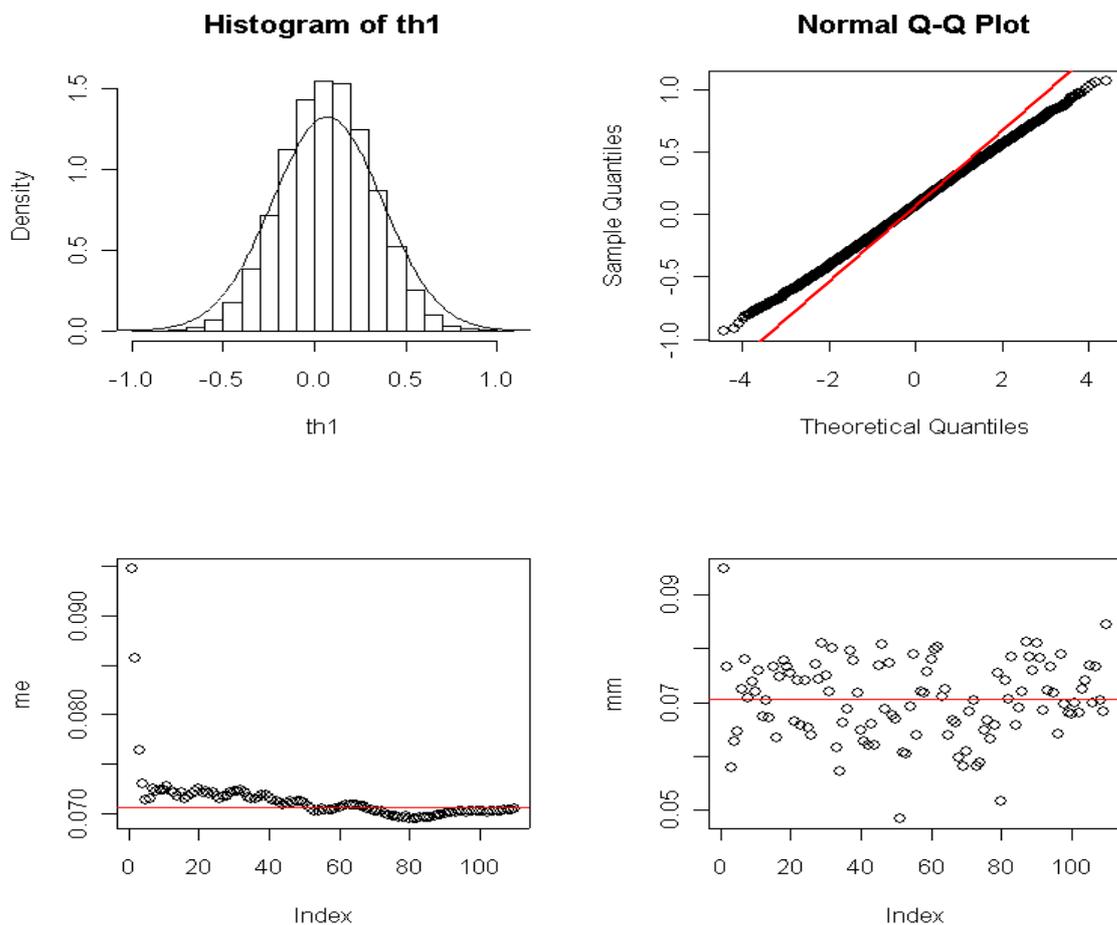


Figura 3.8

Ora proviamo ad aumentare il numero di iterazioni, portando  $N$  uguale a mille... e lasciando  $m$  anch'esso pari a 1000 (figura 3.9, qui sotto):

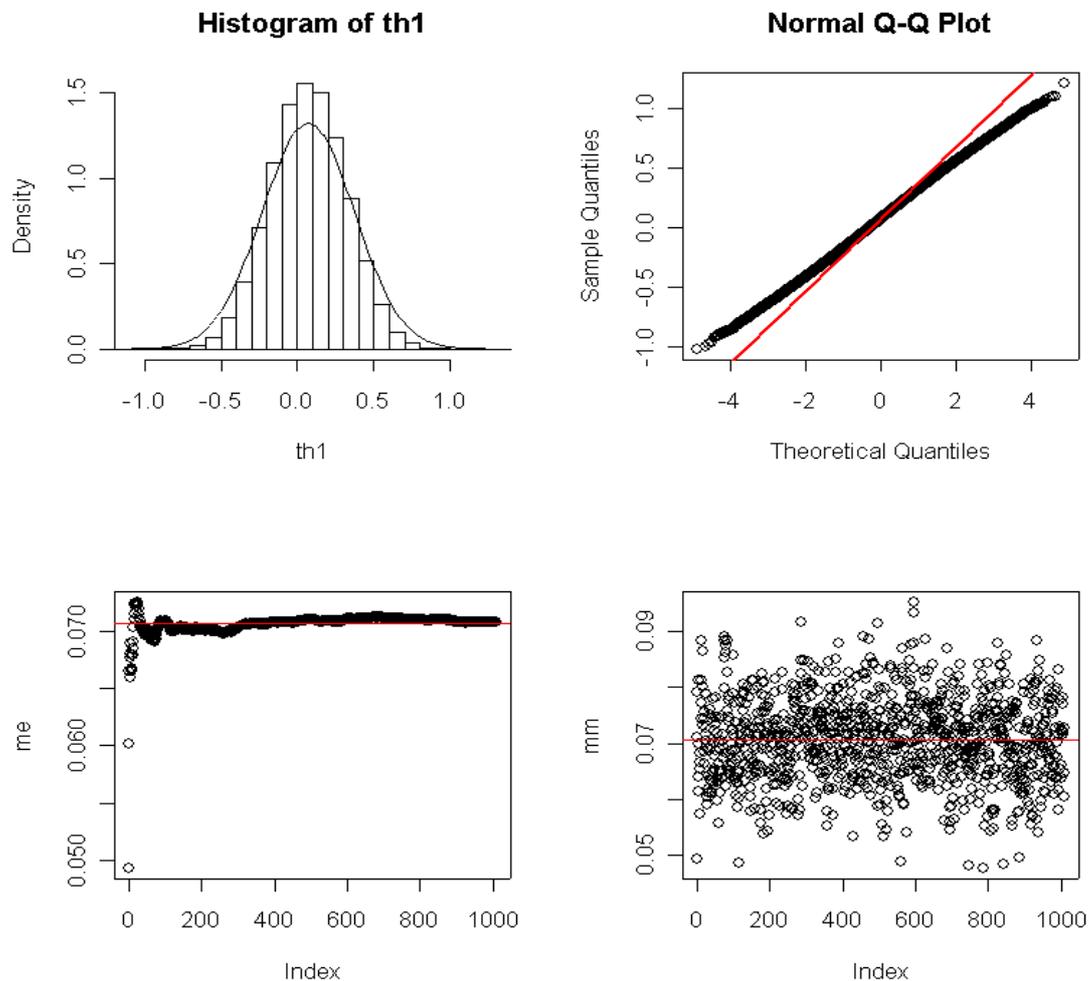


Figura 3.9

La convergenza, sembra andar abbastanza bene, ma il campione simulato dalla “a posteriori” continua ad essere abbastanza distorto. Che cosa succede? Molto probabilmente, il problema per Iterative Importance Sampling, stà nella numerosità campionaria del campione iniziale fin qui utilizzato ( $n=10$ ). Proviamo con un  $n=100$ . Come si vede in figura 3.10, ora l’ approssimazione ottenuta sembra essere molto buona anche con solo  $N=100$  iterazioni e campioni di numerosità  $m=1000$  prodotti di volta in volta.

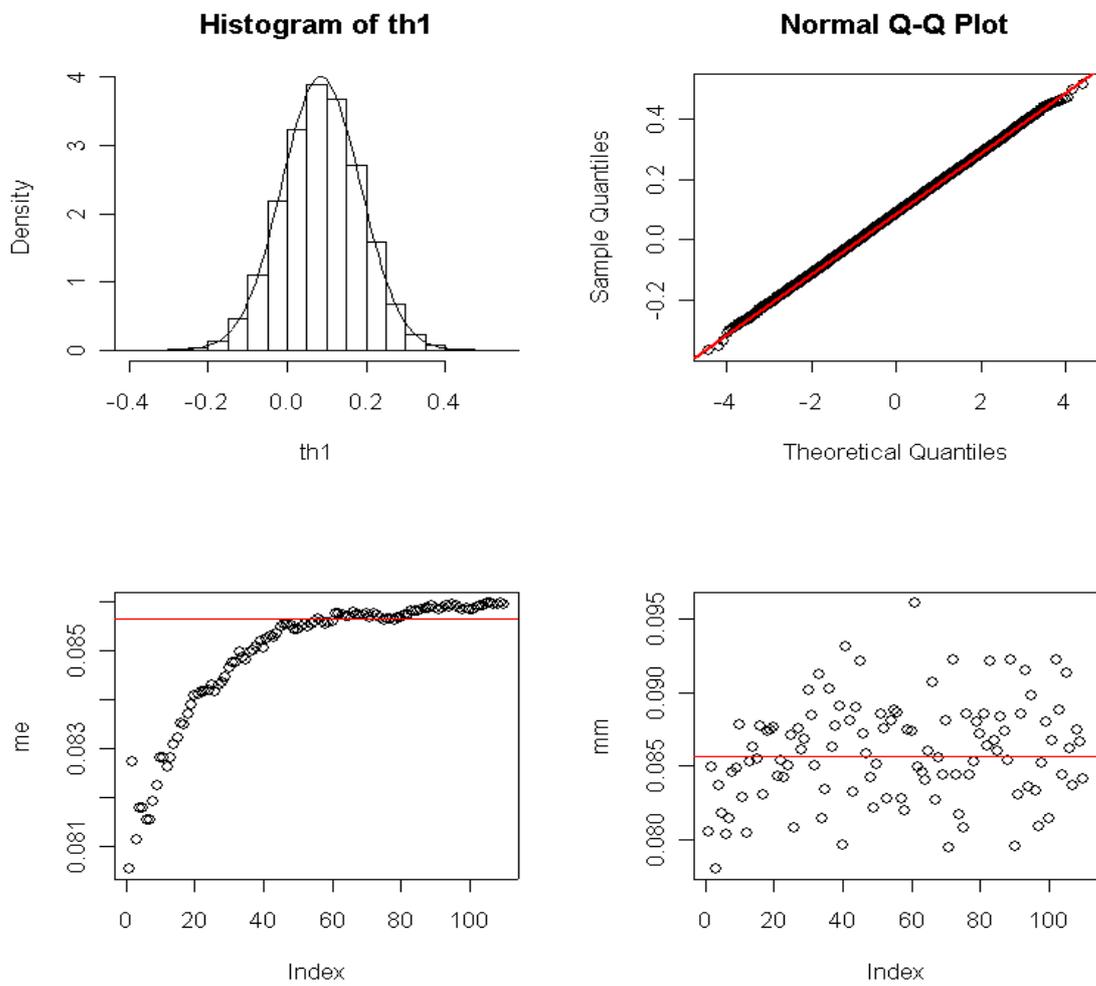


Figura 3.10

Aumentando poi anche N (portandolo a 1000) si ha (figura 3.11)...

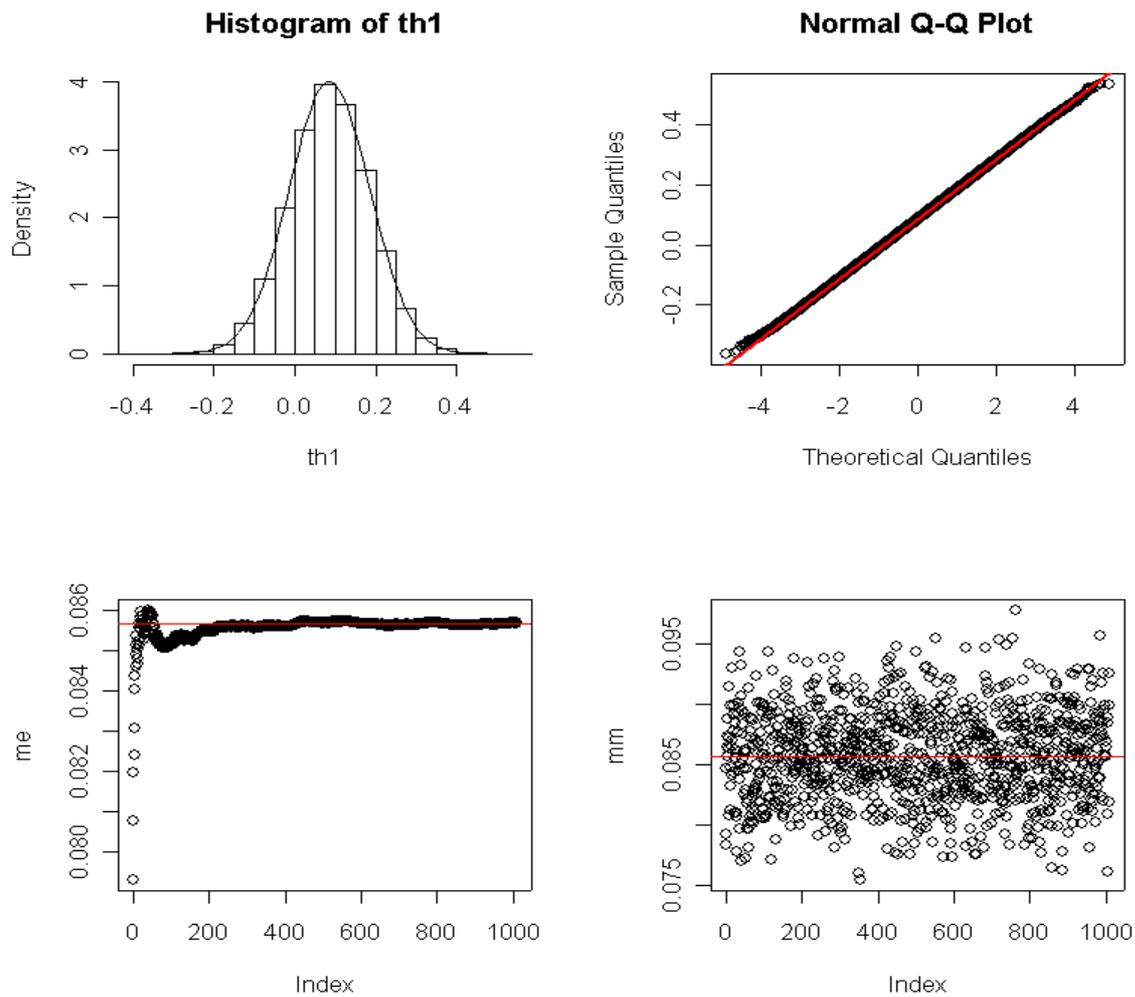


Figura 3.11

L'approssimazione alla  $f(\mu | x)$  sembra essere ottima e la convergenza ad una catena markoviana della media a posteriori della distribuzione avviene circa dopo la centesima iterazione.

Un'analisi di questo genere, non è però sufficiente per valutare quale algoritmo funzioni meglio; non ci permette poi, di controllare come cambi la bontà della simulazione al variare delle scelte iniziali (distribuzione a priori, numerosità del campione iniziale...). Nel paragrafo successivo, sarà introdotto proprio un indice (l'Errore Quadratico Medio) che ci permetterà di valutare la bontà della simulazione.

### 3.1.4 Valutazione della bontà degli algoritmi di simulazione tramite EQM

L'Errore Quadratico Medio (EQM) è una semplice statistica descrittiva che viene utilizzata in diversi ambiti di lavoro per cercare di quantificare l'errore che, di volta in volta, si commette. Ad esempio, quando in un approccio a serie storiche si calcolano le previsioni in relazione ad un certo fenomeno (ad esempio la previsione nelle vendite di un prodotto sul mercato) una quantificazione dell'errore di previsione commesso è necessaria per capire di quanto ci si sbaglia. Così anche in ambiti bayesiani, l'EQM è un buon indice per valutare la precisione delle stime ottenute. Inoltre, è molto semplice da calcolare. Per il caso normale-normale con  $\sigma_0^2$  costante nota, ha senso calcolare l'EQM per i parametri della distribuzione a posteriori che, ricordiamo, sono di una  $N(\mu^*, \tau^{*2})$ . L'EQM per il problema in questione, viene calcolato come:

$$\text{EQM}(E(\mu)) = (1/N) \sum_i (E_i^{\wedge}(\mu) - E(\mu))^2, \quad \text{EQM}(\text{Var}(\mu)) = (1/N) \sum_i (\text{Var}_i^{\wedge}(\mu) - \text{Var}(\mu))^2;$$

dove  $E(\mu)$  e  $\text{Var}(\mu)$  sono rispettivamente, vere media e varianza della media della distribuzione a posteriori;  $E_i^{\wedge}(\mu)$  e  $\text{Var}_i^{\wedge}(\mu)$  sono medie e varianze stimate di campione in campione, e  $N$  il numero di campioni della “a posteriori” creati. Ora valuteremo, come cambia “il valore” dell'EQM al variare delle caratteristiche di base dei diversi algoritmi. Proporranno una serie di situazioni parametriche e vedremo come cambi la bontà delle simulazioni dalla “a posteriori” al variare del metodo e delle componenti essenziali dell'algoritmo. Nelle seguenti situazioni, indicheremo con  $f(\mu)$  la distribuzione a priori; con  $f(x | \mu)$  l'informazione campionaria espressa come dati  $x_1, x_2, \dots, x_n$  e rappresentata tramite la funzione di verosimiglianza (utilizzeremo, di volta in volta, diversi campioni della  $f(x | \mu)$ );  $f(\mu | x)$  la distribuzione a posteriori e il

campione ottenuto di volta in volta da essa tramite IS, MCMC e IIS è del tipo  $\mu_1, \mu_2, \dots, \mu_m$ ; con  $N$  il numero di campioni prodotti della “a posteriori”;  $\varepsilon$  e  $\mu_0$  saranno, rispettivamente, l’ampiezza del semi intervallo utilizzato per ricampionare (sia in IIS che MCMC), e il valore d’inizializzazione di MCMC.

### Situazione 1

Iniziamo con una situazione “povera” nella quale abbiamo a disposizione solo 10 osservazioni provenienti da una  $N(0,1)$  (la nostra  $f(x | \mu)$ ); poi ipotizziamo che la distribuzione a priori sia del tipo  $\mu \sim N(0,1)$  e di esser interessati a simulare ogni volta  $m=100$  valori dalla distribuzione a posteriori  $f(\mu | x)$ ; Ripeteremo  $N=1000$  volte la simulazione dalla distribuzione a posteriori; solamente per IS e MCMC fissiamo  $\varepsilon=0.5$  e  $\mu_0=3$ . Le stime degli EQM per media e varianza di  $\mu$ , ottenuti sono rappresentati in figura 3.12 (ricordiamo che per MCMC abbiamo simulato di volta in volta 120 valori e tolto 20 valori di burn in);

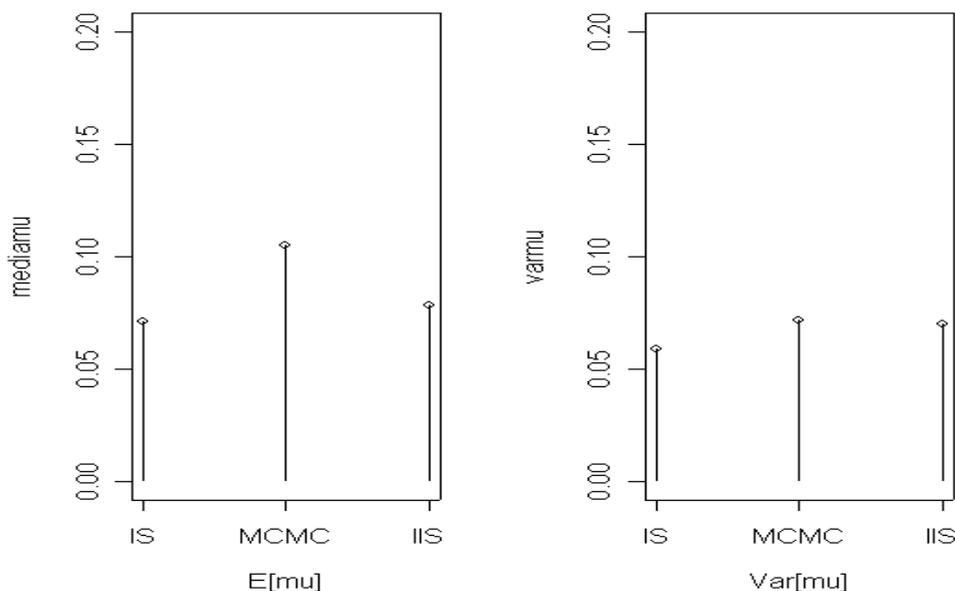


Figura 3.12

In questa situazione, IS sembra funzionare decisamente meglio sia rispetto ad IIS che MCMC: infatti esso presenta gli EQM più piccoli (maggiore è l'altezza delle barre e maggiore è l'errore commesso nella stima) tra i tre metodi. Si osservi che per rendere interpretabile il grafico abbiamo riparametrizzato l'EQM commesso sia per la media che per la varianza delle stime prendendo, rispettivamente, le quantità  $(1/(-\log(\text{EQM}(E(\mu))))$  e  $(1/(-\log(\text{EQM}(\text{Var}(\mu))))$ .

### Situazione 2

Proviamo ora ad aumentare la numerosità del campione della distribuzione a posteriori ( $m=1000$ ) di volta in volta ottenuto tenendo invariate tutte le altre caratteristiche definite nella situazione 1.

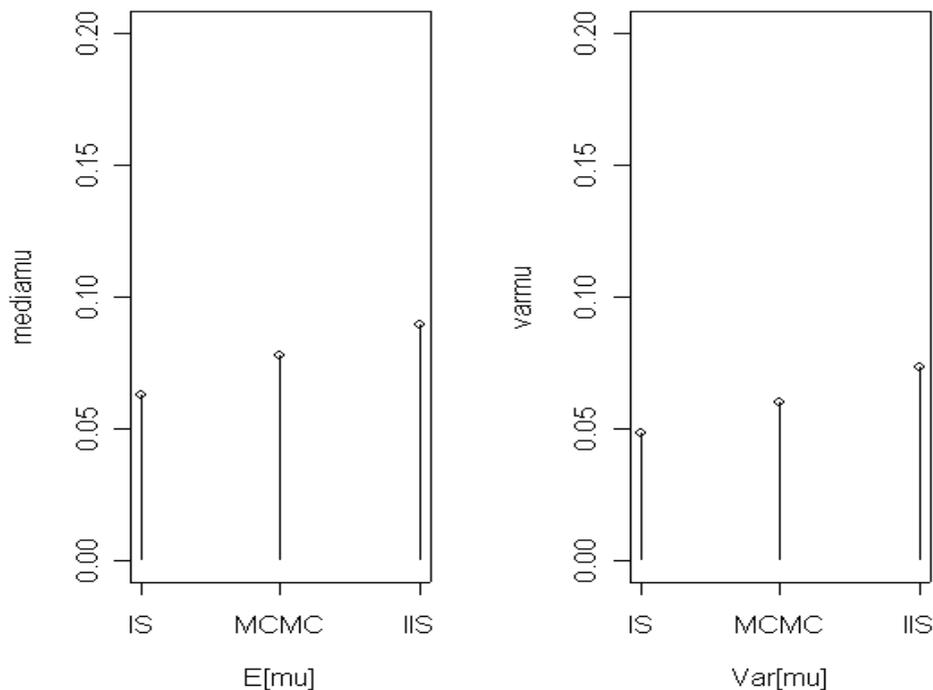


Figura 3.13

Importance Sampling sembra esser sempre il metodo migliore anche aumentando la numerosità campionaria  $m$  della simulazione. Ma perché IS sembra essere sempre l'algoritmo migliore? E perché IIS e MCMC non sembrano funzionare altrettanto bene? Per capirlo, vi rimandiamo alle prossime situazioni.

### Situazione 3

Ipotizziamo ora, di aver a nostra disposizione un campione della  $f(x|\mu)$  più ricco, e di numerosità  $n=100$ . Solite condizioni iniziali  $N=1000$ ,  $m=1000$  (1100 per MCMC),  $\mu \sim N(0,1)$ ,  $\varepsilon=0.5$  e  $\mu_0=3$  (ricordando di togliere un burn-in di numerosità pari a circa 100).

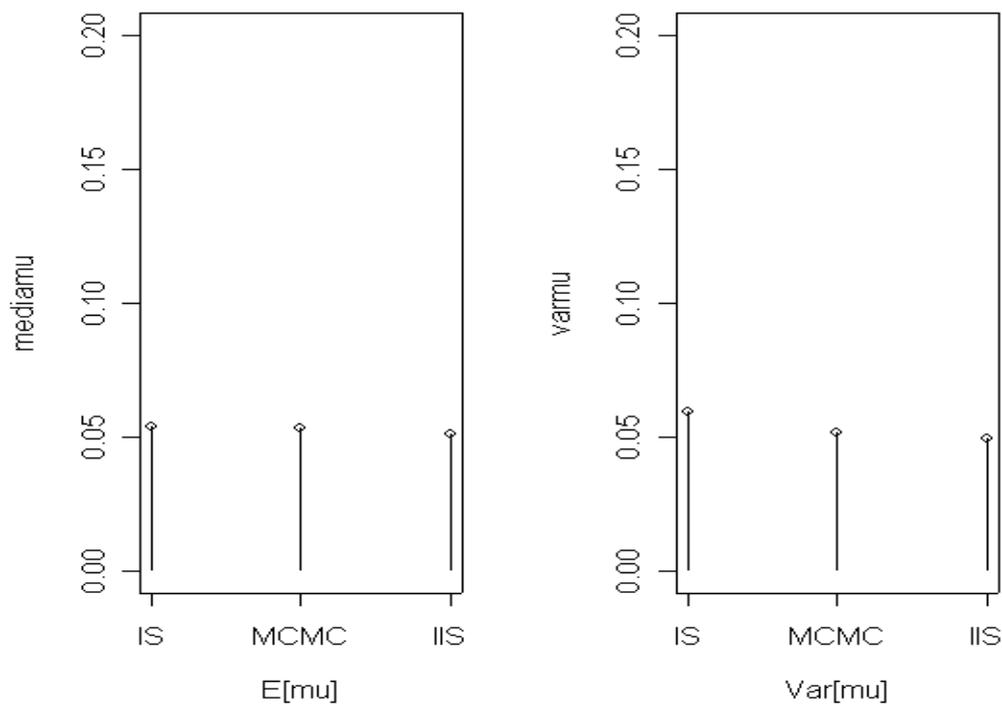


Figura 3.14

Le cose cambiano, e anche di molto (si veda figura 3.14) quando l'informazione proveniente dai dati ha numerosità pari a 100. Ora è IIS che funziona meglio e dà stime più precise (sia per quanto riguarda la media che la varianza). MCMC e IS invece, seppur diano stime rappresentative della “a posteriori” ne aumentano la variabilità rispetto ad IIS.

#### Situazione 4

Proviamo a cambiare la distribuzione a priori iniziale per  $\mu$ , passando ad una  $N(0,100)$ . La scelta di una varianza più elevata è più che lecita dato che, in molte situazioni, la distribuzione a priori  $f(\mu)$  è ignota e nella prassi viene ipotizzata una varianza più elevata per cercar di “controllare” il fenomeno. Cambiando, quindi la distribuzione a priori in  $\mu \sim N(0,100)$  e mantenendo come fatto nella seconda situazione  $m$  ed  $N$  uguali a 1000;  $n=10$  con  $f(x | \mu)$  simulati sempre da una  $N(0,1)$  con  $\varepsilon=0.5$  e  $\mu_0=3$ . In figura 3.15 riportiamo gli EQM, ottenuti nelle condizioni poco più sopra specificate:

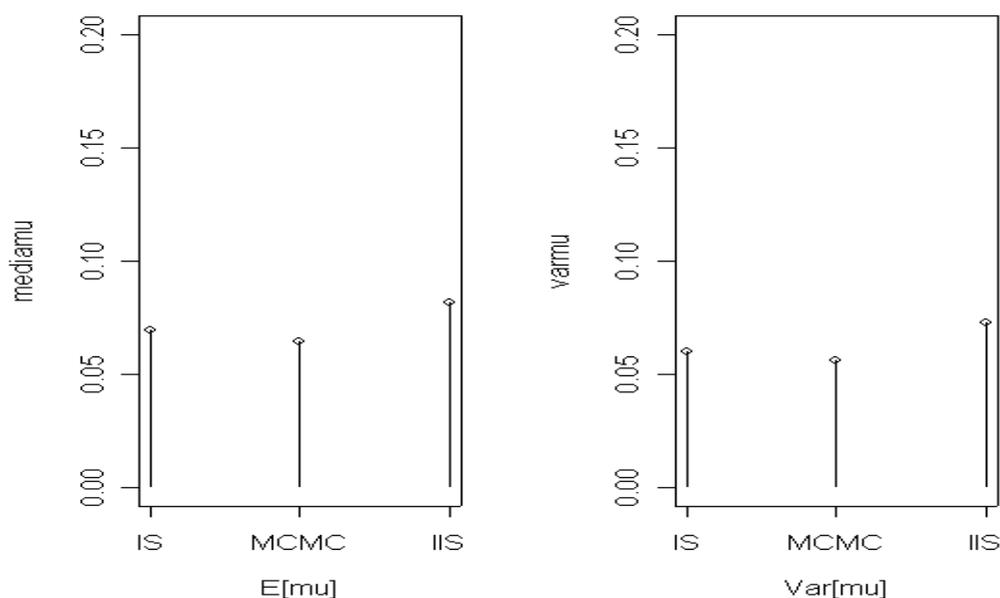


Figura 3.15

Data la scarsa numerosità campionaria iniziale IIS continua a dare gli EQM più elevati. IS e MCMC, sembrano migliori in questa situazione.

### Situazione 5

Stesse condizioni della situazione quattro, a parte il fatto che definiamo una distribuzione “a priori” con varianza più elevata e pari a 1000. Quindi  $\mu \sim N(0,1000)$ . In figura 3.16, riportiamo le stime ottenute:

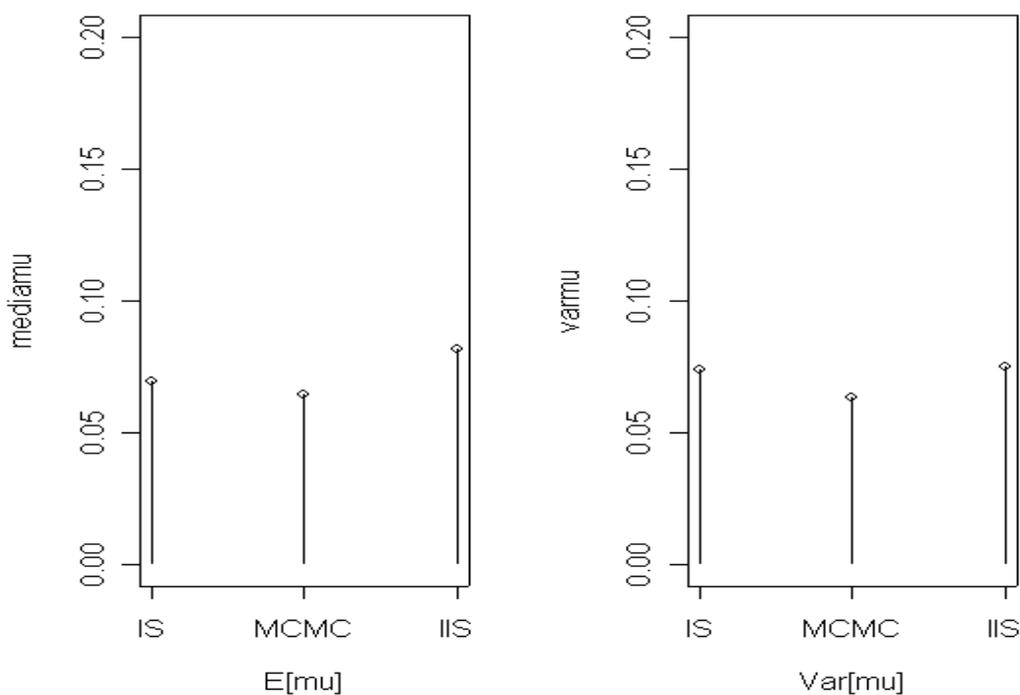


Figura 3.16

Ciò che si nota è che MCMC sembra essere il metodo migliore per questa situazione a varianza a priori elevata, dato che: IS “soffre” quando la distribuzione a priori non rispecchia l’andamento della “a posteriori”, mentre IIS non funziona bene sia per la scarsa numerosità campionaria ( $n=10$ ) che per

il fatto che, al primo stadio, opera anch'esso un Importance Sampling. Nella prossima "situazione" proveremo ad incrementare nuovamente la numerosità campionaria passando ad  $n=100$ . Che cosa ci aspettiamo...? Che IIS, soprattutto, migliori le proprie stime!

### Situazione 6

Tutto come nella situazione 5, a parte un incremento della numerosità (da 10 a 100) del campione tratto dalla  $f(x | \mu)$ .

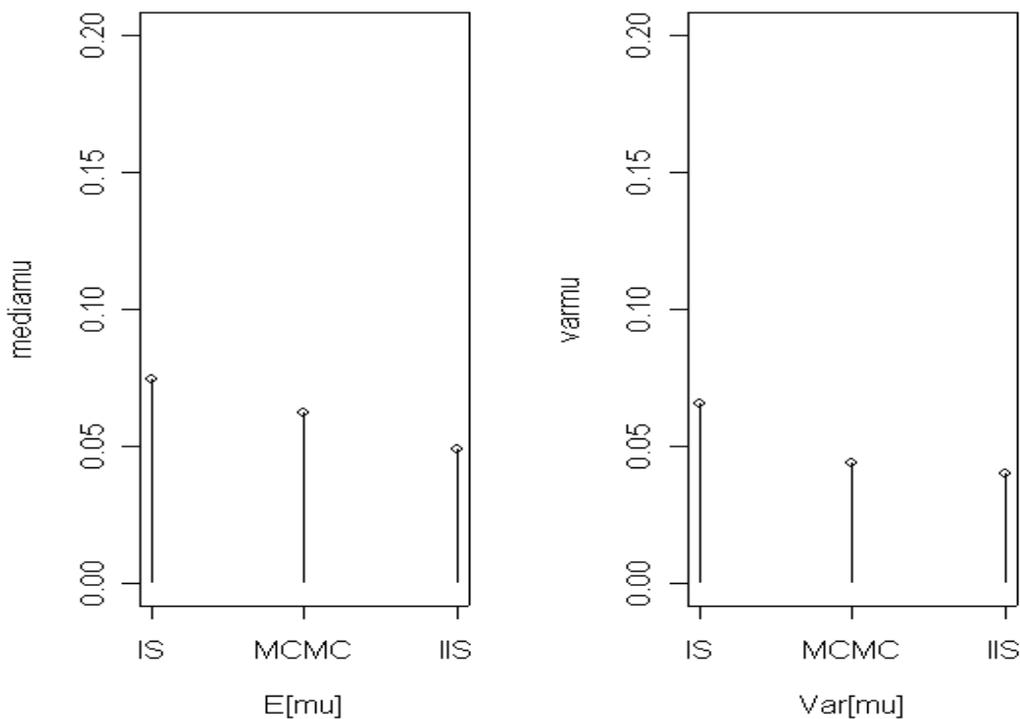


Figura 3.17

Come auspicato, le stime con IIS migliorano, diventando anche più efficienti rispetto a MCMC. Il campione ottenuto con IS rimane, comunque, il peggiore visto quanto spiegato nella situazione precedente.

## Situazione 7

Non ci resta ora che valutare, se e come, cambiano le simulazioni con i tre metodi modificando i valori di  $\epsilon$  e  $\mu_0$  (prendendo come situazione di partenza la numero tre). In figura 3.18 riportiamo le stime ottenute con, rispettivamente, MCMC (prima riga) ed IIS (seconda riga) provando tre diversi valori per  $\epsilon$  (0.3, 0.5 e 0.7):

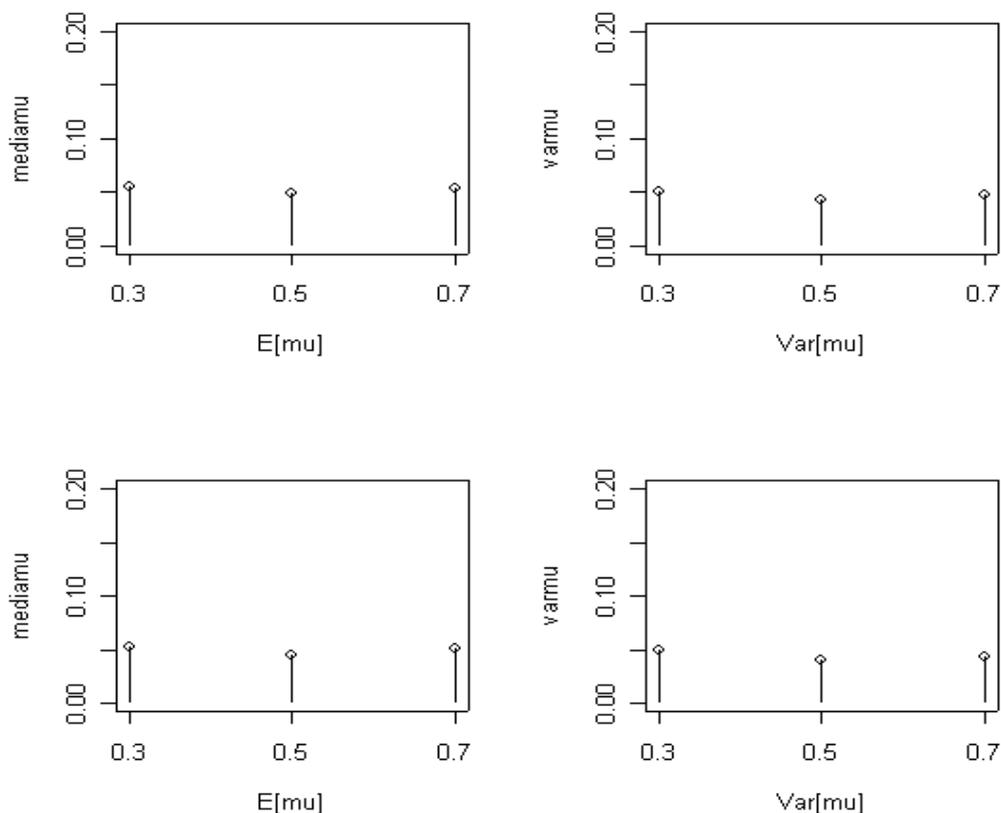


Figura 3.18

Sia per MCMC che IIS, gli Errori Quadratici Medi inferiori si hanno in corrispondenza di un valore di  $\epsilon$  pari a 0.5 (quello mediano, quindi). Questo molto probabilmente è dovuto al fatto che 0.5 rappresenta una sorta di compromesso tra la velocità di convergenza dell'algoritmo e tra la "quota" di dipendenza nella serie dei parametri creati (come già detto in precedenza,

infatti, con un valore di  $\varepsilon$  elevato il “tempo” di convergenza e la dipendenza nella serie dei parametri creati sono più grandi)

Per quanto riguarda la specificazione dei valori iniziali per  $\mu_0$  (presenti nel solo MCMC) più il valore iniziale è vicino al nucleo della distribuzione “a posteriori”, e minore è l’errore commesso dall’algoritmo. In figura 3.19, si nota come al crescere del valore iniziale specificato per  $\mu_0$  (abbiamo ipotizzato i valori 2, 5 e 9), maggiore è l’EQM commesso sia in riferimento alla vera media che alla vera varianza.

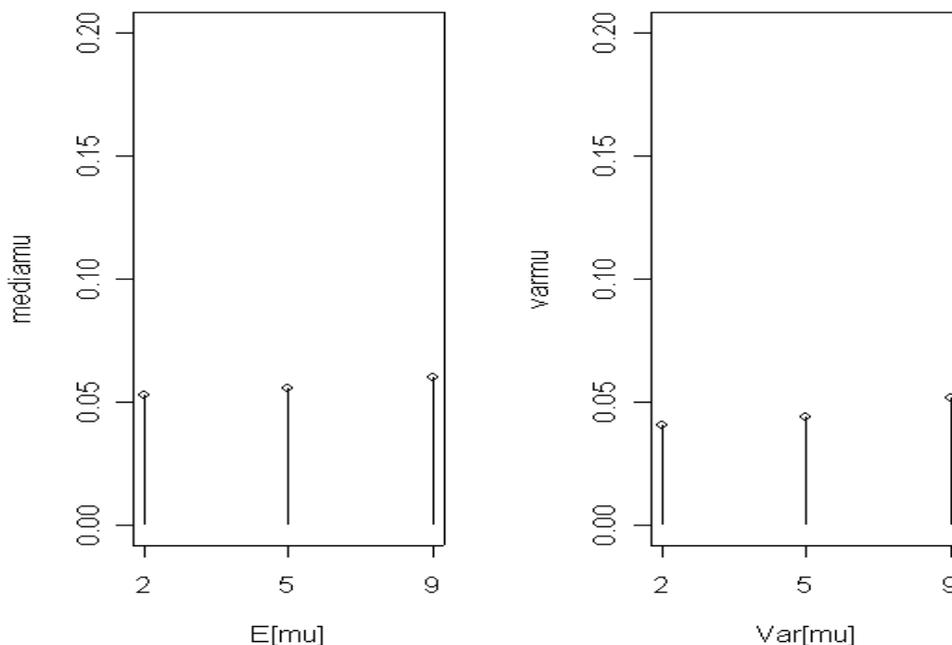


Figura 3.19

### 3.1.5 Conclusioni

Al termine dell’analisi appena effettuata, emerge come non ci sia un metodo che venga univocamente preferito agli altri come bontà delle stime. Infatti, Importance Sampling sembra funzionare meglio quando l’informazione iniziale proveniente dai dati è scarsa. Iterative Importance Sampling, invece,

sembra essere più adatto quando l'informazione campionaria iniziale è maggiore. MCMC, infine, sembra essere migliore nelle situazioni "esplorative" rispetto a IS: cioè quando, la distribuzione a priori è non nota (e queste sono le più frequenti, nella pratica) e si ipotizza una varianza più elevata. Abbiamo che IIS lavora comunque meglio in queste situazioni rispetto a MCMC se l'informazione campionaria è cospicua.

Tutto questo fin qui visto, può valere anche in altre situazioni bayesiane uniparametriche (uniparametrico perché c'è un parametro non noto, gli altri vengono presi come noti), tipo il "matrimonio" Poisson-Gamma o quello Esponenziale-Gamma. Nel prossimo paragrafo il nostro obiettivo sarà, valutare, se quanto visto nel caso uniparametrico, vale anche nel caso biparametrico.

### 3.2 Specificazione del modello normale con media e varianza ignote.

Consideriamo ora un caso in cui il parametro da stimare risulti bidimensionale. Siano  $X_1, X_2, \dots, X_n$   $n$  osservazioni indipendenti con distribuzione  $N(\theta, \sigma^2)$  e assumiamo che entrambi i parametri siano non noti. La verosimiglianza associata all'effettivo campione osservato  $x = (x_1, x_2, \dots, x_n)$  è così calcolabile:

$$\begin{aligned}
 L(\theta, \sigma^2) &= f(x_1, x_2, \dots, x_n | \theta, \sigma^2) \\
 &\propto (1/\sigma^n) \exp \left\{ -(1/2\sigma^2) \sum (x_i - \theta)^2 \right\} \\
 &\propto (1/\sigma^n) \exp \left\{ -(1/2\sigma^2) \sum (x_i - \bar{x} + \bar{x} - \theta)^2 \right\} \\
 &\propto ((1/\sigma^2)^{n/2}) \exp \left\{ -(n/2\sigma^2) [s^2 + (\bar{x} - \theta)^2] \right\}
 \end{aligned}$$

dove  $s^2$  rappresenta la varianza campionaria (non corretta). Per motivi computazionali è conveniente modificare la parametrizzazione consueta e porre  $\sigma^2 = 1/\tau$ . Il parametro  $\tau$  viene di solito chiamato precisione. In tal modo la verosimiglianza diventa

$$L(\theta, \tau) \propto \tau^{n/2} \exp \{ -(n\tau/2) [s^2 + (\bar{x} - \theta)^2] \}$$

Una distribuzione bivariata che viene spesso utilizzata in questo contesto è la *Normale – Gamma* che denoteremo con il simbolo  $NG(a, b, c, d)$ . L'uso di tale distribuzione implica che, a priori:

$$\tau \sim Ga(c, d) \quad \text{e} \quad \mu | \tau \sim N(a, 1/b\tau)$$

con media e varianza a priori di  $\tau$  pari a, rispettivamente,  $E(\tau)=c/d$ ,  $Var(\tau)=c/d^2$  e media e varianza a priori per  $\mu$ , calcolate dalla marginale  $f(\mu) = \int f(\mu | \tau) f(\tau) d\tau$ , e pari a  $E(\mu)=a$  e  $Var(\mu)=d/(b(c-1))$  con  $c>1$ .

Moltiplicando la verosimiglianza per la distribuzione a priori si ottiene che:

$$(i) \quad f(\mu, \tau | x) \propto L(\mu, \tau) f(\mu | \tau) f(\tau)$$

Dove, si dimostra, che il nucleo della distribuzione ottenuta calcolando (i) è ancora quello di una *Normale – Gamma*; Più precisamente, la distribuzione finale di  $(\mu, \tau)$  è  $NG(a^*, b^*, c^*, d^*)$  dove:

$$a^* = (n\bar{x} + b a)/(n+b) \quad b^* = b + n$$

$$c^* = c + (n s^2)/2 + (n b (\bar{x} - a)^2 / 2(n+b)) \quad d^* = d + n/2$$

e la distribuzione finale marginale di  $\mu$  è  $((d^* b^*/c^*)(\mu - a^*))^{1/2} \sim t$  con  $2d^*$  gradi di libertà (per  $\tau$ , la gamma). Quest'ultimo risultato, è molto importante, dato che, essendo la distribuzione  $t$  tabulata in tutti i pacchetti statistici, ci permette di dare diretta interpretazione ai risultati ottenuti.

Una volta detto questo non ci resta che “provare” quanto fatto nel paragrafo precedente e “controllare” come si comportano i tre algoritmi oggetto di analisi.

### 3.2.1 Importance Sampling

Ipotizziamo di aver un campione di numerosità pari a 10 espresso dalla funzione di verosimiglianza  $L(\mu, \tau)$  (ipotizziamo che sia tratto dalla “solita” normale standardizzata). Poi, consideriamo che le distribuzioni a priori per  $\mu$  e  $\tau$  siano del tipo  $\tau \sim Ga(c, d)$  e  $\mu | \tau \sim N(a, 1/b\tau)$  con  $a=0$ ,  $b=0.5$ ,  $c=2$  e  $d=2$  e di esser interessati a simulare un campione di numerosità  $m=100$  dalla distribuzione a posteriori, ottenuta implementando la (i),  $f(\mu, \tau|x)$ . In figura 3.20, presentiamo i grafici quantile-quantile ottenuti con Importance Sampling per, rispettivamente,  $\mu$  (trasformazione necessaria per ricondurci alla distribuzione marginale di  $\mu$ , che è una  $t_{2d^*}$ ) e  $\tau$  (che ha distribuzione marginale, come già visto più sopra,  $\text{Gamma}(c^*, d^*)$ )

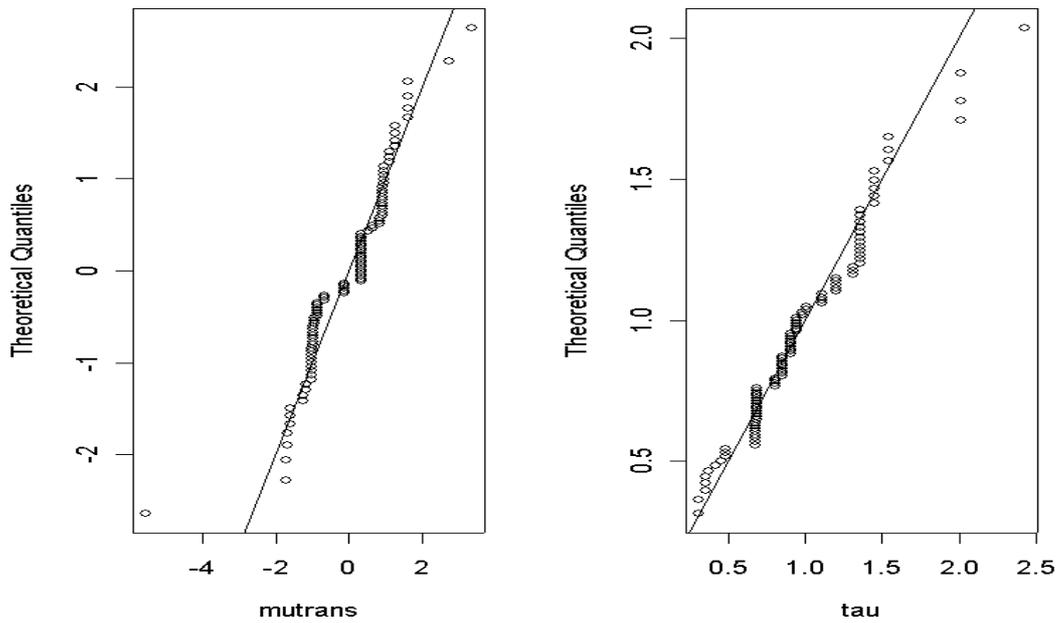


Figura 3.20

Nonostante le basse numerosità campionarie iniziali le simulazioni della a posteriori ottenute, non sono molto lontane dai quantici teorici (per le scelte effettuate, mutrans pari a zero e tau uguale ad uno). Proviamo ora a vedere cosa cambia incrementando il valore di  $m$  e portandolo a 1000.

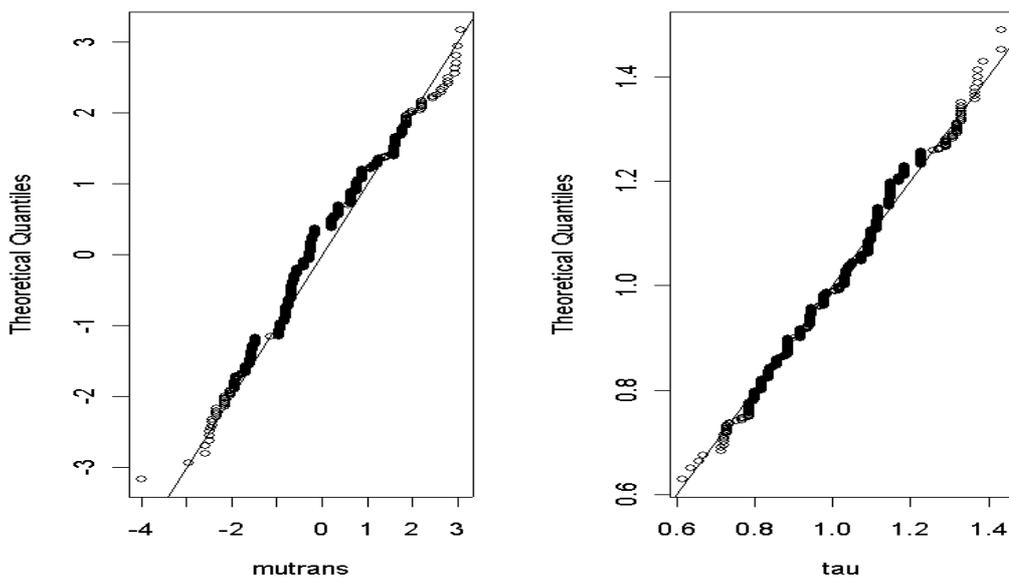


Figura 3.21

In figura 3.21, vediamo come le cose non sembrano cambiare molto, abbiamo più dati simulati dalla “a posteriori” ma, l’andamento di fondo sembra non cambiare. Provando ad aumentare la numerosità campionaria iniziale come, peraltro, già fatto nel paragrafo precedente, passando quindi ad  $n=100$  (mantenendo, comunque,  $m=1000$ ) le cose migliorano ulteriormente. Guardando figura 3.22 si noti come i quantili empirici sembrano rispecchiare quasi perfettamente quelli teorici (e questo per entrambi i parametri).

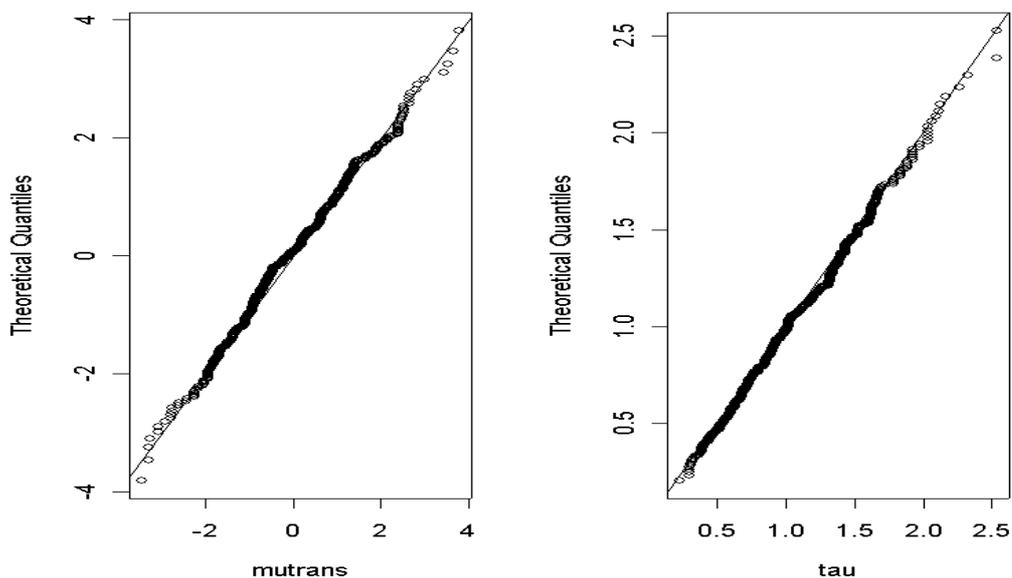


Figura 3.22

Proviamo ora a vedere come si comportano gli altri due algoritmi di simulazione in situazioni di questo tipo.

### 3.2.2 Markov Chain of Monte Carlo

Torniamo alla situazione iniziale prevista per IS nel paragrafo 3.2.1: stesse distribuzioni a priori e stesso campione iniziale tratto dalla  $f(x|\mu,\tau)\sim\mathcal{N}(0,1)$  di numerosità pari a 10. In più, per MCMC, dobbiamo

specificare due valori di partenza per l’algoritmo ( $\mu_0=2$  e  $\tau_0=2$ ) e due passi di ricampionamento ( $\epsilon_\mu=0.5$ ,  $\epsilon_\tau=0.5$ ). Infatti, nel caso biparametrico (caso in questione), dobbiamo costruire 2 catene markoviane: una per ogni parametro. E, in aggiunta, dobbiamo quindi ricordarci di togliere i burn-in. Nella figura seguente (3.23), prima riga, riportiamo le stime dei parametri ottenute “a posteriori” per  $\mu_1, \mu_2, \dots, \mu_m$  e  $\tau_1, \tau_2, \dots, \tau_m$  con  $m$  pari a 120 e un burn-in di 20 (questa è una scelta dettata dal fatto che le catene create per, rispettivamente,  $\tau$  e  $\mu$ , raffigurate nella prima riga, sembrano convergere attorno alla ventesima iterazione).

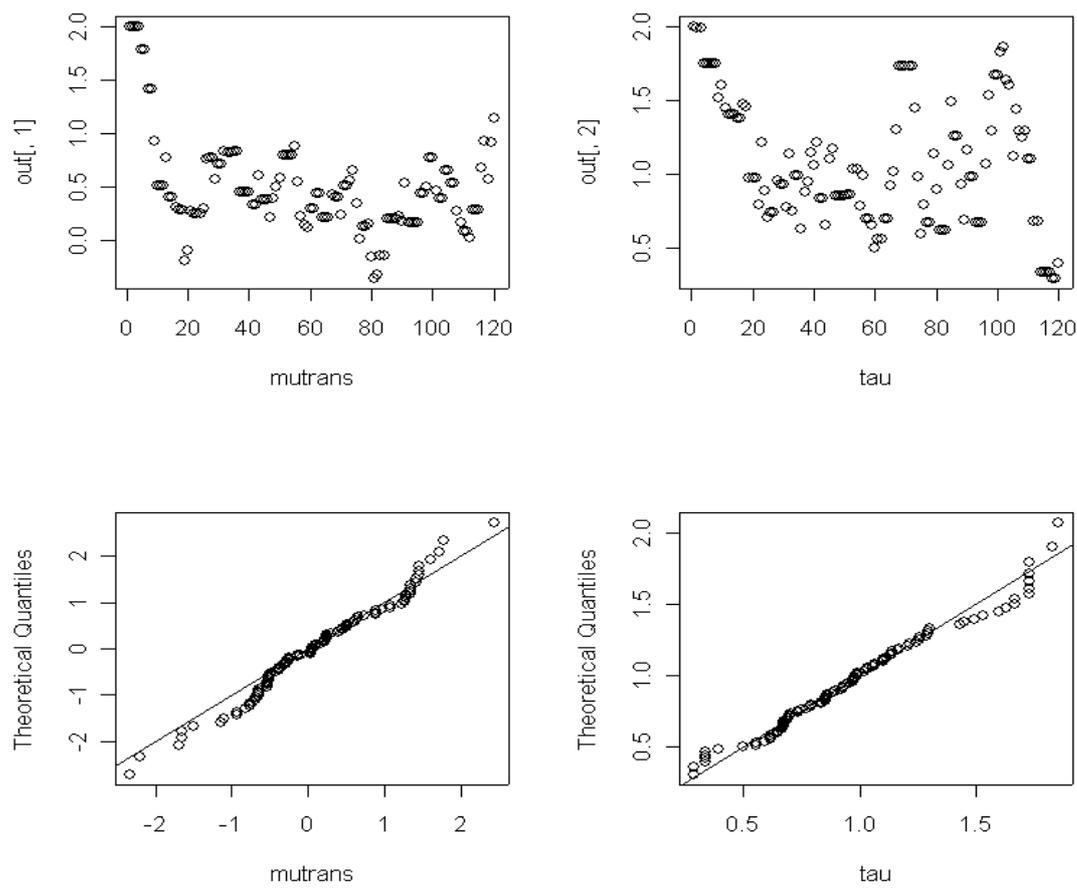


Figura 3.23

Nella seconda riga, invece riportiamo i qqplot ottenuti simulando  $\mu$  e  $\tau$ . Aumentando poi, la numerosità campionaria (Figura 3.24) della simulazione (di nuovo  $m=1100$ , con un burn-in di 100), si ha un'idea più chiara della distribuzione dei 2 parametri a posteriori, e si vede come l'adattamento sia, anche con questo metodo, più che buono.

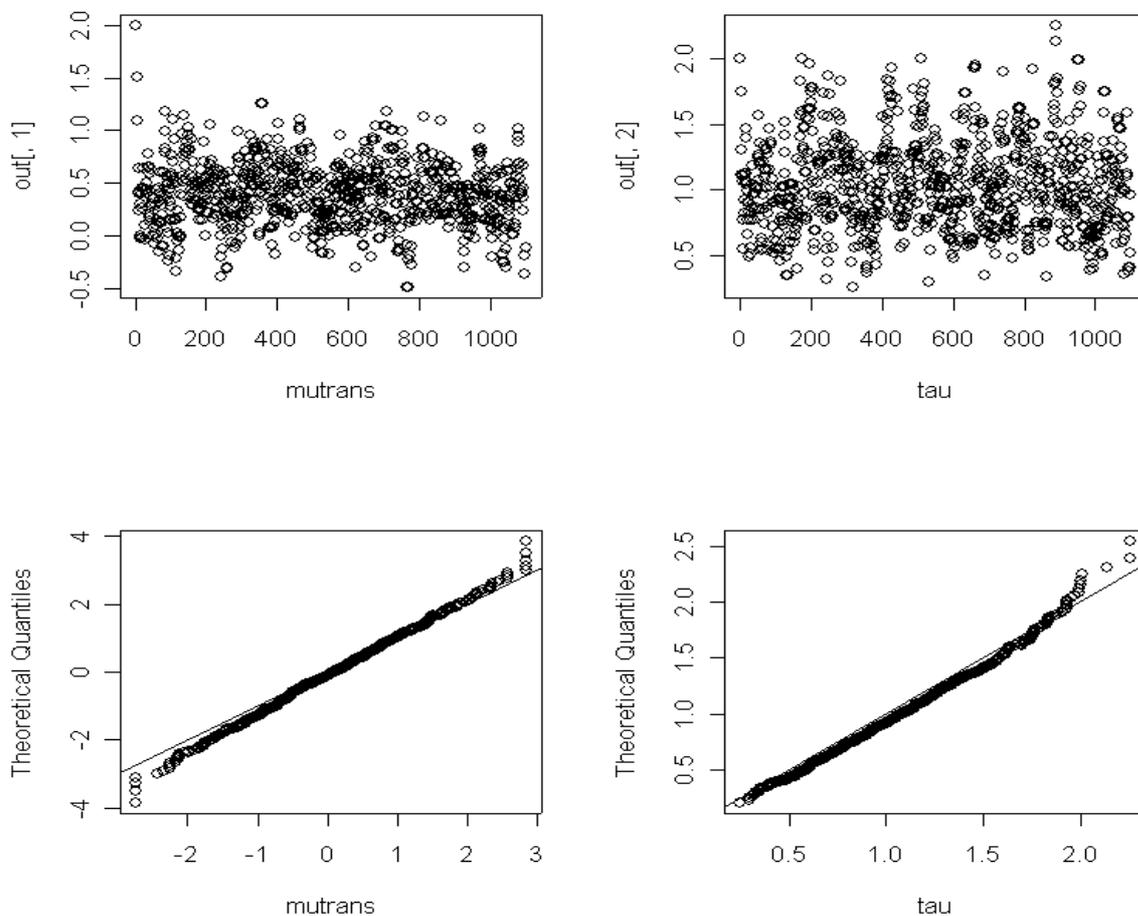


Figura 3.24

Infine, per poter confrontare direttamente MCMC con IS prendiamo come campione iniziale lo stesso di numerosità  $n=100$  utilizzato precedentemente per implementare Importance Sampling. In figura 3.25, si vede chiaramente come

sia la simulazione ottenuta per  $\mu$ trans che quella per  $\tau$ , siano pressoché ottime: i quantili empirici sembrano corrispondere al meglio con quelli teorici.

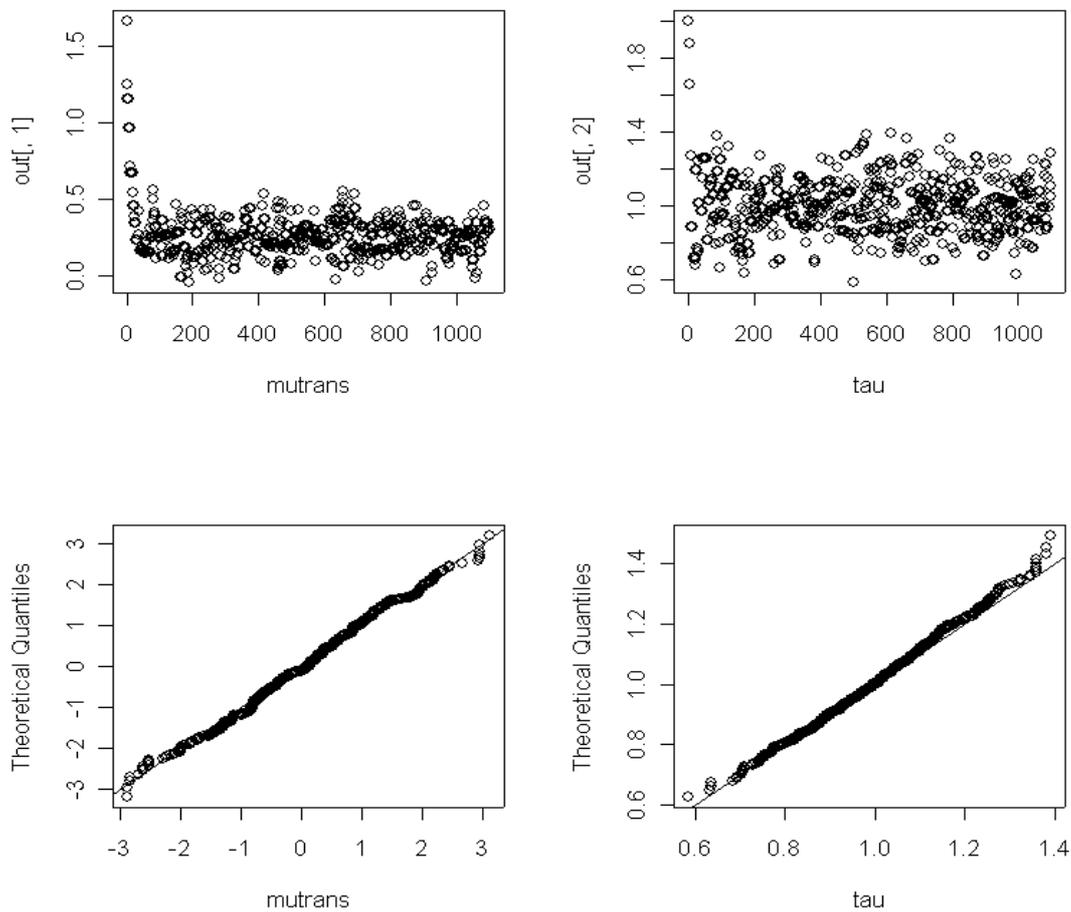


Figura 3.25

### 3.2.3 Iterative Importance Sampling

Non ci resta che vedere, ora, come si comporta IIS nel caso biparametrico: definendo  $N=100$ ,  $n=10$  e  $m$  uguale prima a 100 (prima riga in figura 3.26) e poi 1000 (seconda riga in figura 3.26), e stesse definizioni delle distribuzioni a priori e degli  $\epsilon$ . Le stime, non sembrano esser le migliori fin qui viste anche se quantili empirici e teorici non sembrano esser affatto lontani

(questo non ci stupisce minimamente se ricordiamo quanto visto nel caso uniparametrico, dove con campioni  $n=10$  IIS era l' algoritmo che funzionava peggio).

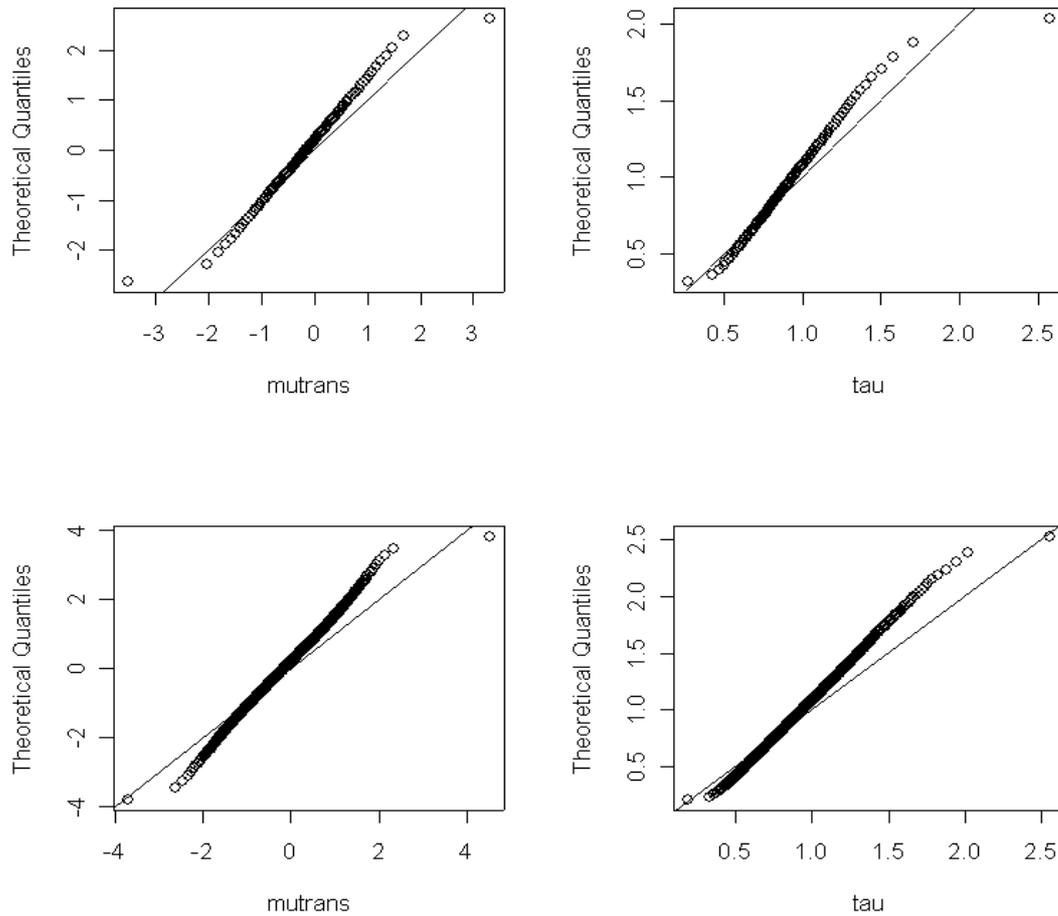


Figura 3.26

IIS infatti, funziona meglio quando l'informazione originaria proveniente dai dati è cospicua.... Aumentando la numerosità del campione ( $n=100$ ) tratto dalla  $f(x | \mu, \tau)$  l'approssimazione diventa quasi ottima:

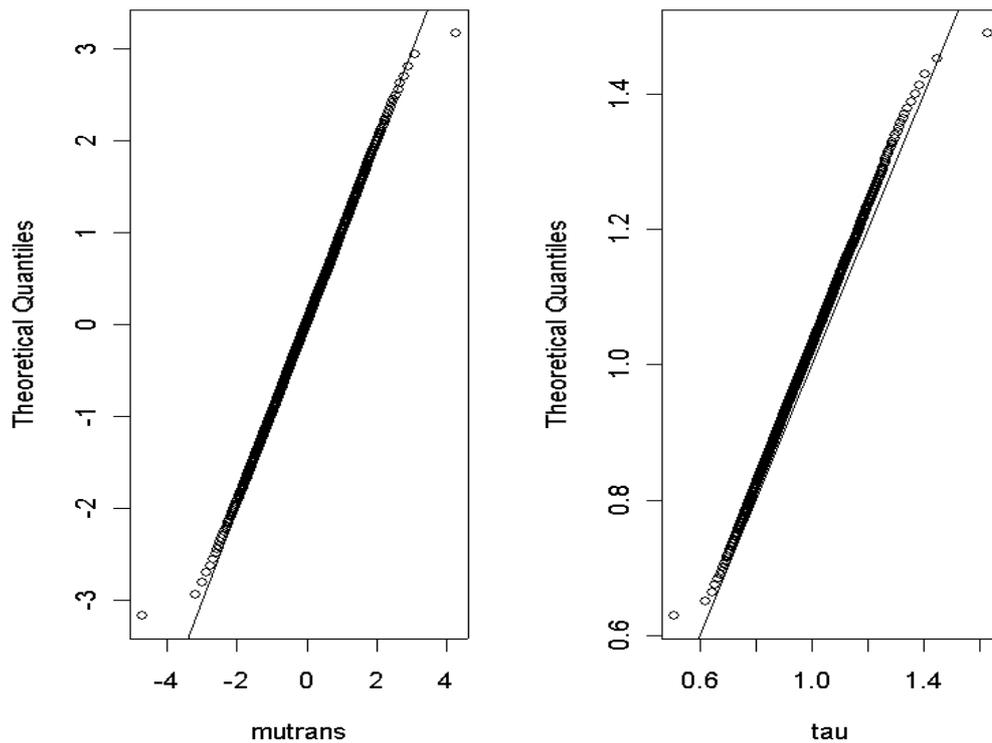


Figura 3.27

Come già detto nel caso uniparametrico, un'analisi del genere non è sufficiente per valutare il funzionamento degli algoritmi di simulazione oggetto di studio. Abbiamo bisogno di una statistica che ci permetta di quantificare l'errore di volta in volta commesso. La statistica che ci rende possibile questo è un'altra volta l'Errore Quadratico Medio.

### 3.2.4 Valutazione della bontà degli algoritmi di simulazione tramite EQM

Rispetto a quanto visto nel paragrafo 3.1.4, calcoleremo due errori quadratici medi in più:  $EQM(E(\tau))$  e  $EQM(Var(\tau))$ . Quindi, in definitiva, calcoleremo i seguenti quattro EQM (due per ogni parametro):

$$\begin{aligned}
EQM(E(\mu)) &= (1/N) \sum_i (E_i^{\wedge}(\mu) - a^*)^2, \\
EQM(\text{Var}(\mu)) &= (1/N) \sum_i (\text{Var}_i^{\wedge}(\mu) - (d^*/(b^*(c^*-1))))^2, \\
EQM(E(\tau)) &= (1/N) \sum_i (E_i^{\wedge}(\tau) - (c^*/d^*))^2, \\
EQM(\text{Var}(\tau)) &= (1/N) \sum_i (\text{Var}_i^{\wedge}(\tau) - (c^*/d^{*2}))^2;
\end{aligned}$$

dove  $E_i^{\wedge}(\mu)$  e  $\text{Var}_i^{\wedge}(\mu)$  sono, rispettivamente, media e varianza di  $\mu$  a posteriori stimate campione dopo campione;  $E_i^{\wedge}(\tau)$  e  $\text{Var}_i^{\wedge}(\tau)$  invece, sono, media e varianza di  $\tau$  a posteriori stimate di campione in campione;  $a^*$ ,  $b^*$ ,  $c^*$ ,  $d^*$  i parametri della distribuzione a posteriori (indicata nel paragrafo 3.2) che, ricordiamo, è una  $NG(a^*, b^*, c^*, d^*)$  con, a posteriori,  $E(\mu) = a^*$ ,  $\text{Var}(\mu) = (d^*/(b^*(c^*-1)))$ ,  $E(\tau) = c^*/d^*$ ,  $\text{Var}(\tau) = c^*/d^{*2}$ ;  $N$  il numero di campioni creati.

Come fatto nel paragrafo 3.1.4, proporremo ora, una serie di situazioni per valutare come cambia la bontà delle simulazioni ottenute al cambiare di alcune componenti negli algoritmi di simulazione. Nelle seguenti situazioni, indicheremo con  $f(\tau)$  e  $f(\mu | \tau)$  le distribuzioni a priori; con  $f(x | \mu, \tau)$  l'informazione campionaria espressa come dati  $x_1, x_2, \dots, x_n$  e rappresentata tramite la funzione di verosimiglianza; con  $f(\mu, \tau | x)$  la distribuzione a posteriori e i campioni ottenuti da essa tramite IS, MCMC e IIS sono, del tipo,  $\mu_1, \mu_2, \dots, \mu_m$  e  $\tau_1, \tau_2, \dots, \tau_m$ ; con  $N$  il numero di campioni prodotti, di volta in volta, attraverso la "a posteriori";  $\epsilon_\mu$ ,  $\epsilon_\tau$  e  $\mu_0$ ,  $\tau_0$  sono, rispettivamente, presi a coppie, l'ampiezza del semi intervallo utilizzato per ricampionare (sia con IIS che MCMC), e il valore per iniziare la procedura di MCMC.

## Situazione 1

Per iniziare, ipotizziamo una situazione abbastanza “semplice”: campione di partenza di numerosità  $n=10$  tratto da una  $N(0,1)$ ; prendiamo come distribuzioni a priori  $\tau \sim Ga(c, d)$  e  $\mu | \tau \sim N(a, 1/b\tau)$  con  $a=0$ ,  $b=0.5$ ,  $c=2$  e  $d=2$ ; poniamo  $m=100$  (120 per MCMC, con 20 di burn-in) e  $N$  (numero di volte che ripetiamo gli algoritmi) pari a 1000; infine, definiamo  $\varepsilon_\mu=0.5$ ,  $\varepsilon_\tau=0.5$ ,  $\mu_0=2$  e  $\tau_0=2$ . Nella seguente figura (3.28) riportiamo gli EQM ottenuti per la media e la varianza di  $\mu$  e per media e varianza di  $\tau$ . I grafici in figura 3.28, vanno letti nello stesso modo di quelli utilizzati nel capitolo precedente per il caso uniparametrico: maggiore è l’altezza delle barre, e maggiore è l’EQM commesso dall’algoritmo di simulazione. I valori indicati, sono ottenuti riparametrizzando i valori originari dell’EQM che essendo prossimi a zero, erano difficili da rappresentare ( $1/(-\log(\text{EQM}()))$ ). Non sembra qui esservi un algoritmo univocamente migliore. Nella stima di  $\text{EQM}(E(\mu))$  e  $\text{EQM}(\text{Var}(\mu))$ , rispettivamente, IS e MCMC sembrano essere gli algoritmi migliori. Nella stima di  $\text{EQM}(E(\tau))$  e  $\text{EQM}(\text{Var}(\tau))$  sembra essere IIS l’algoritmo che sembra funzionare meglio. Rispetto alla situazione 1 del caso unidimensionale, non riusciamo ad individuare in modo univoco l’algoritmo migliore per questa situazione di “povertà” campionaria.

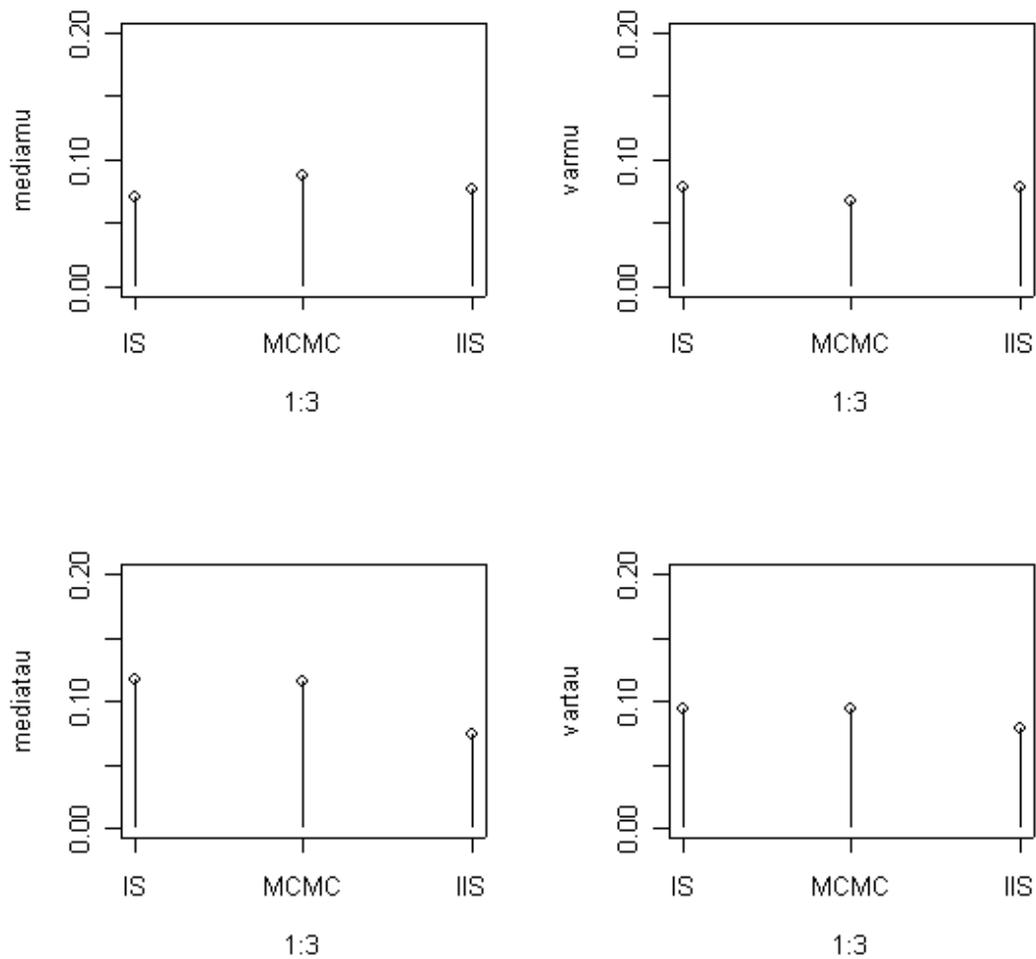


Figura 3.28

## Situazione 2

Tutto come situazione 1, eccetto che incrementiamo la numerosità campionaria della simulazione a posteriori in corrispondenza di ogni iterazione (portiamo  $m$  da 100 a 1000, MCMC con  $m=1100$  e burn in pari a 100). In figura 3.29, riportiamo le stime ottenute:

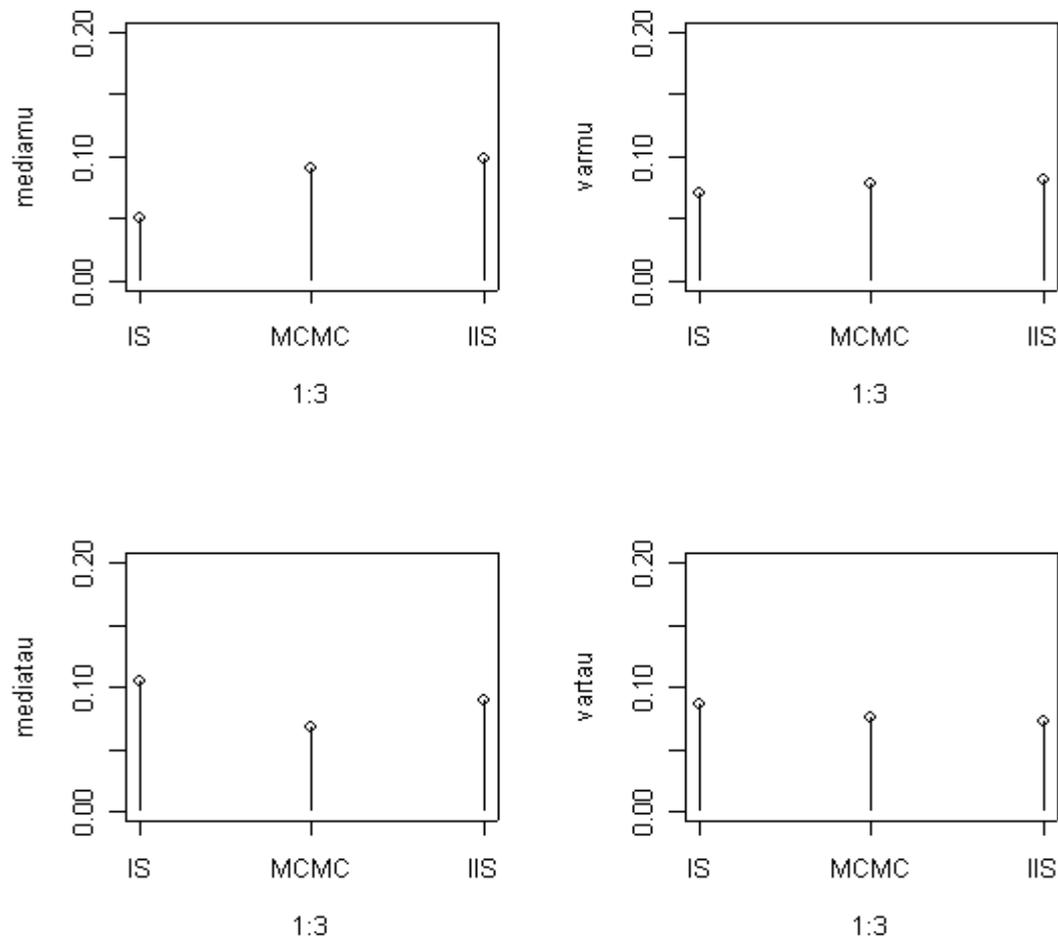


Figura 3.29

Aumentando  $m$  le cose non sembrano cambiare di molto: IS risulta l’algoritmo migliore in due situazioni su quattro (media e varianza di  $\mu$ ). Nella stima degli altri EQM (seconda riga di grafici in basso in Figura 3.29) MCMC e IIS sembrano funzionare un pochino meglio. Anche in questa situazione, quindi, non siamo riusciti ad individuare l’algoritmo “principe” cioè quello che sia il migliore in riferimento a tutti gli EQM.

### Situazione 3

In questa situazione vedremo come cambiano le cose quando l'informazione campionaria iniziale è più cospicua. Nel caso uniparametrico IIS era l'algoritmo che funzionava meglio: vediamo se questo comportamento, è confermato anche nel caso biparametrico. Prendiamo come caratteristiche iniziali, quelle di situazione 2, incrementando il valore di  $n$  e portandolo a 100. In figura 3.30, riportiamo le stime ottenute in questa situazione:

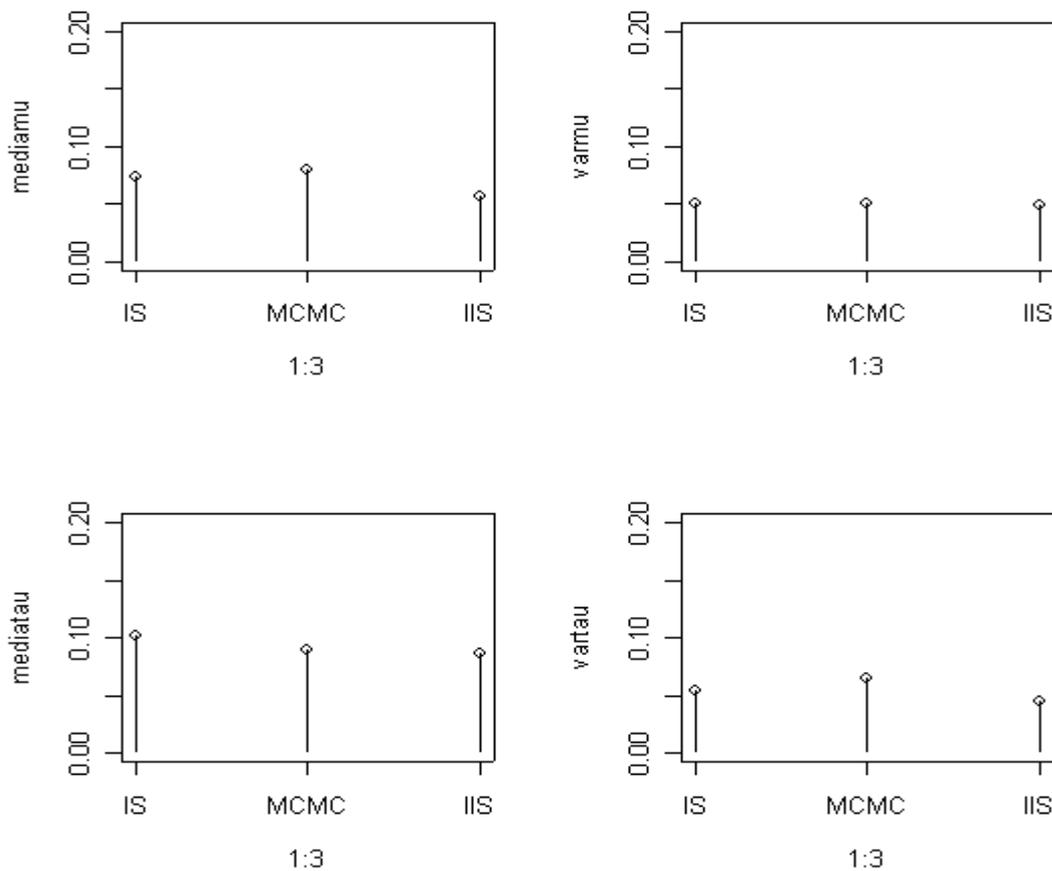


Figura 3.30

In effetti IIS, in ogni situazione, sembra l'algoritmo più preciso (e, a mano a mano che aumenta il valore di  $n$ , aumenta la precisione nelle stime di IIS, rispetto agli altri metodi, per questione di spazio, non riportiamo i grafici per

valori di  $n$  maggiori). Quindi, come nel caso uniparametrico, quando l'informazione campionaria iniziale è ricca IIS è l'algoritmo migliore tra quelli oggetto di studio.

#### Situazione 4

Proviamo ora a vedere, come cambiano i comportamenti dei tre metodi di simulazione al variare dei “passi di campionamento”  $\epsilon_\mu$  e  $\epsilon_\tau$ , e dei valori di inizializzazione dell'algoritmo  $\mu_0$  e  $\tau_0$  (utilizzati solo con MCMC). Prendiamo come caratteristiche iniziali, le stesse della situazione 3. Abbiamo proposto tre valori (0.3, 0.5 e 0.7) per  $\epsilon_\mu$  e  $\epsilon_\tau$ , e tre (2, 5 e 9) per  $\mu_0$  e  $\tau_0$ . In figura 3.31, riportiamo i valori assunti dagli EQM per le simulazioni ottenute con MCMC con valori di  $\epsilon_\mu$  e  $\epsilon_\tau$  pari a 0.3, 0.5 e 0.7.

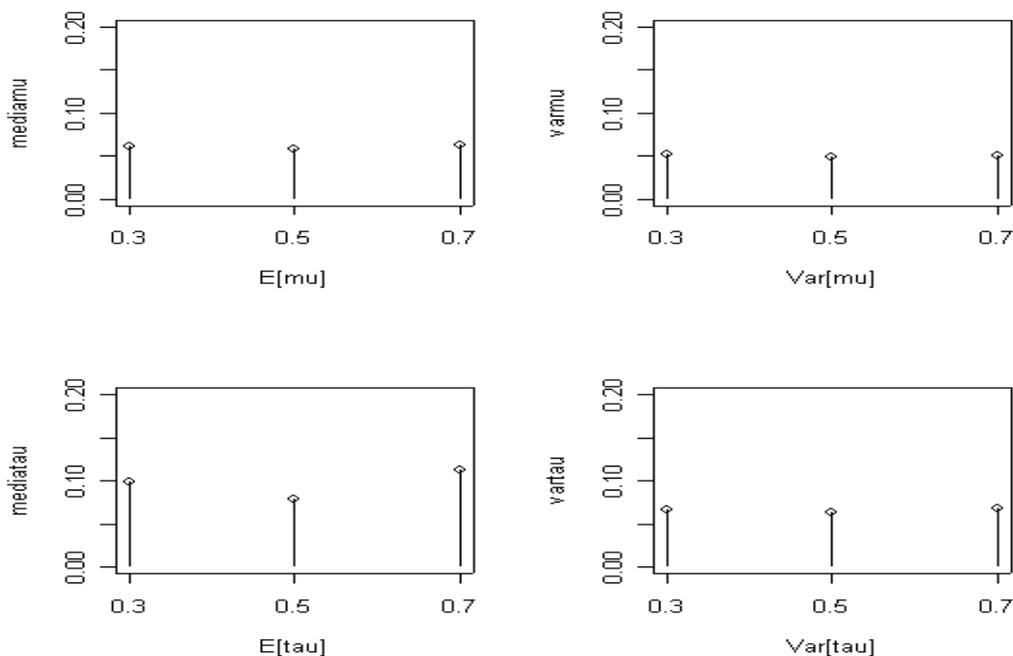


Figura 3.31

Dai grafici della prima riga di figura 3.31 (gli EQM per  $\mu$ ), si vede come le stime più precise, vengono ottenute (in entrambe i metodi di poco) con un  $\epsilon_\mu$  pari a 0.5. Anche per  $\tau$  (seconda riga) le stime migliori si hanno in corrispondenza di  $\epsilon_\tau=0.5$  (molto probabilmente per il solito discorso di compromesso tra convergenza e dipendenza nella catena creata). Proviamo ora a vedere se, con gli stessi valori per  $\epsilon$ , IIS si comporta nello stesso modo. In figura 3.32, ne riportiamo le stime ottenute.

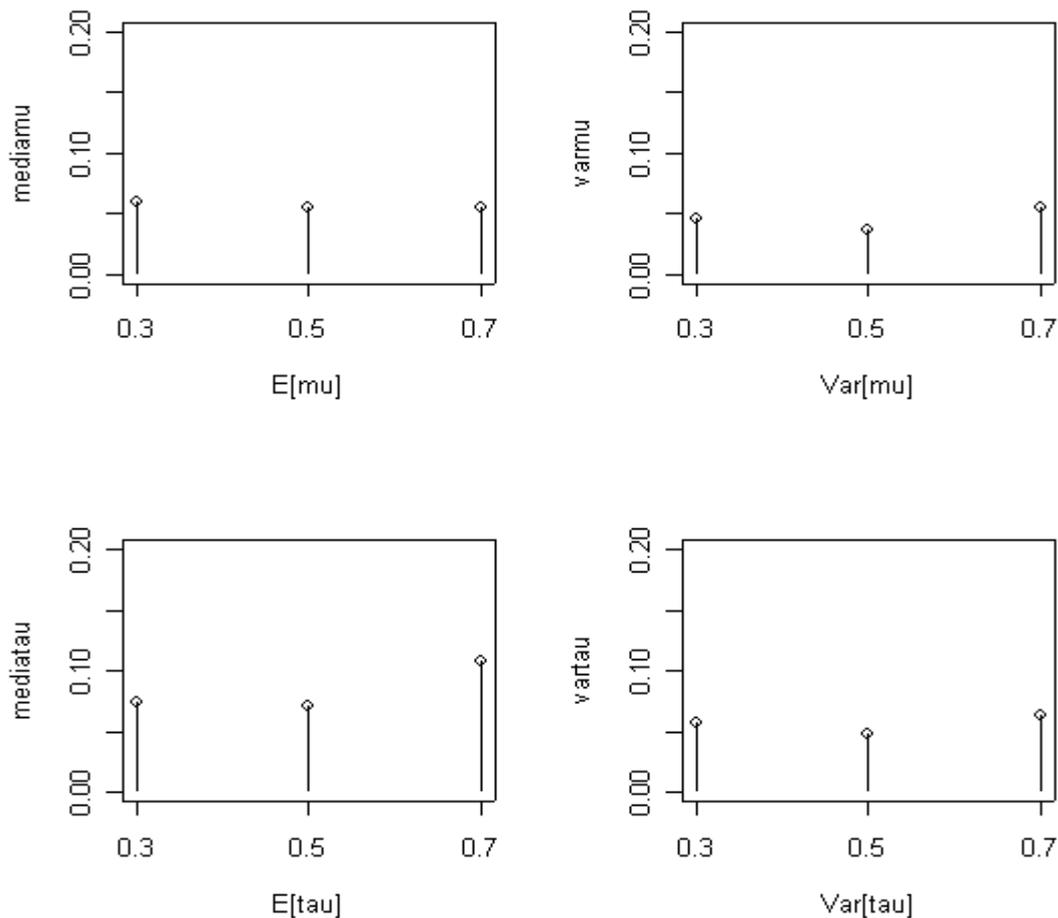


Figura 3.32

Anche con IIS le stime più precise si ottengono con valori di  $\epsilon_\mu$  e  $\epsilon_\tau$  uguali a 0.5. Un'ultima cosa ci rimane da vedere: come cambia la bontà della simulazione al cambiare dei valori di inizializzazione dell'algoritmo (utilizzati

solo da MCMC)  $\mu_0$  e  $\tau_0$ . Abbiamo ipotizzato come valori iniziali 2, 5 e 9 e le stime ottenute, le abbiamo riportate in figura 3.33.

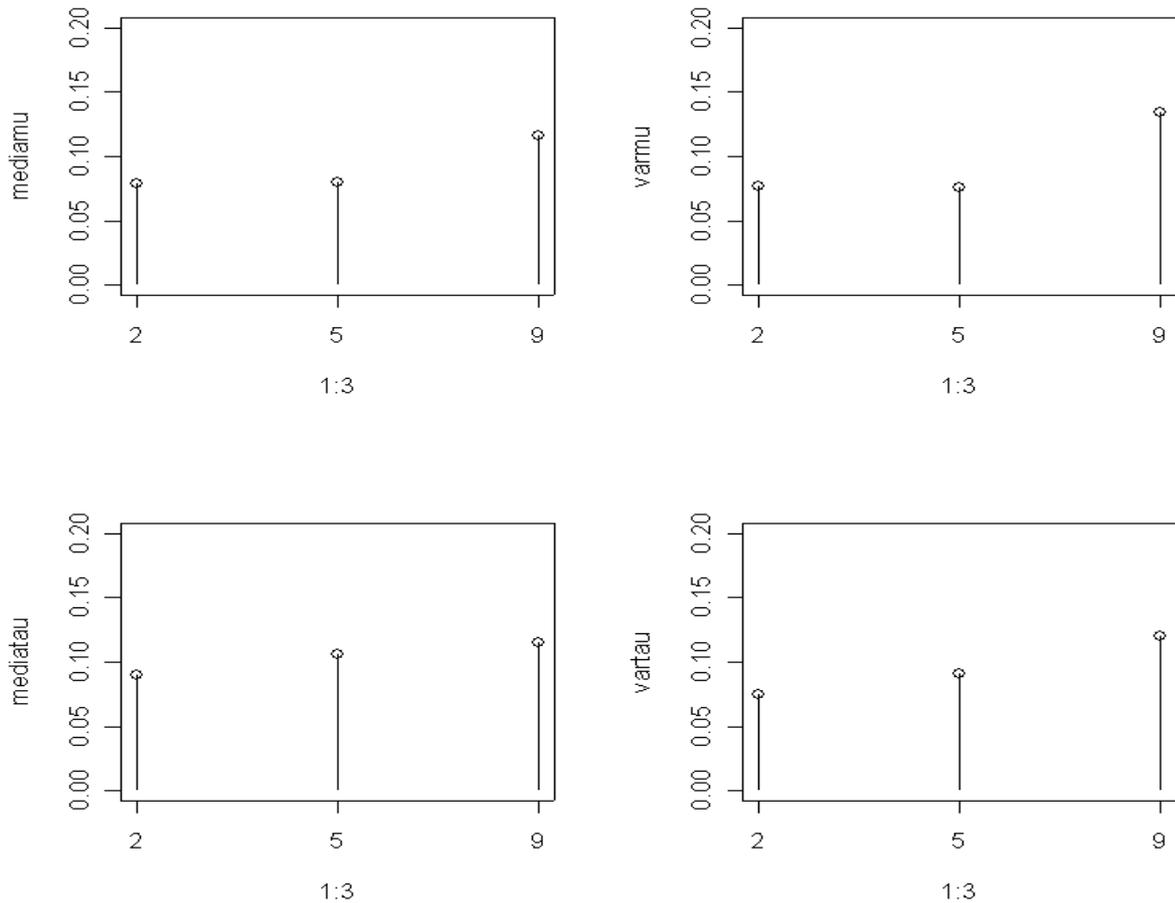


Figura 3.33

In figura 3.33 si vede chiaramente come le stime migliori si abbiano con valori di  $\mu_0$  e  $\tau_0$  pari a 2. Ciò non ci stupisce minimamente: Più i valori di inizializzazione dell'algoritmo, sono vicini al nucleo della distribuzione a posteriori e più veloce è la convergenza della catena Markoviana.

## Situazione 5

Per ripercorrere in pieno, tutto quanto fatto nel paragrafo 3.1 (caso uniparametrico) ci manca solo da analizzare una situazione esplorativa nella quale  $E(\mu)$  ed  $E(\tau)$  siano note nella popolazione, ma non siano conosciute  $\text{Var}(\mu)$  e  $\text{Var}(\tau)$ . Fino ad adesso, nelle quattro situazioni precedenti di questo paragrafo 3.2, abbiamo ipotizzato che  $a$ ,  $b$ ,  $c$  e  $d$  fossero, rispettivamente, pari a 0, 0.5, 2, 2. Sostituendo, si otteneva  $E(\mu)=a=0$ ,  $\text{Var}(\mu)=d/(b(c-1))=4$ ,  $E(\tau)=c/d=1$ ,  $\text{Var}(\tau)=c/d^2=0.5$  :quindi varianze abbastanza piccole. In una situazione esplorativa sarebbe invece giusto ipotizzare varianze più elevate dato che il fenomeno non è conosciuto. Quindi, abbiamo posto  $a=0$ ,  $b=0.0001$   $c=400$  e  $d=2$  ottenendo varianze più elevate che tengano conto del fatto che la vera distribuzione a priori è non nota. Con questi valori, infatti, si ottiene che:  $E(\mu)=a=0$ ,  $\text{Var}(\mu)=d/(b(c-1))\cong 50.13$ ,  $E(\tau)=c/d=200$ ,  $\text{Var}(\tau)=c/d^2=100$ . Abbiamo provato ad implementare IS, MCMC e IIS con le stesse condizioni iniziali della situazione 2: quindi,  $\varepsilon_\mu=0.5$ ,  $\varepsilon_\tau=0.5$ ,  $\mu_0=2$  e  $\tau_0=2$ ,  $N=1000$ ,  $m=1000$ ,  $n=10$ . Abbiamo però incontrato un grosso problema. Gli stessi algoritmi (anche modificando più volte i valori di  $a$ ,  $b$ ,  $c$ ,  $d$ ) funzionanti con  $a=0$ ,  $b=0.5$ ,  $c=2$  e  $d=2$ , cambiando i valori dei parametri in modo tale da considerare varianze elevate, davano dei problemi di tipo numerico. La risoluzione di problemi di questo tipo si è resa alquanto difficile e richiederebbe tecniche che vanno al di là dei contenuti e delle problematiche di questa tesi. Abbiamo quindi deciso di non approfondire questa cosa e di lasciarne a chi interessato la risoluzione.

### *3.2.5 Conclusioni*

Ciò che emerge in questo paragrafo (3.2), ci porta ad ottenere solo in parte, conclusioni simili a quanto visto nel caso uniparametrico. IIS, infatti, è l'algoritmo migliore nel caso di "ricchezza" campionaria, mentre quando l'informazione campionaria iniziale è scarsa non si riesce ad individuare un'algoritmo migliore in ogni situazione. Ci resta il dubbio di quale possa essere il comportamento degli algoritmi di simulazioni nel caso di situazione esplorativa (media conosciuta e varianza sconosciuta, ipotizzata elevata).

Dopo quest'analisi, nel prossimo capitolo ci occuperemo di analizzare come si comportano gli algoritmi di simulazione in situazioni "bayesiane" dove ci siano problematiche del tipo Valori Estremi.



## CAPITOLO 4

### INFERENZA BAYESIANA PER VALORI ESTREMI

Ci sono diversi ambiti in cui si ha a disposizione un certo numero di osservazioni ma non si è interessati alla “forma” della distribuzione ma, piuttosto, ai valori massimi (o minimi) che un certo fenomeno può assumere. Saper valutare, ad esempio, quale possa essere il valore massimo assunto dalle maree in un certo intervallo di tempo, può essere di rilevante importanza per cercare di limitarne l’effetto. I metodi statistici standard risultano essere inadatti a modellare questi fenomeni. Bisogna quindi proporre nuovi metodi: la teoria dei valori estremi, permette questo. Il difetto di un approccio a “valori estremi” è che si hanno poche osservazioni a disposizione (tornando all’esempio delle maree, le informazioni sulla bassa marea non sono particolarmente “informative” per prevedere l’alta) dato che gran parte delle osservazioni campionarie non sono utili allo scopo. Avere poca informazione, “obbliga” ad ottenere stime che possono essere sì corrette, ma con un’elevata variabilità. Ma se sono note delle informazioni aggiuntive (a priori) sulla distribuzione del fenomeno, arricchiamo l’insieme informativo a nostra disposizione. Ciò ci permette di ottenere stime più precise, attendibili e di ridurre sensibilmente la variabilità. In questo capitolo, infatti, daremo prima una serie di richiami sulla teoria dei valori estremi (ci soffermeremo solo sulla teoria classica dei valori estremi tralasciando i modelli a soglia: per riferimenti si veda *Coles, 2001*), cercando poi di riproporla secondo un’ottica di tipo bayesiano, provando ad applicare IS, MCMC e IIS per simulare valori dalle distribuzioni ottenute.

#### 4.1 Richiami sulla teoria dei valori estremi: l'approccio classico

Ipotizziamo di avere una sequenza di variabili casuali indipendenti ed identicamente distribuite  $X_1, X_2, \dots, X_n$  con una comune funzione di ripartizione  $F$  e di essere interessati a valutare il comportamento della seguente statistica:

$$M_n = \max \{ X_1, X_2, \dots, X_n \}$$

Dove  $M_n$ , rappresenta il massimo tra le  $n$  osservazioni disponibili. Ad esempio, se  $n$  rappresenta il numero di osservazioni sulle temperature disponibili in un anno,  $M_n$  corrisponde al valore massimo annuale registrato. In teoria, la distribuzione di  $M_n$  può essere ottenuta come:

$$\begin{aligned} \Pr\{ M_n \leq z \} &= \Pr\{ X_1 \leq z, \dots, X_n \leq z \} \\ &= \Pr\{ X_1 \leq z \} \times \dots \times \Pr\{ X_n \leq z \} \\ &= \{ F(z)^n \} \end{aligned}$$

Il problema è che questo risultato non è direttamente applicabile, dato che  $F$  è sconosciuta. Come facciamo quindi a stimare  $\{F(z)^n\}$ ? Un possibile approccio è il seguente (quest'approccio è analogo a quanto succede con il teorema del limite centrale). Accettiamo il fatto che  $F$  sia sconosciuta e guardiamo all'approssimazione asintotica di  $F^n$  (che può essere stimata direttamente con i dati originari) per  $n \rightarrow \infty$ . Ma questo da solo non è sufficiente: per ogni  $z < z_+$ , dove  $z_+$  è il limite superiore della funzione di ripartizione  $F$ ,  $F^n(z) \rightarrow 0$  per

$n \rightarrow \infty$ , la distribuzione di  $M_n$ , degenera in un punto di massa su  $z_+$ . Per evitare questo, normalizziamo la variabile  $M_n$  ottenendo  $M_n^*$  nel seguente modo:

$$M_n^* = (M_n - a_n) / b_n$$

Per sequenze di costanti  $\{ a_n > 0 \}$  e  $\{ b_n \}$ . Appropriate scelte di  $\{ a_n \}$  e  $\{ b_n \}$  stabilizzano la distribuzione di  $M_n^*$  all'aumentare di  $n$  evitando le difficoltà connesse nello studio di  $M_n$ .

Scegliendo quindi, appropriati valori delle costanti  $\{ a_n \}$  e  $\{ b_n \}$  otteniamo una classe di distribuzioni limite per  $M_n^*$ . I risultati fin qui ottenuti, sono enunciati nel seguente teorema (4.1).

#### *Teorema 4.1*

Se esistono sequenze di costanti  $\{ a_n > 0 \}$  e  $\{ b_n \}$  tali che:

$$\Pr\{ (M_n - b_n) / a_n \leq z \} \rightarrow G(z) \quad \text{per } n \rightarrow \infty$$

Dove  $G$  è una distribuzione di densità non degenera che può “appartenere” ad una delle seguenti tre famiglie:

$$\text{I: } G(z) = \exp\{ -\exp[-((z - b)/a)] \}, \quad -\infty < z < \infty;$$

$$0, \quad \text{per } z \leq b;$$

$$\text{II: } G(z) =$$

$$\exp\{ -((z - b)/a)^{-\alpha} \}, \quad \text{per } z > b;$$

$$\text{III: } G(z) = \begin{cases} \exp \{-[-((z - b)/a)^\alpha]\}, & \text{per } z < b; \\ 1, & \text{per } z \geq b; \end{cases}$$

per parametri  $a > 0$ ,  $b$  e nel caso delle famiglie II e III,  $\alpha > 0$ .

In parole povere, il teorema 4.1 dice che  $M_n^*$  converge in distribuzione ad una delle tre famiglie sopra descritte. Queste tre famiglie prendono il nome, nell'ordine, di *Gumbel*, *Fréchet* e *Weibull* e nel loro insieme, formano la distribuzione dei valori estremi (anche detta GEV). Ogni famiglia ha un parametro di posizione ( $b$ ) ed uno di scala ( $a$ ), *Fréchet* e *Weibull* hanno anche un parametro di forma  $\alpha$ . Ciò che emerge in maniera rilevante è che  $M_n^*$  asintoticamente può “convergere” solamente ad una delle tre distribuzioni sopraindicate e a nessun'altra. Ciò è molto importante, perché riusciamo a limitare, e di molto, lo spazio delle possibili distribuzioni assunte da  $M_n^*$ .

Le tre famiglie sopra indicate si differenziano per il comportamento assunto (soprattutto in corrispondenza delle loro code) nella distribuzione delle  $X_i$  (ad esempio, la funzione di densità della  $G$  nel caso Gumbel decade esponenzialmente, mentre nel caso Fréchet polinomialmente). Ma come facciamo a saper quale sia la distribuzione di volta in volta più adatta per modellare i dati a nostra disposizione? La soluzione, si ha combinando le famiglie Gumbel, Fréchet e Weibull in un'unica famiglia: la GEV (Generalized Extreme Value). La GEV, ha la seguente funzione di densità:

$$G(z) = \exp \{-[1 + \xi ((z - \mu)/\sigma)]^{-1/\xi}\},$$

Definita solo se  $z: 1 + \xi ((z - \mu)/\sigma) > 0$  e con parametri  $-\infty < \mu < \infty$ ,  $\sigma > 0$  e  $-\infty < \xi < \infty$ . Il modello ha tre parametri: un parametro di locazione,  $\mu$ ; un parametro di scala,  $\sigma$ ; ed un parametro di forma  $\xi$ . Proprio a seconda del valore assunto da  $\xi$ , possiamo ricondurci alle famiglie ottenute nel teorema 4.1: con  $\xi > 0$  e  $\xi < 0$  otteniamo, rispettivamente *Fréchet* e *Weibull*; con  $\xi = 0$  si ottiene la *Gumbel*. L'unione delle tre distribuzioni nella GEV è molto importante perché, stimando un modello e facendo inferenza sul parametro  $\xi$ , possiamo capire quale famiglia possa essere di volta in volta la più adatta a modellare i dati, senza stimare preventivamente modelli magari non adatti (ad esempio stimiamo una Weibull quando la distribuzione più adatta è una Gumbel). Possiamo ora riformulare il teorema 4.1 nel seguente modo:

#### *Teorema 4.2*

Se esistono sequenze di costanti  $\{ a_n > 0 \}$  e  $\{ b_n \}$  tali che:

$$\Pr\{ (M_n - b_n) / a_n \leq z \} \rightarrow G(z) \quad \text{per } n \rightarrow \infty$$

Per una distribuzione  $G$  non degenera allora  $G$  fa parte della famiglia GEV:

$$G(z) = \exp \{ - [1 + \xi ((z - \mu) / \sigma)]^{-1/\xi} \},$$

definita se  $\{ z : 1 + \xi ((z - \mu) / \sigma) > 0 \}$ , dove  $-\infty < \mu < \infty$ ,  $\sigma > 0$  e  $-\infty < \xi < \infty$

Un inciso: ci sono alcune situazioni dove si può essere interessati ai valori minimi di un certo fenomeno piuttosto che a quelli massimi. Ad esempio, un'impresa può essere interessata a conoscere la lunghezza di rottura delle proprie fibre di vetro prodotte. In questo caso, importante è rilevare quale sia la *minima* lunghezza alla quale avviene la rottura della fibra. Per situazioni come queste, il fenomeno oggetto di interesse è:

$$M_n^\# = \min \{ X_1, X_2, \dots, X_n \}$$

Dove le  $X_i$  sono indipendenti, identicamente distribuite e rappresentano le osservazioni disponibili sul fenomeno in questione. Per modellare questa tipologia di situazione ci sono 2 metodi. Il primo metodo, consiste nel considerare  $Y_i = -X_i$ . Questo cambio nel segno delle osservazioni originarie, ci consente di applicare la modellazione con la famiglia GEV studiata nel corso di questo paragrafo. Infatti, studiare  $M_n = \max \{ Y_1, Y_2, \dots, Y_n \}$  equivale a studiare  $M_n^\# = \min \{ X_1, X_2, \dots, X_n \}$  con l'uguaglianza  $M_n^\# = -M_n$ . Il secondo metodo consiste nell'applicare la "GEV distribution for minima" che fa cambiare il teorema 4.2 nel seguente modo:

### *Teorema 4.3*

Se esistono sequenze di costanti  $\{ a_n > 0 \}$  e  $\{ b_n \}$  tali che:

$$\Pr \{ (M_n^\# - b_n) / a_n \leq z \} \rightarrow G^\#(z) \quad \text{per } n \rightarrow \infty$$

Per una distribuzione  $G^\#$  non degenera allora  $G^\#$  fa parte della "GEV distribution for minima":

$$G^\#(z) = 1 - \exp \{ - [1 - \xi ((z - \mu^\#) / \sigma)]^{-1/\xi} \},$$

definita se  $\{ z : 1 - \xi ((z - \mu^\#) / \sigma) > 0 \}$ , dove  $-\infty < \mu^\# < \infty$ ,  $\sigma > 0$  e  $-\infty < \xi < \infty$

Nella nostra analisi, ci limiteremo allo studio di esempi con la distribuzione GEV per valori “massimi”

#### 4.2 Inferenza per la Generalized Extreme Value (GEV)

Per stimare i valori dei parametri in problemi con “valori estremi” si possono utilizzare diverse tecniche (tra le altre, ricordiamo): metodi grafici (tramite, ad esempio, grafici di probabilità), metodo dei momenti o la “solita” verosimiglianza. Sarà proprio tramite quest’ultima che stimeremo i parametri della GEV facendo però attenzione che con valori di  $\xi \leq -0.5$ , possono esser violate le condizioni di regolarità e che, quindi, i risultati asintotici standard ottenuti con la verosimiglianza possono non essere direttamente applicabili (per fortuna, nella pratica, valori minori di  $-0.5$  sono raramente utilizzati e quindi il problema nella maggior parte dei casi non si propone). Costruiamo ora, la verosimiglianza per un modello GEV.

Infatti, ipotizzando che  $Z_1, Z_2, \dots, Z_m$  siano  $m$  variabili aleatorie indipendenti ed identicamente distribuite con distribuzione GEV comune, la log-verosimiglianza nel caso  $\xi \neq 0$  assume la seguente forma:

$$l(\mu, \sigma, \xi) = -m \log \sigma - (1 + 1/\xi) \sum \log [1 + \xi ((z_i - \mu)/\sigma)] - \sum [1 + \xi ((z_i - \mu)/\sigma)]^{-1/\xi}, \quad (i)$$

definita solo se  $1 + \xi ((z_i - \mu)/\sigma) > 0$ , per ogni  $i = 1, \dots, m$ .

Se la condizione di qui sopra è violata (e pari a zero) per almeno un  $i$ , la verosimiglianza tende a zero e di conseguenza la log verosimiglianza, tende a  $-\infty$ .

Con  $\xi = 0$ , la log-verosimiglianza assume invece la seguente forma:

$$l(\mu, \sigma) = -m \log \sigma - \sum ((z_i - \mu)/\sigma) - \sum \exp \{ -((z_i - \mu)/\sigma) \}. \quad (ii)$$

La massimizzazione delle 2 log-verosimiglianze (i) e (ii) non può essere fatta analiticamente e deve esser fatta utilizzando metodi numerici (del tipo Newton Rapson). Attenzione deve esser fatta nell'implementazione di queste tipologia di metodi dato che bisogna rispettare la condizione che  $1 + \xi ((z_i - \mu)/\sigma) > 0$ , per ogni  $i = 1, \dots, m$ . Ai fini dell'inferenza, altra cosa da sottolineare è che la distribuzione approssimata di  $(\mu^\wedge, \sigma^\wedge, \xi^\wedge)$  è una normale multivariata con media  $(\mu, \sigma, \xi)$  e matrice di varianza – covarianza uguale all'inversa della matrice di informazione osservata ottenuta dalla stima di massima verosimiglianza. Una volta calcolata quest'ultima (sempre grazie a metodi numerici) e nota la distribuzione approssimata, è semplice calcolare test e intervalli di confidenza per i parametri in questione.

### 4.3 Valori estremi e statistica bayesiana

Come già ampiamente discusso nel capitolo 2, entriamo nell'ambito della statistica bayesiana, quando è disponibile dell'informazione a priori sui parametri coinvolti nell'esperimento statistico. Quindi, ogni volta che si è costretti a lavorare con poche osservazioni (come, di fatto, succede in modelli del tipo GEV) un'informazione aggiuntiva sui parametri non ci fa altro che portare a stime più accurate (più precise e meno variabili).

Come nel capitolo 2, indicheremo con  $f(\theta)$  la distribuzione a priori; con  $f(\theta | x) = \prod f(x_i; \theta)$  l'informazione proveniente dai dati e per il teorema di Bayes la distribuzione a posteriori sarà del tipo:

$$f(\theta | x) = (f(\theta) * f(x | \theta)) / \int (f(\theta) * f(x | \theta)) d\theta \quad (i)$$

Il grosso problema che ci si presenta quando si lavora secondo un'ottica bayesiana è che per fare inferenza sui parametri è necessario calcolare l'integrale della (i). Il calcolo dell'integrale, come peraltro già ricordato in precedenza, può essere alquanto difficile soprattutto quando  $\theta$  è multidimensionale. Però, il trucco è ormai stato svelato nel capitolo precedente: gli algoritmi di simulazione tipo MCMC, IS e IIS ci permettono di superare il problema e di calcolare "facilmente" la distribuzione a posteriori.

Nel prossimo paragrafo, valuteremo proprio cosa cambia nell'applicazione dei tre metodi di simulazione oggetto di studio in quest'elaborato rispetto a quanto visto nel capitolo 3.

#### 4.4 Algoritmi di simulazione per GEV

Nel software statistico utilizzato per le elaborazioni (come già ricordato R 1.9.1) la GEV, non è tabulata e non è quindi possibile simulare direttamente come si è fatto nel capitolo 3. Inoltre, anche la funzione di densità della GEV non è tabulata (i comandi analoghi a `rnorm` o `dnorm` utilizzati per il caso normale non esistono nel caso GEV). Il primo passo è stato, quindi, realizzare funzioni che permettessero: primo, di simulare (e lo abbiamo fatto invertendone la funzione di ripartizione) direttamente da una GEV; secondo, realizzare una funzione che permettesse di calcolare direttamente la log-verosimiglianza per la famiglia GEV.

Una volta fatte le operazioni scritte precedentemente, possiamo ora analizzare come si comportano i metodi computazionali sopra descritti in problematiche tipo valori estremi.

##### 4.4.1 Importance Sampling per valori estremi

Ipotizziamo di avere un campione di numerosità 10 proveniente da una distribuzione GEV con parametri (  $\mu=0$ ,  $\sigma=1$ ,  $\xi=0.2$  ). Questo campione (generato da una Fréchet dato che  $\xi > 0$  ), rappresenterà l'informazione preliminare proveniente dai dati. Ipotizziamo, poi, di assumere che a priori i parametri, per semplicità, abbiano tutti eguale distribuzione normale con media e varianza uguali (abbiamo preso queste varianze perché funzionano abbastanza bene nel problema in questione) :

$$\mu \sim N(0, 0.5^2), \quad \sigma \sim N(0, 0.5^2), \quad \xi \sim N(0, 0.5^2);$$

e di essere poi interessati a simulare un campione dalla distribuzione a posteriori di numerosità pari a  $m = 1000$ . Applicando Importance Sampling ai dati in questione si ottengono le seguenti stime (IS viene applicato in modo analogo a quanto visto nei capitoli 2 e 3):

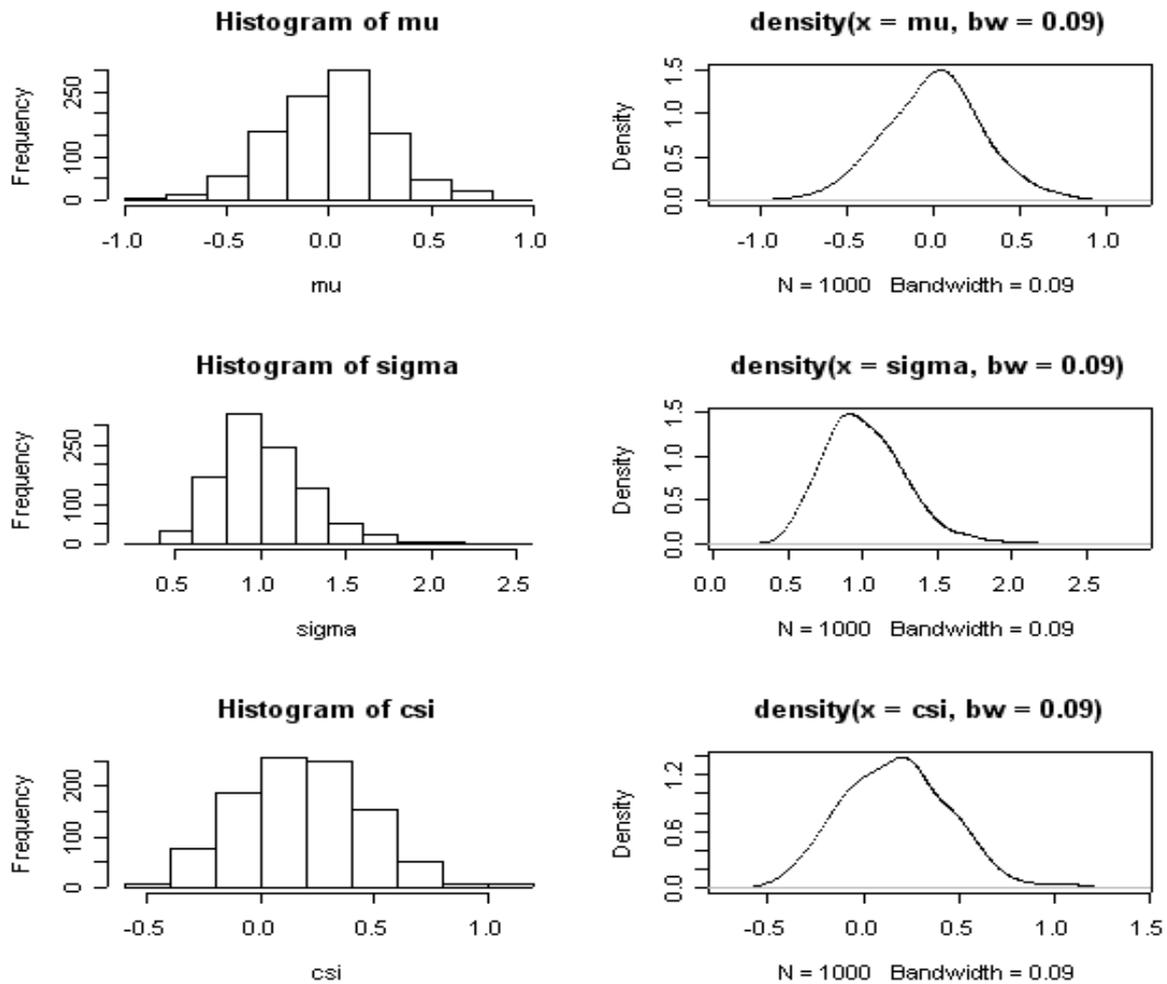


Figura 4.1

In figura 4.1 sono riportati gli istogrammi e le funzioni di densità marginali ottenute con Importance Sampling. L'approssimazione appare abbastanza buona per tutti e 3 i parametri coinvolti dato che, come si vede chiaramente nella colonna di grafici a sinistra, le densità sono centrate attorno ai valori di partenza della  $f(x | \mu, \sigma, \xi)$ . Ma Importance Sampling, per

quest'analisi, ha un grosso limite. Se, infatti incrementiamo la numerosità del campione iniziale  $n$  (passando da 10 a 100) mantenendo invariato tutto il resto succede che:

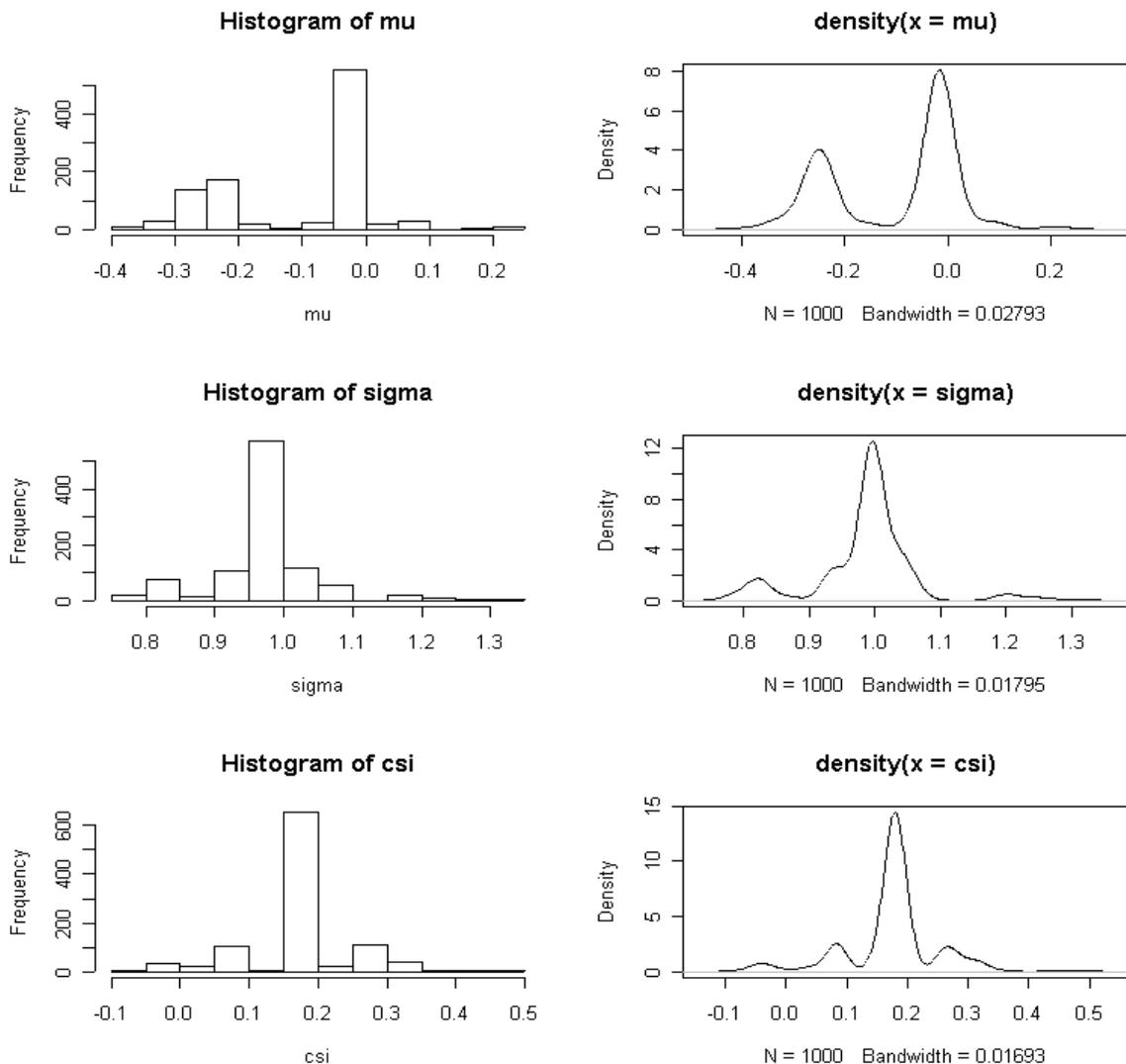


Figura 4.2

La distribuzione qui si compatta e assume anche una bimodalità per  $\mu$ . Questo dipende dal fatto che IS funziona bene solamente quando distribuzione a priori ed informazione campionaria hanno un andamento analogo. Partendo, quindi, da un campione di numerosità più elevata, e quindi più preciso, la distribuzione

che si ottiene dovrebbe esser più centrata e meno variabile: è per questa motivazione che qui IS non sembra funzionar particolarmente bene.

Nei prossimi paragrafi, proveremo a vedere se gli altri 2 metodi (IIS e MCMC) funzionino meglio in situazioni di questo tipo.

#### 4.4.2 MCMC e Valori Estremi

Anche in quest'esempio ipotizziamo:

$$X_i \sim \text{GEV} (0, 1, 0.2), \quad i=1, \dots, 10;$$

e la densità a priori (dove utilizziamo la parametrizzazione  $\phi = \log \sigma$  in modo da rispettare la positività di  $\sigma$ ) è

$$f(\mu, \phi, \xi) = f_\mu(\mu) f_\phi(\phi) f_\xi(\xi)$$

dove definiamo  $\mu \sim N(0, 0.5^2)$ ,  $\phi \sim N(0, 0.5^2)$  e  $\xi \sim N(0, 0.5^2)$  in modo da render possibile il confronto con Importance Sampling. La procedura utilizzata è uguale a quella vista nel paragrafo 2.2.2 con la sola differenza nel cambio del passo di convergenza della catena markoviana così creata. Infatti

$$\mu^* = \mu + \varepsilon_\mu, \quad \phi^* = \phi + \varepsilon_\phi, \quad \xi^* = \xi + \varepsilon_\xi;$$

e  $\varepsilon_\mu$ ,  $\varepsilon_\phi$  e  $\varepsilon_\xi$  sono variabili con distribuzione di media zero e varianza pari a 0.01. La scelta di queste varianze, è del tutto arbitraria e non modifica la bontà di adattamento del modello ma agisce solo sull'efficienza dell'algoritmo. Questa scelta, inoltre, ben si adatta con quanto deciso prima. Inizializzando

l' algoritmo con  $(\mu, \sigma, \xi) = (1, 2, 1)$  le stime delle catene markoviane ottenute (nell'ordine per  $\mu, \sigma, \xi$ ) sono riportate in figura 4.3.

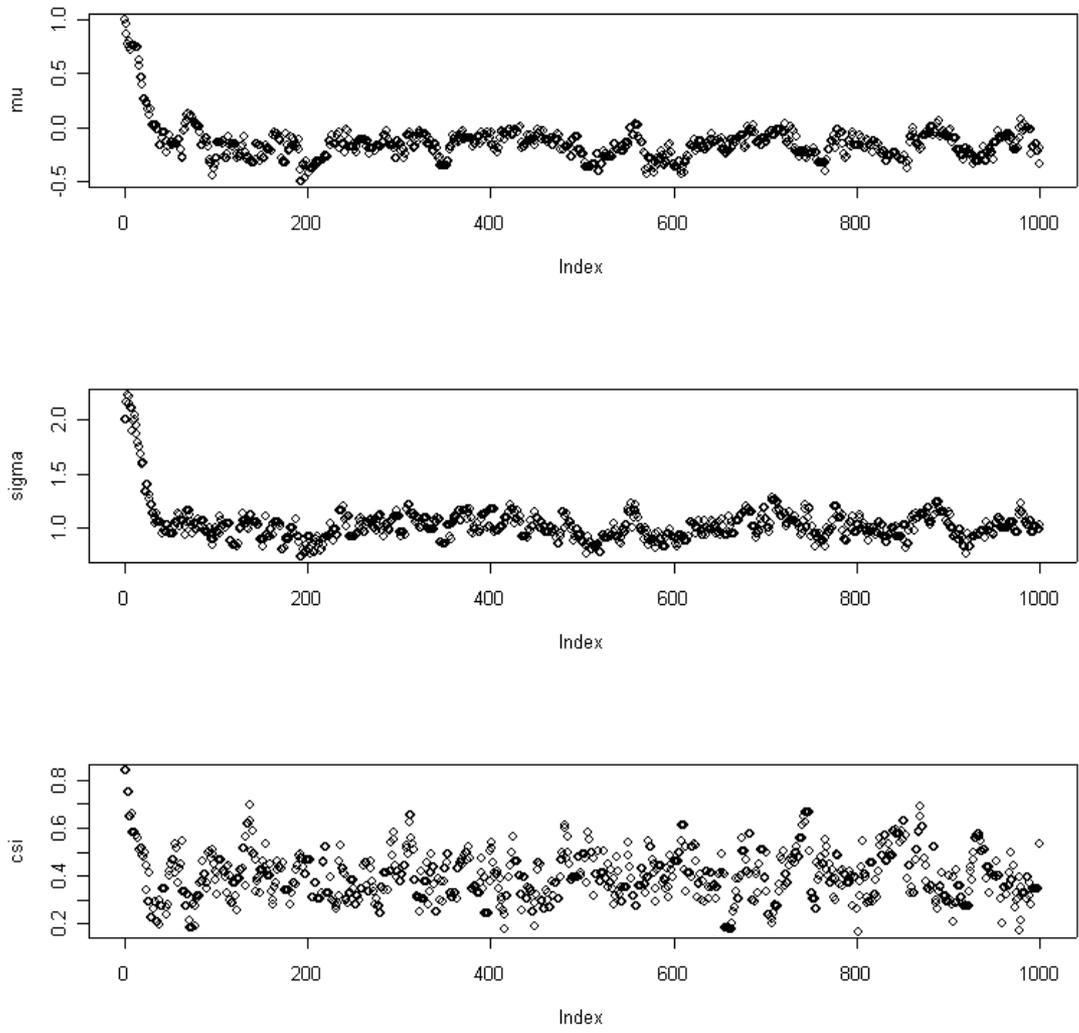


Figura 4.3

Dopo un burn-in di numerosità pari, circa, a 50 le 3 catene markoviane create (una in corrispondenza di ogni parametro) sembrano convergere ai valori iniziali del campione di partenza ( $\mu = 0, \sigma = 1, \xi = 0.2$ ). Togliendo i primi 100 valori ( $m=1100$ ) i campioni creati dovrebbero essere rappresentativi della

distribuzione a posteriori (che, ricordiamo, è non nota). Infatti in figura 4.4 presentiamo gli istogrammi e funzioni di densità ottenute.

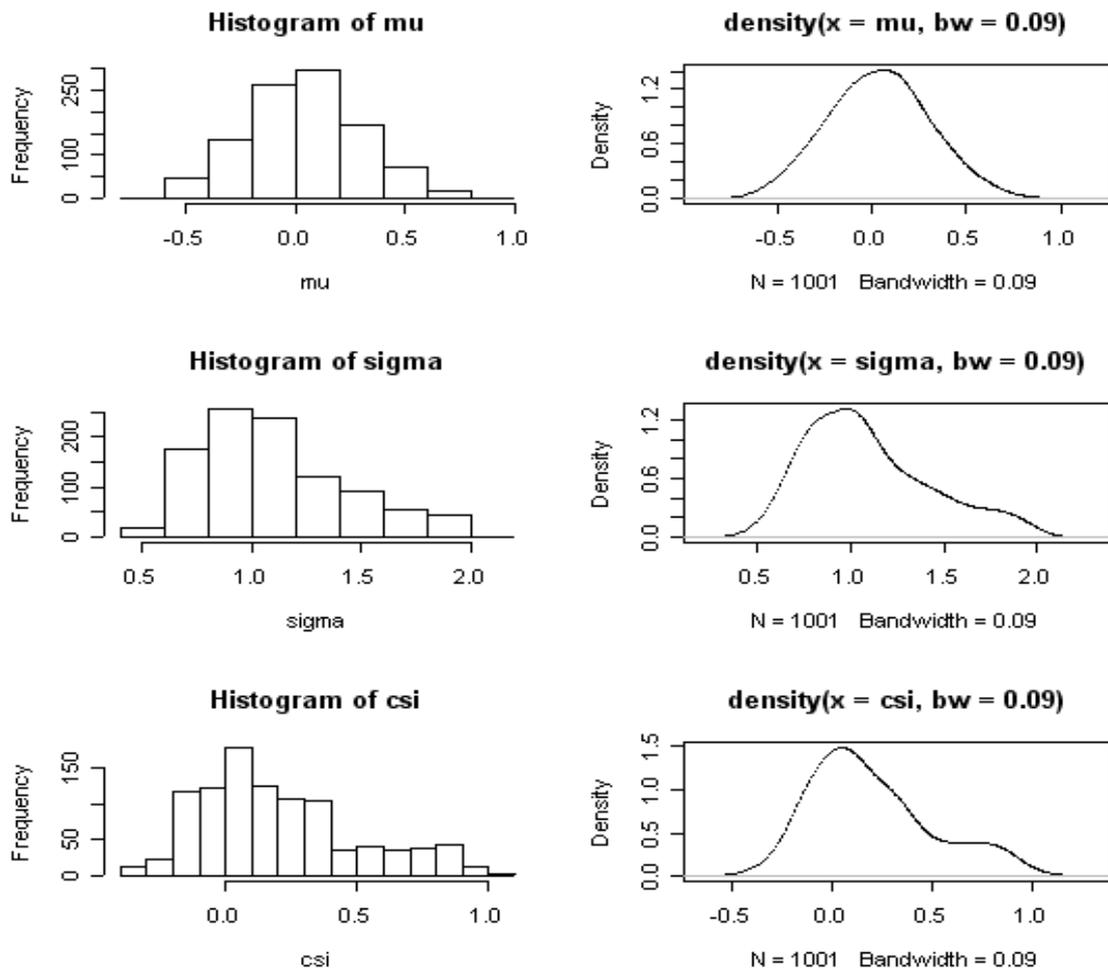


Figura 4.4

Le stime ottenute con MCMC dopo aver tolto il burn-in, sembrano abbastanza buone. Proviamo a vedere cosa cambia se ipotizziamo di prendere in considerazione il campione iniziale di numerosità pari a 100 della GEV utilizzato in precedenza per IS anziché 10, in modo da poter effettuare il confronto con MCMC. Prima, diamo un'occhiata alla convergenza dell'algoritmo.

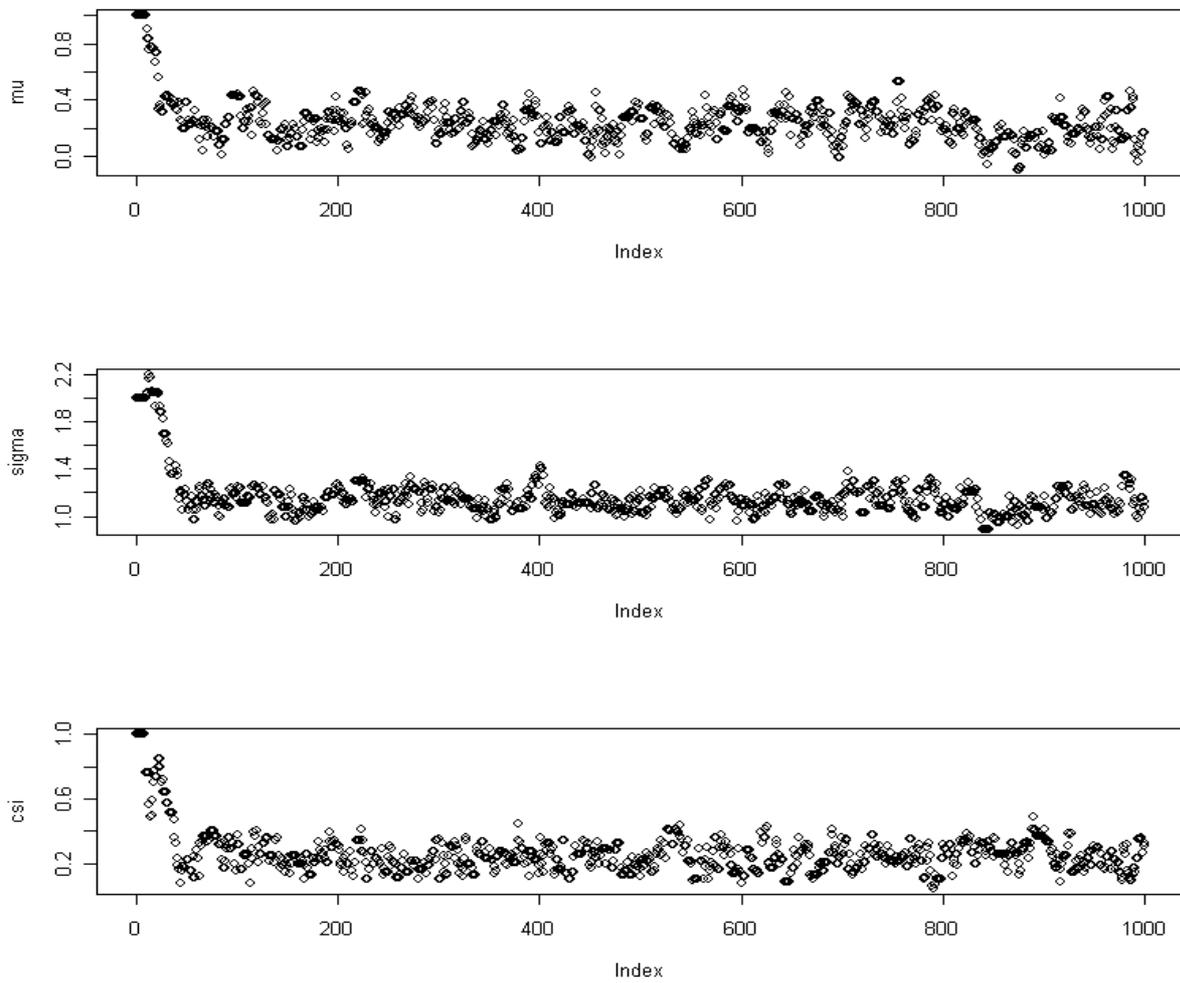


Figura 4.5

La convergenza è qui più veloce già al trentesimo valore simulato la distribuzione a posteriori sembra ben approssimata. Una buona approssimazione della distribuzione a posteriori si dovrebbe ottenere già per  $m \geq 50$ . Buttando via i primi 100 valori otteniamo:

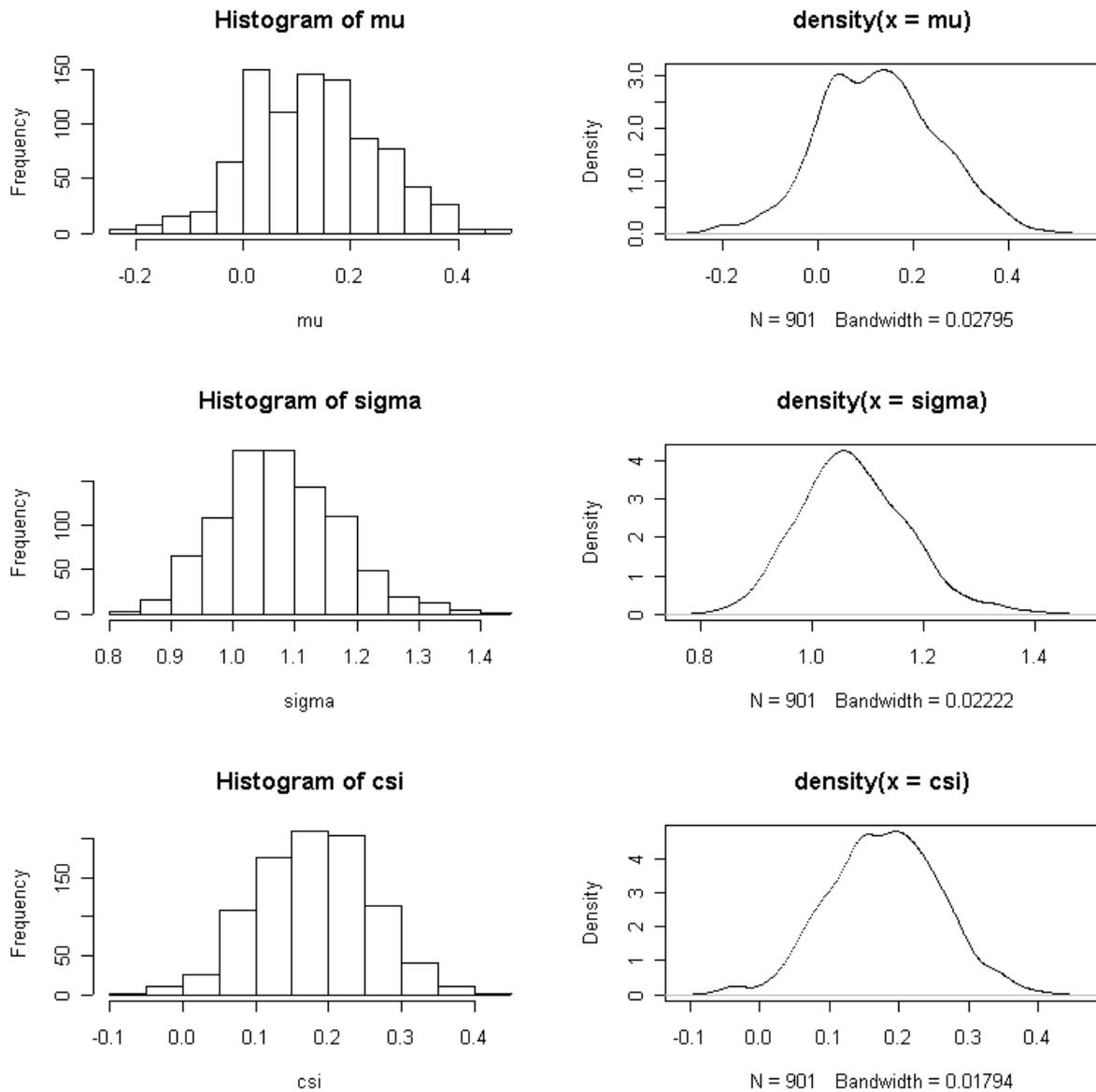


Figura 4.6

L'approssimazione delle distribuzioni a posteriori dei tre parametri, sembrano buone. Il metodo sembra quindi funzionare meglio di IS per numerosità campionarie  $n$  più elevate. Non ci resta ora, che valutare, come si comporta Iterative Importance Sampling.

#### 4.4.3 IIS per valori estremi

Iterative Importance Sampling per i motivi già spiegati e per quanto visto nel capitolo precedente dovrebbe fornire stime migliori rispetto ad IS e MCMC. L'algoritmo è analogo a quello visto nel capitolo 3 eccetto che nel "passo di ricampionamento": sostituiamo all'uniforme, tre normali con media pari a zero e varianze pari a 0.01 e, ad ogni  $j$  iterazione (con  $j=1,\dots,N$ ), ci si avvicina sempre più alla vera distribuzione a posteriori ottenendo campioni sempre più precisi. Per rendere possibile il confronto con i 2 metodi precedenti, manteniamo immutate le altre condizioni iniziali necessarie per far partire l'algoritmo. Prendiamo, sempre, distribuzioni a priori  $\mu \sim N(0, 0.5^2)$ ,  $\phi \sim N(0, 0.5^2)$  e  $\xi \sim N(0, 0.5^2)$  e come campione iniziale lo stesso utilizzato per i metodi precedenti di numerosità uguale a 10 e parametri pari a ( $\mu = 0, \sigma = 1, \xi = 0.2$ ). In figura 4.6 riproponiamo i grafici ottenuti alla  $N=100$  iterazione con  $m$ , come di consueto, pari a 1000.

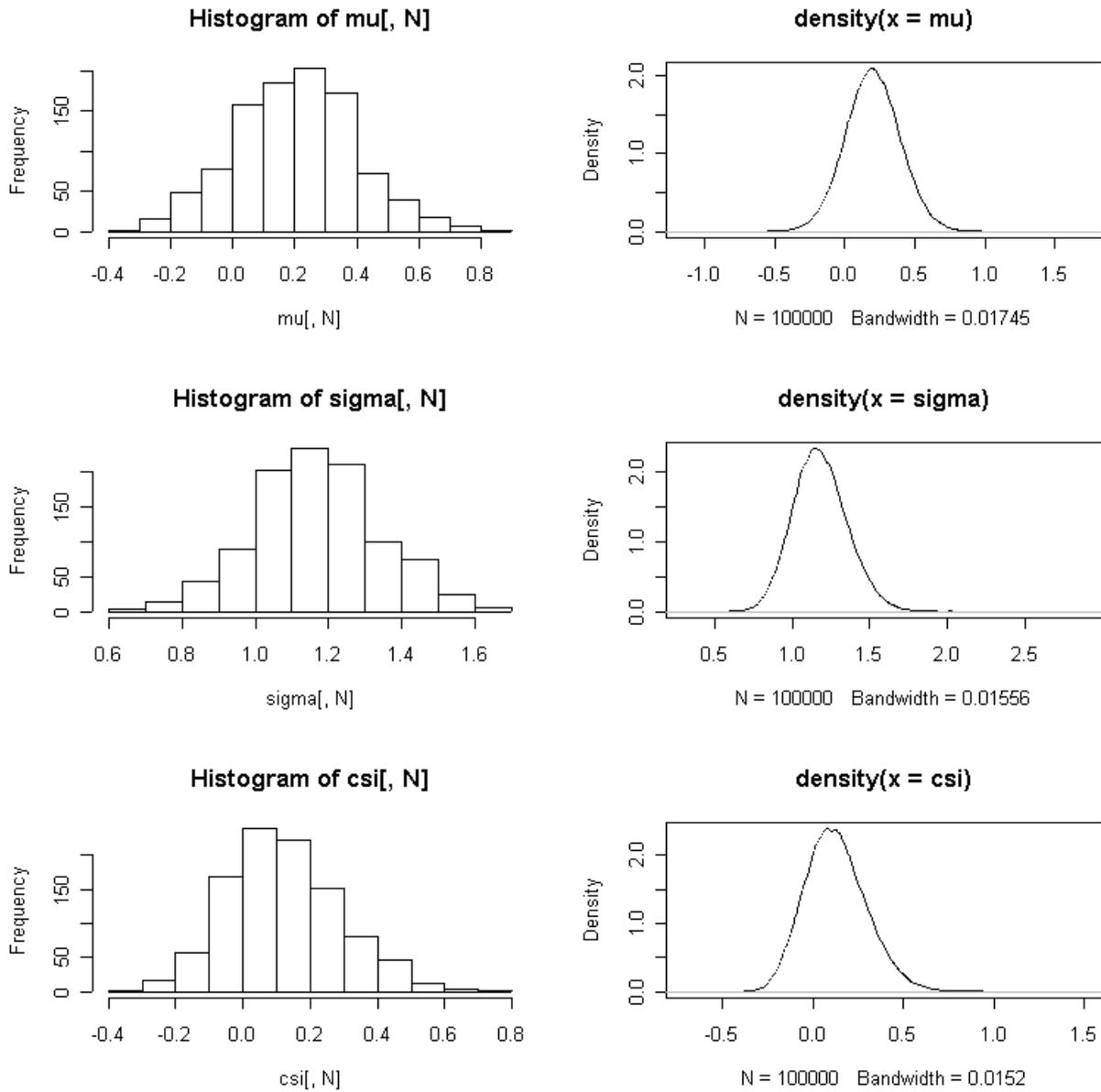


Figura 4.7

L'approssimazione sembra discreta, ma forse MCMC funzionava meglio...anche se il tutto potrebbe essere solo un'impressione grafica. Proviamo ad aumentare la numerosità del campione iniziale passando ad  $n=100$ .

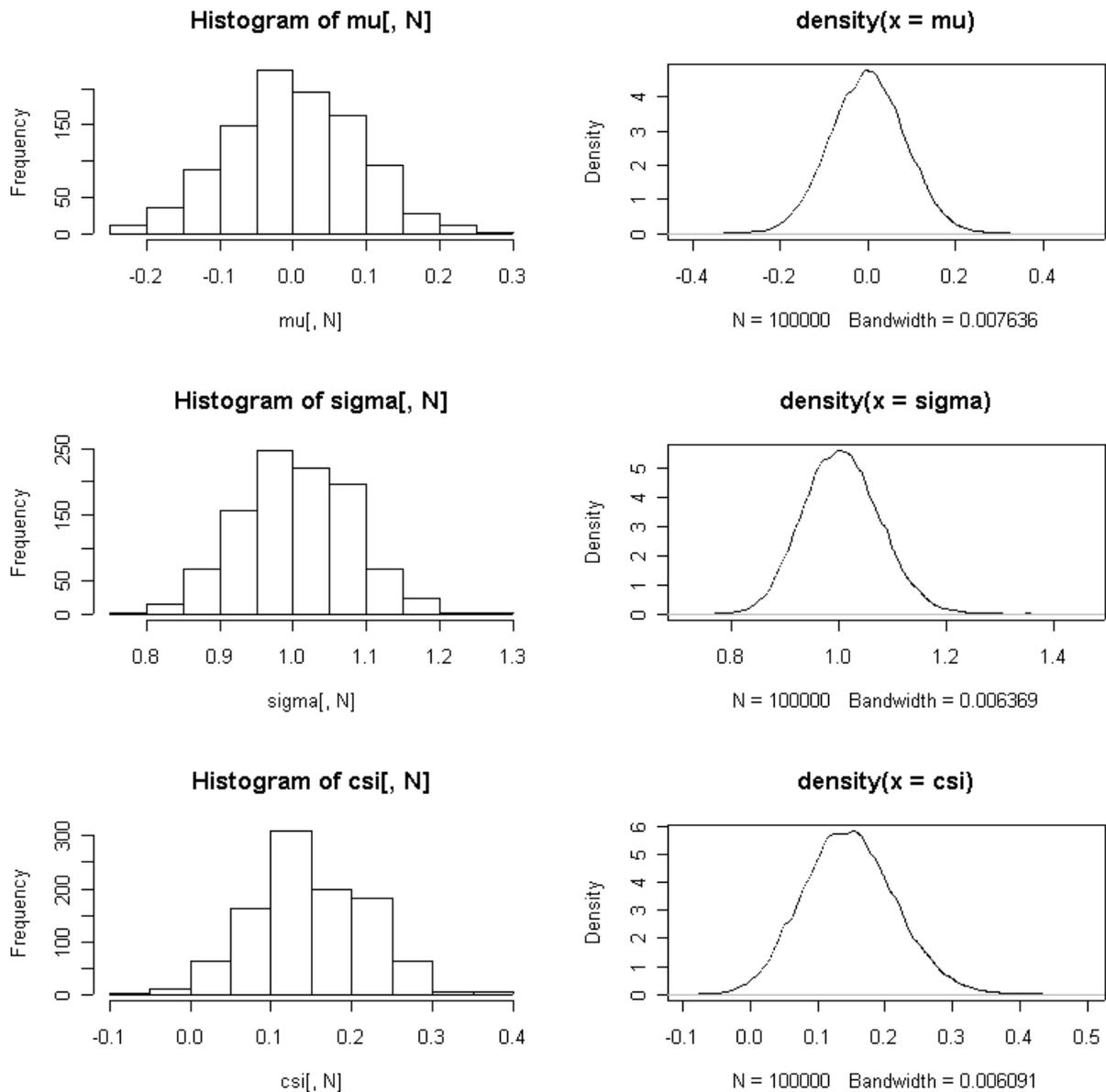


Figura 4.8

In base a quest' analisi grafica preliminare, le stime ottenute per  $n=100$  sembrano essere abbastanza buone.

Ricapitolando, limitandoci all'analisi grafica svolta in questi paragrafi IS sembra funzionare abbastanza bene quando  $n=10$ , anche se sembra presentare qualche problema quando il campione iniziale aumenta di numerosità. Questo potrebbe essere dovuto al fatto che, prendendo una  $f(x | \mu, \sigma, \xi)$  più concentrata, quando si va a campionare con IS la distribuzione a posteriori

$f(\mu, \sigma, \xi | x)$  è molto schiacciata. Anche MCMC dopo aver tolto il burn-in sembra funzionare abbastanza bene per  $n=10$  ma sembra presentare qualche piccolo problema di precisione per numerosità campionarie più elevate. IIS, invece sembra funzionare meglio quando la numerosità campionaria  $n$  inizia a salire. Un'analisi grafica di questo genere, non è però sufficiente a valutare la bontà delle simulazioni ottenute. Dobbiamo cercar di capire di quanto sbagliano gli algoritmi prodotti. Nel paragrafo successivo, sarà nostra cura, cercar di trovare una misura che ci consenta di valutare la “bontà di funzionamento” dell'algoritmo.

#### 4.5 Qual è l'algoritmo migliore?

Il problema è che, in questo caso, a differenza degli esempi riportati nel capitolo 3, non conosciamo la vera distribuzione a posteriori e non possiamo quindi calcolare, gli errori quadratici medi di media e varianza delle stime ottenute a posteriori in  $N$  simulazioni. Una quantificazione dell'errore può essere data da un EQM calcolato nel seguente modo:

$$(1/N) (\sum_i (\mu_i - \mu_{\#})^2), \quad (1/N) (\sum_i (\sigma_i - \sigma_{\#})^2), \quad (1/N) (\sum_i (\xi_i - \xi_{\#})^2);$$

dove  $N$  = numero di campioni simulati dalla densità a posteriori  $f(\mu, \sigma, \xi | x)$ ;  $\mu_i, \sigma_i$  e  $\xi_i$  rappresentano le medie dei 3 parametri simulati di volta in volta e dove  $i=1, \dots, N$ ;  $\mu_{\#}, \sigma_{\#}$  e  $\xi_{\#}$  rappresentano i valori dei parametri del campione iniziale da cui calcoliamo la verosimiglianza  $f(x | \mu_{\#}, \sigma_{\#}, \xi_{\#})$ . A differenza di quanto fatto nel capitolo tre, non sarà qui possibile calcolare l'errore quadratico medio della varianza delle stime ottenute della distribuzione “a posteriori” dato che, come già ricordato più sopra, la a posteriori è appunto non nota. Nel nostro

esempio, lo ripetiamo,  $f(x | \mu_{\#}, \sigma_{\#}, \xi_{\#}) = f(x | 0, 1, 0.2)$ . Nella figura 4.9, seguente, riportiamo gli errori quadratici medi ottenuti con  $N=1000$ ,  $n=10$  (numerosità campione simulato dalla a priori),  $m=1000$  (numerosità campioni di volta in volta simulati);  $\mu_{\#}=0, \sigma_{\#}=1, \xi_{\#}=0.2$ ; distribuzioni a priori  $\mu \sim N(0, 0.5^2)$ ,  $\phi \sim N(0, 0.5^2)$ ,  $\xi \sim N(0, 0.5^2)$ ; per IIS e MCMC (il passo di campionamento è del tipo)  $\varepsilon_{\mu} \sim N(0, 0.01)$ ,  $\varepsilon_{\sigma} \sim N(0, 0.01)$  e  $\varepsilon_{\xi} \sim N(0, 0.01)$ ; solo per MCMC  $\mu_0=0, \sigma_0=2, \xi_0=1$ ;

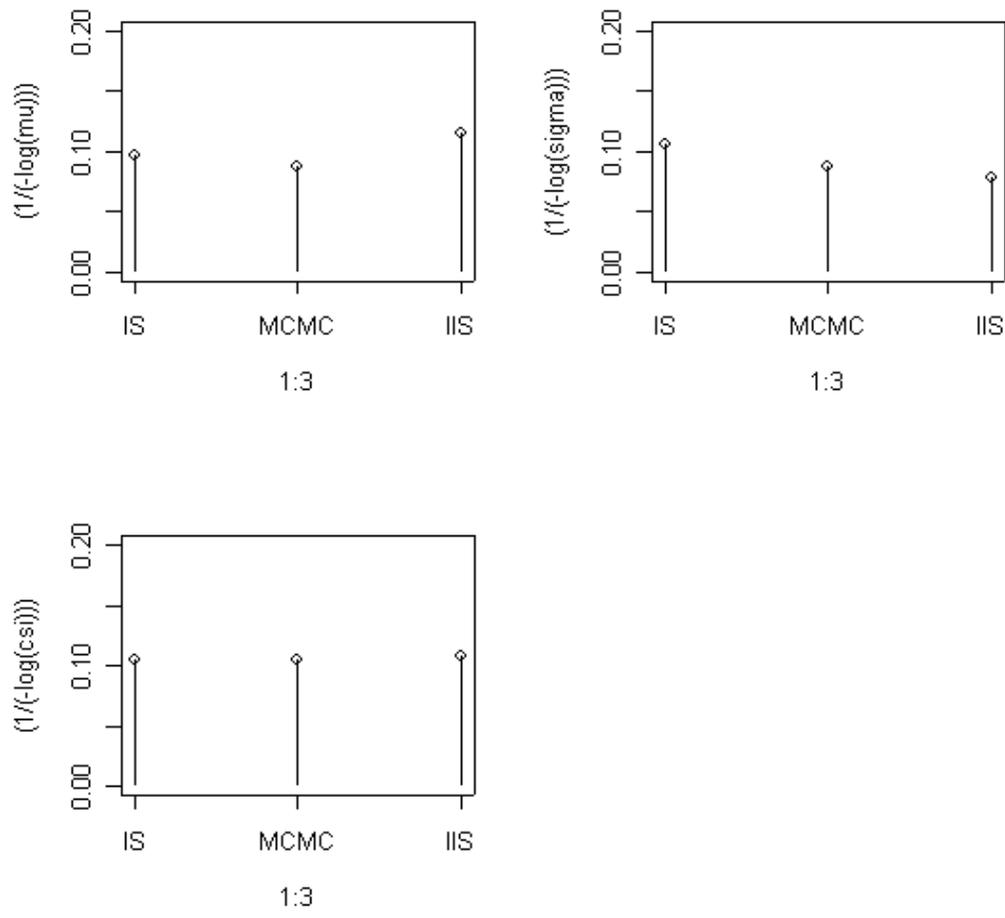


Figura 4.9

Nei grafici di figura 4.9 si può notare come non sembrano esservi differenze significative negli EQM calcolati per tutti e 3 i parametri. Questo ci può far comprendere come le conclusioni espresse nei paragrafi precedenti di questo

capitolo fossero un po' affrettate (gli EQM calcolati per  $\xi$ , ad esempio, sembrano più o meno uguali con tutti e 3 i metodi). Non risulta esserci un metodo univocamente migliore. Ricordiamo, inoltre, che per costruire i grafici precedenti e permetterne la leggibilità, abbiamo dovuto effettuare una trasformazione del tipo  $1/(-\log(\theta))$  dato che tutti i valori dell'Errore Quadratico Medio erano poco più elevati di zero (siamo nell'ordine di 4-6 cifre decimali dopo lo zero). Proviamo, ora, a cambiare le condizioni iniziali del modello per vedere se le conclusioni in termini di EQM cambiano. Ipotizziamo che, a parità di tutte le condizioni iniziali più sopra specificate, incrementi la numerosità del campione simulato dalla a priori ( $n=100$ ).

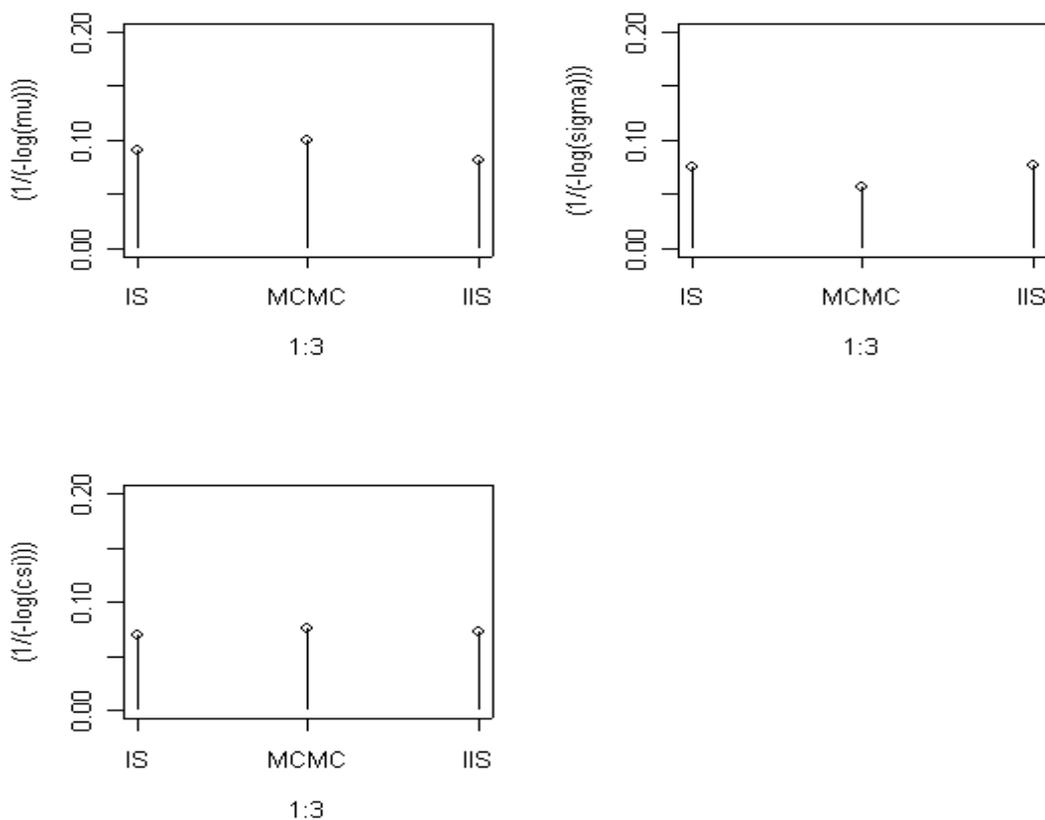


Figura 4.10

Anche cambiando  $n$  (figura 4.10) i tre algoritmi di simulazione, sembrano portare a stime abbastanza equivalenti (anche qui, non si nota un algoritmo che

sia univocamente migliore...). Abbiamo provato a cambiare anche l'insieme dei valori iniziali di MCMC (0.3,1.3,0.5) avvicinandoli ai veri valori della a priori(0,1,0.2) ma la cose non sembrano cambiare significativamente (c'è solo una convergenza più rapida delle catene markoviane create). Anche aumentando a dismisura  $N$  e le varianze del già citato "passo di campionamento", le stime e i relativi EQM rimangono analoghi per tutti e tre i parametri non riuscendo ad individuare l'algoritmo in ogni situazione migliore. In definitiva, i tre metodi sembrano portare a stime molto simili, ammettendo circa la stessa "quota" di errore.

## CAPITOLO 5

# SIMULAZIONE E STATISTICA BAYESIANA PER DATI TEMPORALI

In questo capitolo, analizzeremo come le tecniche di simulazione fin qui studiate possano essere utili anche per la simulazione in ambiti in cui ci sia dipendenza di tipo temporale nei dati e si possano specificare distribuzioni a priori per i parametri.

### *5.1 Introduzione*

Sono molteplici le situazioni in cui siamo interessati a studiare fenomeni la cui componente di primaria importanza è data dal tempo. La quotazione di un titolo sul mercato borsistico, la serie storica delle vendite di un determinato prodotto o il livello giornaliero dello smog rilevato in una città, sono tutti fenomeni in cui la collocazione temporale è di fondamentale importanza. Ad esempio, la quotazione di un titolo al tempo  $t$  può dipendere dalle quotazioni ai tempi  $t-1, t-2, \dots$  ecc. In statistica, la modellazione di questi fenomeni risulta di fondamentale importanza. La costruzione e stima di un modello adatto a “seguire” i dati è un elemento essenziale per prevederne il comportamento futuro. In questo capitolo, per “lisciare” e modellare i dati, utilizzeremo 2 dei metodi utilizzati in quest’elaborato: Iterative Importance Sampling (IIS) e Markov Chain Montecarlo (MCMC). Abbiamo scelto questi due metodi

anziché altri (ad esempio, il modello di regressione lineare semplice, i modelli della classe ARIMA  $(p,d,q)$ , ecc...) perché ci permettono di sfruttare eventuali specificazioni bayesiane (oltre che per proseguire ed approfondire l'analisi iniziata nei capitoli precedenti). Infatti, attraverso una specificazione bayesiana (tramite l'assunzione di una distribuzione a priori), valuteremo quale tra i due metodi di simulazione prescelti, funzioni meglio producendo stime più precise. Per valutarne la bontà, utilizzeremo dati simulati in modo da poter confrontare le stime ottenute tramite i due metodi con i valori "teorici" iniziali dei parametri. Dopo aver fatto questo, forniremo un'applicazione dei due metodi ad una situazione con dati veri.

## 5.2 Simulazione campionaria

La situazione che si presenta di solito davanti al ricercatore è che, a partire da un'informazione campionaria iniziale del tipo  $y_1, y_2, \dots, y_t, \dots, y_n$  si cerca di stimare un modello tale da poter cogliere l'andamento di fondo nei dati disponibili. Una delle possibili scelte è costruire una serie di parametri  $\theta_1, \theta_2, \dots, \theta_t, \dots, \theta_n$  che permettano di "filtrare" la serie. Tramite MCMC e IIS, infatti, realizzeremo una sorta di kalman filter non gaussiano (per un approfondimento si veda *Harvey, 1990*).

In un'ottica di simulazione, invece, ipotizziamo di conoscere la serie dei parametri  $\theta_1, \theta_2, \dots, \theta_t, \dots, \theta_n$  e di essere interessati ad ottenere i dati  $y_1, y_2, \dots, y_t, \dots, y_n$ . Ipotizziamo che la distribuzione dei  $\theta_t$  sia del tipo Random Walk e che, quindi:

$$\theta_t = \theta_{t-1} + \varepsilon_t \quad (i)$$

assumendo che  $\theta_1 \sim N(0, \sigma^2)$  e  $\varepsilon_t \sim WN(0, \sigma^2)$ . Un'altra possibile specificazione può essere la seguente (tipica di un processo autoregressivo del primo ordine):

$$\theta_t = \alpha \theta_{t-1} + \varepsilon_t \quad (\text{ii})$$

sempre con  $\theta_1 \sim N(0, \sigma^2)$  e  $\varepsilon_t \sim WN(0, \sigma^2)$  e, in aggiunta,  $|\alpha| < 1$  per rendere il processo stazionario. La (i) e la (ii) sono due semplici metodi (molte altre scelte potevano essere plausibili) che ci permettono di ottenere una serie  $\theta_1, \theta_2, \dots, \theta_t, \dots, \theta_n$ . Una volta simulata la serie dei  $\theta_t$ , ipotizzando che:

$$y_t \sim \text{Poisson}(\exp(\theta_t))$$

si ottiene facilmente l'informazione campionaria  $y_1, y_2, \dots, y_t, \dots, y_n$ . (la scelta di una Poisson come distribuzione di riferimento per ottenere gli  $y_t$ , è una delle scelte più frequenti con dati di questo tipo). Come si vede chiaramente in Figura 5.1 la serie dei  $\theta_t$  ottenuti (indicati con la linea) sembra modellare abbastanza bene i dati simulati  $y_t$  (pallini) in corrispondenza di tutti e due i modelli. Ciò non ci stupisce affatto dato che gli  $y_t$  vengono generati direttamente dai  $\theta_t$ . Per realizzare le simulazioni riportate in figura 5.1, abbiamo posto  $n=100$ ,  $\alpha=0.95$  e  $\sigma=0.5$ .

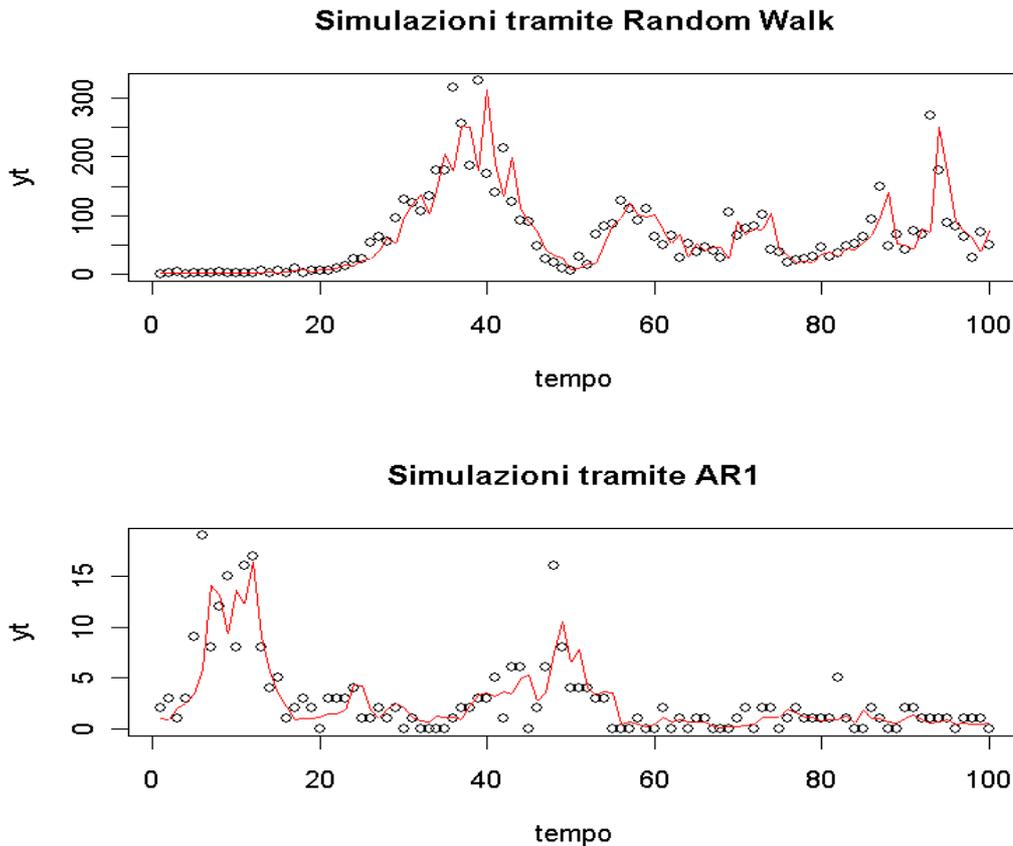


Figura 5.1

### 5.3 Un confronto tra MCMC e IIS per filtraggio

Il nostro obiettivo è qui quello di modellare una serie di dati  $y_1, y_2, \dots, y_t, \dots, y_n$  attraverso una serie di parametri  $\theta_1^*, \theta_2^*, \dots, \theta_t^*, \dots, \theta_n^*$  ottenuti, utilizzando i due algoritmi di simulazione prescelti (MCMC e IIS). Sarà possibile valutare direttamente quale dei due metodi funzioni meglio, semplicemente confrontando, elemento per elemento, la serie di parametri  $(\theta_1^*, \theta_2^*, \dots, \theta_t^*, \dots, \theta_n^*)$  con la serie dei valori “originari”  $\theta_1, \theta_2, \dots, \theta_t, \dots, \theta_n$  (la serie di parametri “generatrice” della serie “campionaria”  $y_1, y_2, \dots, y_t, \dots, y_n$ ). Ipotizziamo quindi di voler ottenere la serie  $\theta_1^*, \theta_2^*, \dots, \theta_t^*, \dots, \theta_n^*$  partendo dall’informazione campionaria  $y_1, y_2, \dots, y_t, \dots, y_n$  ottenuta simulando attraverso

un AR(1) la serie  $\theta_1, \theta_2, \dots, \theta_t, \dots, \theta_n$  (ii, nel paragrafo 5.2) e in un'impostazione bayesiana specificando una distribuzione a priori.

### 5.3.1 MCMC per filtraggio

L'obiettivo che ci proponiamo con MCMC è di creare una catena markoviana  $n$ -dimensionale  $\theta_N$  del tipo:

$$\begin{aligned} &(\theta_1^1, \theta_2^1, \dots, \theta_n^1) \\ &(\theta_1^2, \theta_2^2, \dots, \theta_n^2) \\ &(\theta_1^3, \theta_2^3, \dots, \theta_n^3) \\ &\vdots \\ &(\theta_1^M, \theta_2^M, \dots, \theta_n^M) \end{aligned}$$

dove, partendo da una serie di valori iniziali  $\theta_1^1, \theta_2^1, \dots, \theta_n^1$  (nel nostro caso, tutti posti pari ad uno) aggiorniamo di volta in volta i valori dei parametri. Al secondo passo attraverso MCMC otterremo la serie di parametri  $\theta_1^2, \theta_2^2, \dots, \theta_n^2$ ; al terzo passo  $\theta_1^3, \theta_2^3, \dots, \theta_n^3$  e così via fino ad ottenere la serie  $\theta_1^M, \theta_2^M, \dots, \theta_n^M$ . Ad esempio, al secondo passo per creare la serie  $\theta_1^2, \theta_2^2, \theta_3^2, \dots, \theta_n^2$ :

- i. Determiniamo  $\theta_1^\# = \theta_1^1 + u$  (con  $u \sim \text{Uniforme}(-\varepsilon, +\varepsilon)$ );
- ii. Determiniamo  $\alpha$  come il  $\min \{1, f(\theta_1^\# | y_1, \theta_2^\#) / f(\theta_1^1 | y_1, \theta_1^2)\}$ ;
- iii. Campioniamo  $\theta_1^2$  come  $\theta_1^2 = \theta_1^\#$  con la probabilità  $\alpha$  o come  $\theta_1^2 = \theta_1^1$  con probabilità  $1-\alpha$ ;
- iv. Determiniamo  $\theta_2^\# = \theta_2^1 + u$  (con  $u \sim \text{Uniforme}(-\varepsilon, +\varepsilon)$ );
- v. Determiniamo  $\alpha$  come il  $\min \{1, f(\theta_2^\# | y_2, \theta_1^\#, \theta_3^\#) / f(\theta_2^2 | y_2, \theta_2^1, \theta_2^3)\}$ ;

- vi. Campioniamo  $\theta_2^2$  come  $\theta_2^2 = \theta_2^\#$  con la probabilità  $\alpha$  o come  $\theta_2^2 = \theta_2^1$  con probabilità  $1-\alpha$ ;
- vii. Ripetiamo i punti dal (iv) al (vi) in modo da ottenere le stime  $\theta_3^2, \dots, \theta_{n-1}^2$ ;
- viii. Determiniamo  $\theta_n^\# = \theta_n^1 + u$  (con  $u \sim$  Uniforme  $(-\epsilon, +\epsilon)$ );
- ix. Determiniamo  $\alpha$  come il  $\min \{1, f(\theta_n^\# | y_n, \theta_{n-1}^\#) / f(\theta_n^1 | y_n, \theta_{n-1}^1)\}$ ;
- x. Campioniamo  $\theta_n^2$  come  $\theta_n^2 = \theta_n^\#$  con la probabilità  $\alpha$  o come  $\theta_n^2 = \theta_n^1$  con probabilità  $1-\alpha$ ;

In modo analogo si agisce negli altri  $M-2$  passi ottenendo le altre  $M-2$  “serie”. Per il calcolo dei pesi  $\alpha$ , si utilizzano le distribuzioni a posteriori dei parametri che si dimostrano essere:

$$\begin{aligned}
 f(\theta_1 | y_1, \theta_2) &\propto f(\theta_1) f(y_1 | \theta_1) f(\theta_2 | \theta_1) \\
 f(\theta_2 | y_2, \theta_1, \theta_3) &\propto f(\theta_2 | \theta_1) f(y_2 | \theta_2) f(\theta_3 | \theta_2) \\
 f(\theta_3 | y_3, \theta_2, \theta_4) &\propto f(\theta_3 | \theta_2) f(y_3 | \theta_3) f(\theta_4 | \theta_3) \\
 &\vdots \\
 f(\theta_n | y_n, \theta_{n-1}) &\propto f(\theta_n | \theta_{n-1}) f(y_n | \theta_n)
 \end{aligned}$$

Dopo aver visto come funziona MCMC, proviamo ora a vedere come si comporta con dati “temporali” del tipo  $y_1, y_2, \dots, y_t, \dots, y_n$ . Partiamo prima con un esempio con dati campionari simulati  $(y_1, y_2, \dots, y_t, \dots, y_n)$  in modo da poter confrontare le stime ottenute  $(\theta_1^*, \theta_2^*, \dots, \theta_t^*, \dots, \theta_n^*)$  con i veri valori  $\theta_1, \theta_2, \dots, \theta_t, \dots, \theta_n$  da cui abbiamo ottenuto (come indicato nel paragrafo 5.2, utilizzando un autoregressivo del primo ordine) il campione  $y_1, \dots, y_n$  (a questo scopo, indicheremo da qui in avanti indicheremo le stime ottenute con  $\theta_t^*$  e i valori originari con  $\theta_t$ ). Per la simulazione tramite l’AR(1) abbiamo posto

$\alpha=0.95$ ,  $\sigma=1$  e  $n=200$  (è semplicemente una scelta, altri valori dei parametri potevano comunque essere plausibili) ottenendo la serie  $\theta_1, \dots, \theta_{200}$  e il campione  $y_1, \dots, y_{200}$ . Da questo campione, combinando l'informazione campionaria con quella disponibile a priori (seguendo un'approccio di tipo bayesiano) tramite MCMC, otteniamo le stime a posteriori della catena markoviana n-dimensionale creata (ottenendo una serie per tutti i  $\theta_1^*, \dots, \theta_t^*, \dots, \theta_{200}^*$ ). In figura 5.2, riportiamo i diagrammi di dispersione e gli istogrammi ottenuti per  $\theta_1^*, \theta_{50}^*, \theta_{125}^*$  e  $\theta_{200}^*$  ponendo  $\alpha=0.95$ ,  $\sigma=1$ ,  $n=200$ ,  $M=2200$  (con un burn in di  $m=200$  che non consideriamo)  $f(\theta_1^*) \sim N(0,1)$  ed  $\varepsilon=2$ . Per problemi di spazio, riportiamo qui di seguito, solo le stime di  $\theta_1^*, \theta_{50}^*, \theta_{125}^*$  e  $\theta_{200}^*$ ; stime che, peraltro ci danno già un'idea sulla bontà di "funzionamento" dell'algorithm.

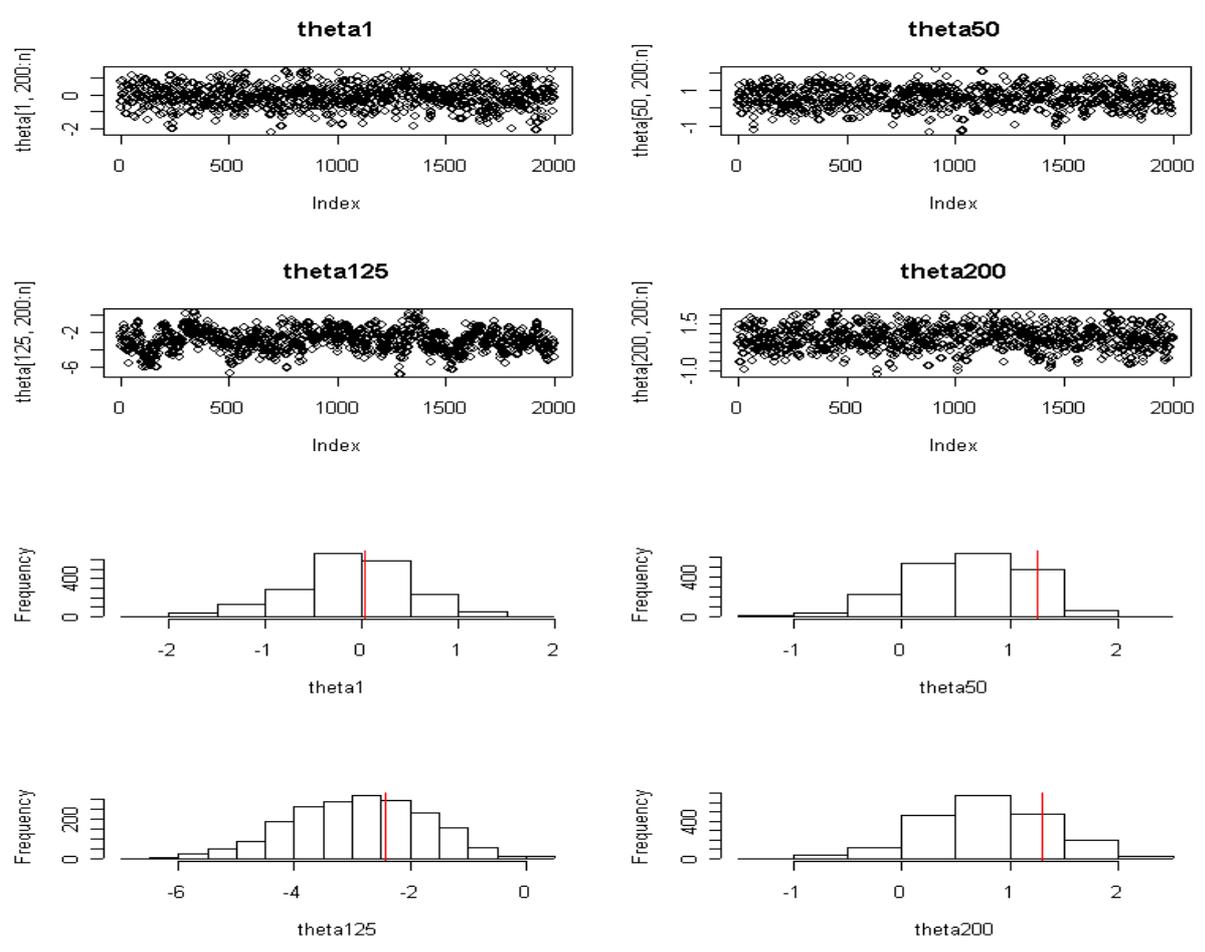


Figura 5.2

Le catene ottenute per  $\theta_1^*$ ,  $\theta_{50}^*$ ,  $\theta_{125}^*$  e  $\theta_{200}^*$  se confrontate con i “veri” valori  $\theta_1=0.02$ ,  $\theta_{50}=1.25$ ,  $\theta_{125}=(-2.44)$  e  $\theta_{200}=1.29$  sembrano abbastanza buone dato che sembrano centrate attorno ai veri valori. Una rappresentazione più chiara e completa la si può vedere in figura 5.3 con il grafico di dispersione di  $\theta_1, \theta_2, \dots, \theta_t, \dots, \theta_n$  (thetateorico, indicati sulle ascisse) e  $\theta_1^*, \theta_2^*, \dots, \theta_t^*, \dots, \theta_n^*$  (thetastimato, indicati sulle ordinate). In figura 5.3, ogni  $\theta_t^*$  viene calcolato come media aritmetica delle  $M$ - $m$  simulazioni effettuate (2200-200) di passo in passo:

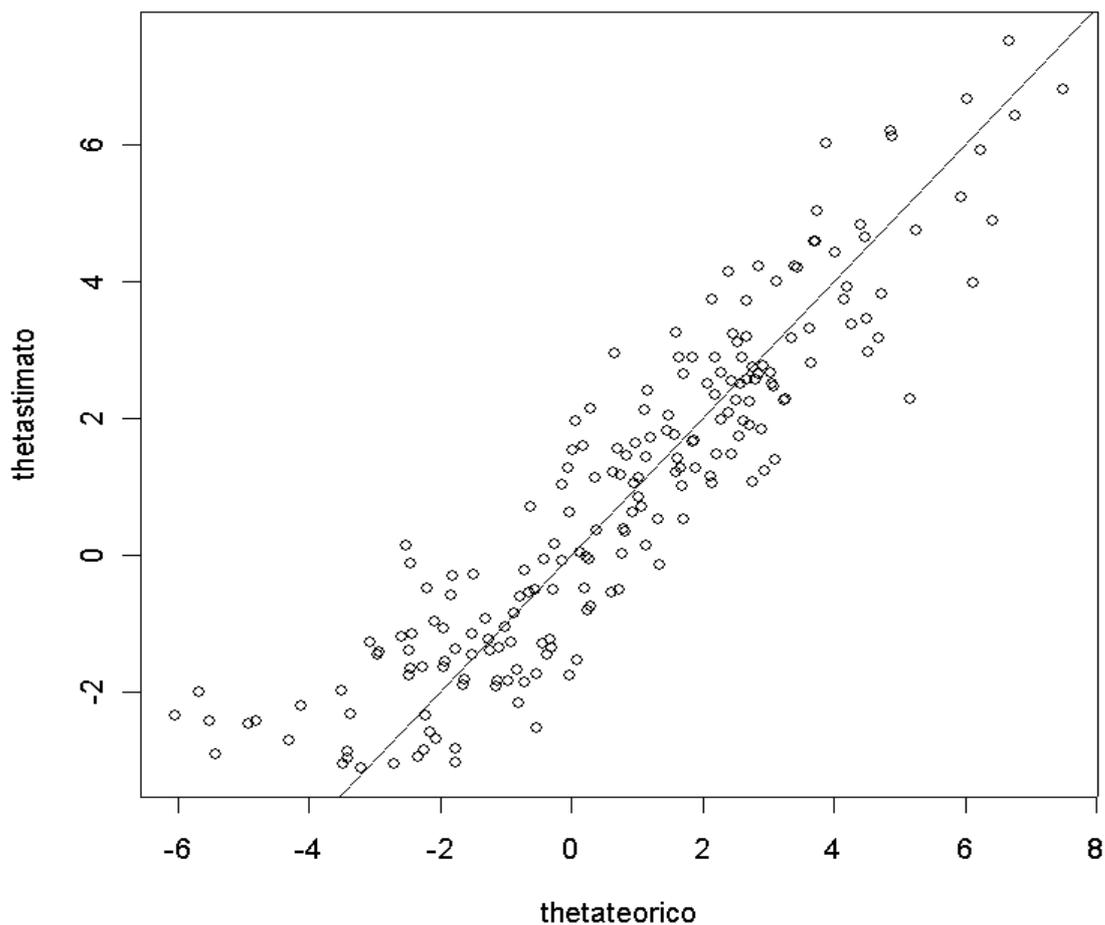


Figura 5.3

Come si vede chiaramente in figura 5.3, le cose sembrano andare abbastanza bene: l'approssimazione tra i  $\theta$  e i  $\theta^*$  è infatti buona, dato che, i punti, tendono a disporsi attorno alla bisettrice (anche se, forse, c'è qualche problema di approssimazione per valori bassi sia di  $\theta$  che di  $\theta^*$ ). Proveremo a vedere nel prossimo paragrafo (5.3.2) se ci sono significative differenze nel “comportamento” delle simulazioni ottenute con IIS.

### 5.3.2 Iterative Importance Sampling per filtraggio

Dopo aver visto come si comporta MCMC in quest'ambito, non ci resta che valutare come “reagisce” IIS. Abbiamo già ampiamente discusso nel capitolo due come IIS sia di solito preferibile a MCMC. Per costruzione, infatti, IIS “eredita” quanto di buono Importance Sampling (IS) e MCMC permettono di fare. IIS lavora, infatti, in due stadi: al primo si implementa un Importance Sampling (IS) classico; al secondo si costruisce una sorta di MCMC che viene iterato più volte. E' proprio questa combinazione di metodi che permette di solito ad IIS di essere migliore. Vedremo, ora, se questa preferenza per IIS vale anche per problemi di “filtraggio”. L'obiettivo che ci si prefigge è sempre lo stesso: partendo dalla stessa serie di parametri simulata  $\theta_1, \theta_2, \dots, \theta_t, \dots, \theta_n$  (e quindi stesso campione  $y_1, y_2, \dots, y_t, \dots, y_n$  utilizzato nel paragrafo 5.3.1) vogliamo ottenere una serie di parametri  $\theta_1^*, \theta_2^*, \dots, \theta_t^*, \dots, \theta_n^*$  che ben approssimi  $\theta_1, \theta_2, \dots, \theta_t, \dots, \theta_n$ .

L'algoritmo visto nel capitolo due, può essere qui aggiornato nel seguente modo (mantenendo la struttura a due stadi lì definita):

## 1° Stadio

- (i) Simulazione diretta dalla a priori di  $\theta_1$  (che ipotizziamo essere una  $N(0, \sigma^2)$ ): otteniamo il campione  $\theta_{11}^{**}, \theta_{12}^{**}, \dots, \theta_{1t}^{**}, \dots, \theta_{1M}^{**}$ ;
- (ii) Campioniamo con pesi pari a  $w_t = f(y_1 | \exp(\theta_{1t}^{**}))$  i valori  $\theta_{11}^*, \theta_{12}^*, \dots, \theta_{1t}^*, \dots, \theta_{1M}^*$  ottenendo un campione della a posteriori  $\theta_1 | y_1$ ;

## 2° Stadio

- (i) Applichiamo l'equazione di transizione  $\theta_{2t}^{**} = \alpha \theta_{1t}^* + \varepsilon_t$  (con  $\varepsilon \sim N(0, \sigma^2)$ ) in modo da ottenere la serie di parametri  $\theta_{21}^{**}, \theta_{22}^{**}, \dots, \theta_{2t}^{**}, \dots, \theta_{2M}^{**}$ ;
- (ii) Campioniamo con pesi pari a  $w_t = f(y_2 | \exp(\theta_{2t}^{**}))$  i valori  $\theta_{21}^*, \theta_{22}^*, \dots, \theta_{2t}^*, \dots, \theta_{2M}^*$  ottenendo un campione della a posteriori  $\theta_1 | y_1, y_2$ ;
- (iii) Ripetiamo i passi (i) e (ii) del secondo stadio ( $n-2$  volte) in modo da ottenere campioni della distribuzione a posteriori  $f(\theta | y)$  di tutti i parametri  $\theta_1^*, \theta_2^*, \dots, \theta_t^*, \dots, \theta_n^*$ ;

L'applicazione del metodo sopra descritto, porta ai diagrammi di dispersione e agli istogrammi riportati in figura 5.4. Per realizzarli, abbiamo utilizzato lo stesso campione simulato  $y_1, y_2, \dots, y_t, \dots, y_n$  impiegato con MCMC, abbiamo poi posto  $\alpha=0.95$ ,  $\sigma=1$ ,  $n=200$  e  $M=2000$  (questo, naturalmente, per poter confrontare direttamente i due metodi).

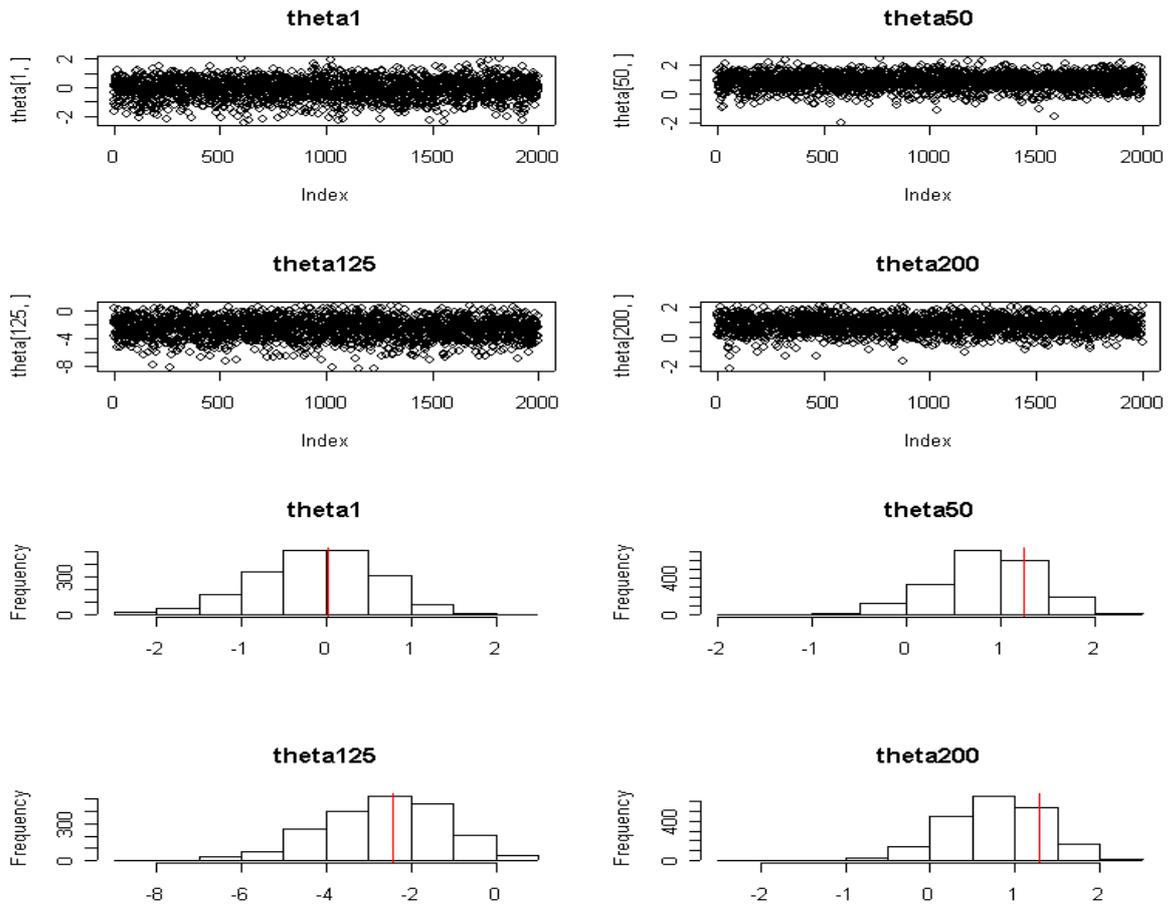


Figura 5.4

Anche qui, abbiamo dato rappresentazione solo dei campioni ottenuti per  $\theta_1^*, \theta_{50}^*, \theta_{125}^*$  e  $\theta_{200}^*$ . L'approssimazione ottenuta per questi  $\theta^*$  è buona se la confrontiamo con i valori di  $\theta_1, \theta_{50}, \theta_{125}$  e  $\theta_{200}$  (che risultano essere uguali, rispettivamente, a 0.02, 1.25, -2.44 e 1.29). Un grafico sicuramente più interessante è quello di figura 5.5: nel diagramma di dispersione ottenuto incrociando i “veri” valori  $\theta_1, \theta_2, \dots, \theta_t, \dots, \theta_{200}$  (ascisse) con i valori stimati  $\theta_1^*, \theta_2^*, \dots, \theta_t^*, \dots, \theta_{200}^*$  (ordinate) si vede come l'approssimazione sia molto buona anche per valori negativi dei  $\theta$  e  $\theta^*$  (con MCMC sembravano esserci un pò di problemi per valori dei  $\theta$  e  $\theta^*$  negativi)

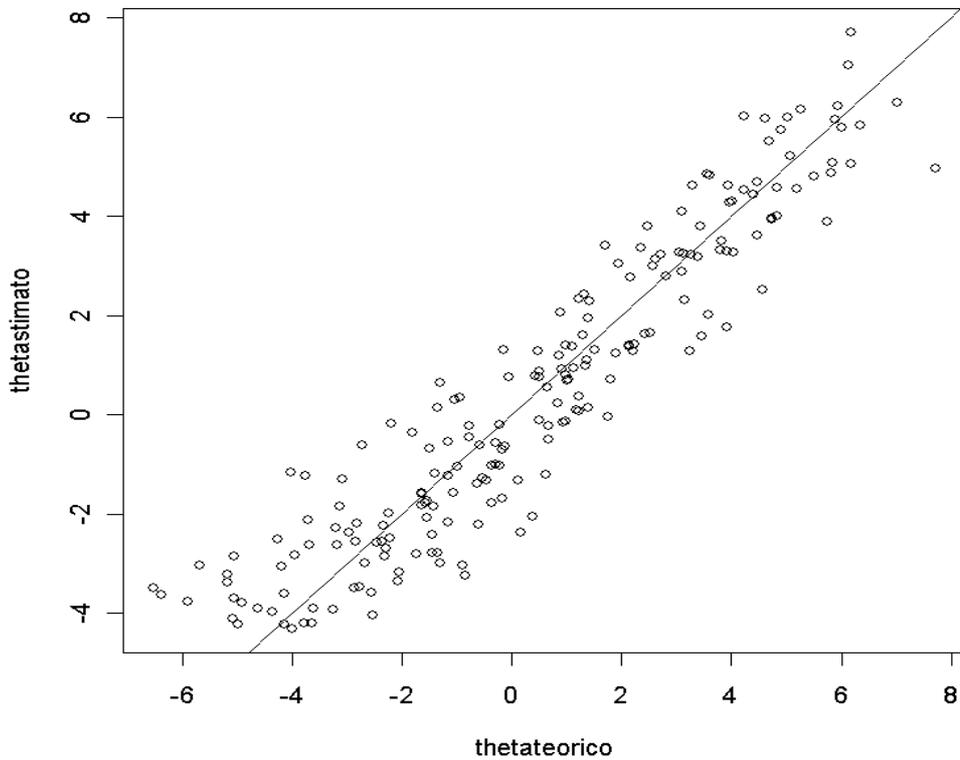


Figura 5.5

Dopo questa prima analisi grafica IIS sembrerebbe funzionare meglio. Ma vediamo se questa nostra prima impressione, viene confermata anche numericamente.

### 5.3.4 Un confronto tra MCMC e IIS

Per dare una “quantificazione” dell’errore commesso dai due metodi, utilizzeremo ancora l’Errore Quadratico Medio (EQM) (che consiste nel reiterare N volte ogni algoritmo nel suo complesso); che in quest’ambito viene calcolato nel seguente modo:

$$EQM(\theta_1, \dots, \theta_n) = (1/(N*n)) \sum_z^N \sum_i^n (\theta_{iz}^* - \theta_i)^2$$

Dove  $\theta_i^* = (1/m)\sum_j^m(\theta_{ij}^\#)$  (con  $\theta_{ij}^\#$  indichiamo la  $j$ -esima stima ottenuta per l' $i$ -iesimo parametro con  $i=1,\dots,n$  e  $j=1,\dots,m$ ); con  $\theta_1,\dots,\theta_i,\dots,\theta_n$  indichiamo la serie di parametri simulati in partenza con cui fare il confronto; con  $\theta_{iz}^*$  la  $i$ -esima media ottenuta nella simulazione  $z$  (con  $z=1,\dots,N$ ). Ponendo  $\alpha=0.95$ ,  $\sigma=1$ ,  $n=200$ ,  $m=2000$ ,  $N=10$  ed  $\varepsilon=2$  abbiamo costruito la Figura 5.6. Nel grafico, si nota come l'errore commesso da IIS sia minore, anche se non di molto, rispetto a quello commesso con MCMC. Sebbene di poco, quindi, l'algoritmo migliore sembra IIS (abbiamo semplicemente confermato quanto ci aspettavamo). La nostra analisi, ha comunque un limite: a causa di alcune difficoltà computazionali abbiamo dovuto tenere un valore di  $N$  (10) piuttosto basso. La presenza, quindi, di un campione simulato  $\theta_{iz}^*$  un po' strano potrebbe cambiare completamente le nostre conclusioni. Per ovviare a questo, abbiamo comunque ripetuto molte volte la procedura e in tutti i casi tranne uno le stime ottenute con IIS sembrano le migliori.

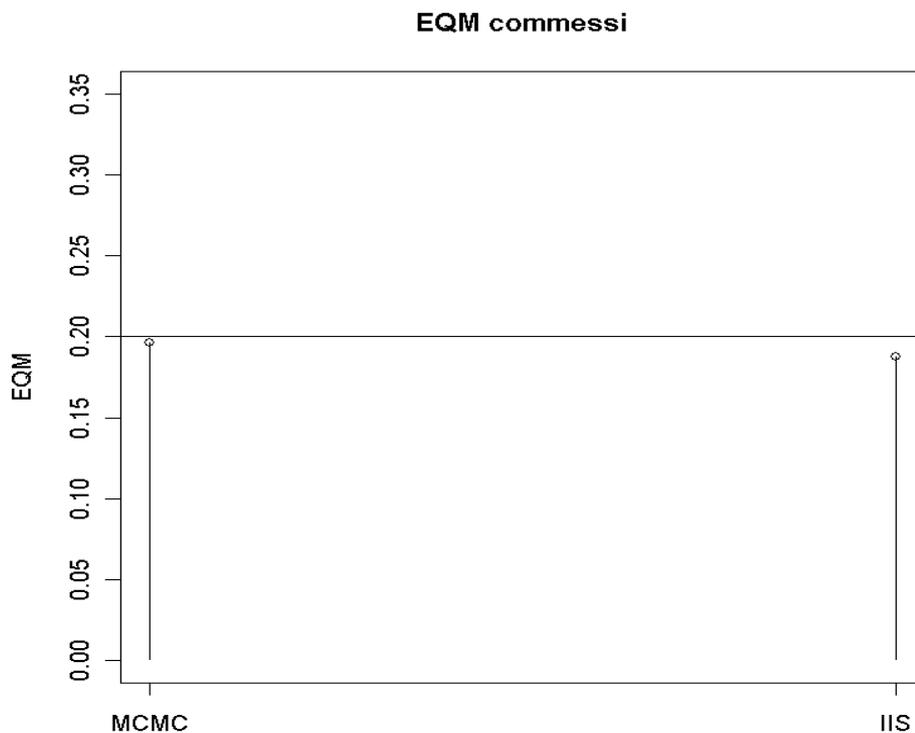


Figura 5.6

## 5.4 Un'analisi con dati reali

Fin qui, abbiamo lavorato sempre con dati simulati. Vediamo come si comportano i nostri algoritmi di simulazione nel lisciare dati reali. Utilizziamo la serie storica (si veda figura 5.7 qui sotto) dei goal realizzati nella serie A italiana nelle stagioni 2001-2002, 2002-2003, 2003-2004 e 2004-2005 (solo le prime trentaquattro giornate) per un totale di 136 giornate di campionato. Ipotesi essenziale per effettuare quest'analisi è data dal fatto che, il numero di gol segnati di giornata in giornata, abbia distribuzione di Poisson (per approfondimenti, si veda *Dixon e Coles, 1997*). Ipotizziamo ora, di esser interessati a valutare, la presenza di un certo fattore stagionale nel numero di goal segnati di giornata in giornata. Alcuni scommettitori, ipotizzano che si segni di più a fine campionato. Proveremo a lisciare la serie storica per vedere se effettivamente è presente questa ciclicità.

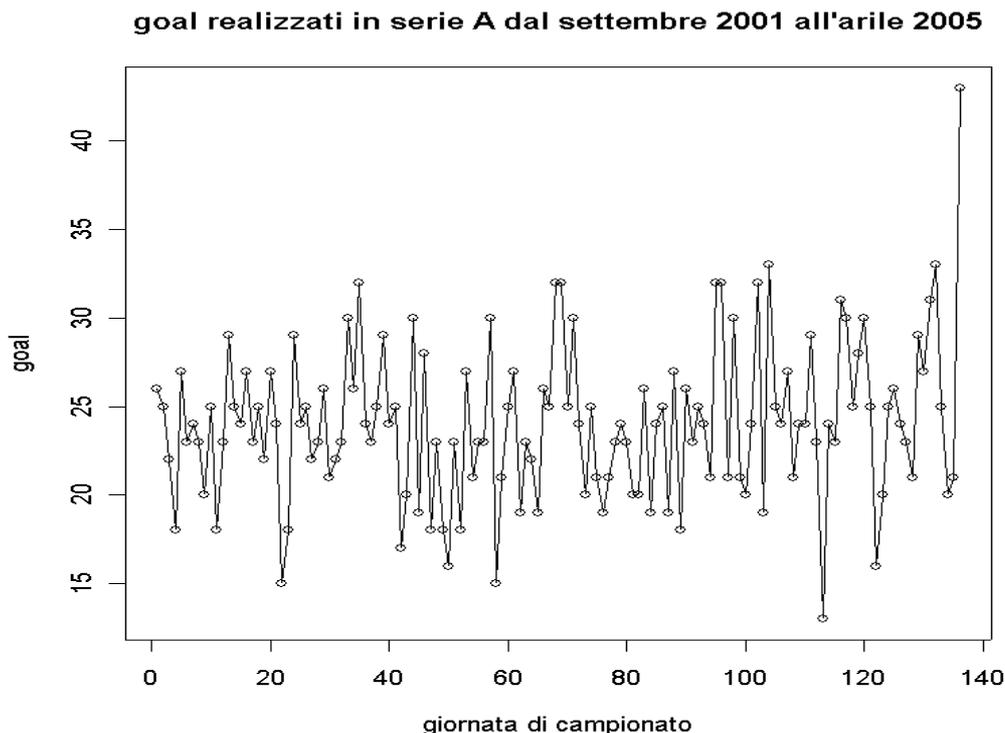


Figura 5.7

Guardando la serie storica così come riportata in figura 5.7 è abbastanza difficile cogliere eventuali ciclicità. Come più sopra già detto bisognerà cercare di lisciare i dati in modo da poterne capire l'andamento di fondo.

#### 5.4.1 Lisciamento con Markov Chain of Montecarlo e IIS

L'obiettivo è quello di ottenere la serie dei parametri di lisciamento  $\theta_1^*, \theta_2^*, \dots, \theta_{136}^*$  utilizzando il campione  $y_1, y_2, \dots, y_{136}$ .

In figura 5.8, riportiamo i grafici della serie storica "lisciata" tramite MCMC con parametro di lisciamento ( $\alpha$ ) pari a 0.99, e con  $\sigma$  posto uguale a 0.1. Dopo aver provato con diversi valori per  $\alpha$  abbiamo scelto un valore di  $\alpha$  pari a 0.99 perché era quello che ci permetteva un maggior lisciamento. La scelta del valore di  $\sigma$  (0.1), è stata fatta per garantire un compromesso tra un certo grado di lisciamento (con valori di  $\sigma > .1$  il lisciamento di MCMC dava troppa variabilità e non era in grado di "cogliere" l'andamento di fondo della serie) e il fatto di non "perdere" l'informazione di fondo data dai dati (con  $\sigma < .05$  la serie dei parametri di lisciamento simulati  $\theta_1^*, \theta_2^*, \dots, \theta_{136}^*$  sottostimava regolarmente la serie dei dati originari  $y_1, y_2, \dots, y_{136}$ ). Per implementare l'algoritmo come visto nel paragrafo 5.3.1, abbiamo poi posto  $n=136$  (lunghezza del campione),  $M-m=(2200-200)$  simulazioni in corrispondenza dei  $\theta_1^*, \theta_2^*, \dots, \theta_t^*, \dots, \theta_{136}^*$  (togliendo un burn in pari a 200) ed  $\epsilon=2$ . Inoltre abbiamo specificato, per poter eseguire l'algoritmo  $f(\theta_1^*) \sim N(\log(24), \sigma)$  e facendo partire la catena markoviana n-dimensionale da uno (abbiamo posto  $\log(24)$  perché la media dei gol segnati è pari appunto a 24).

MCMC con alpha=.99 e sigma=.1

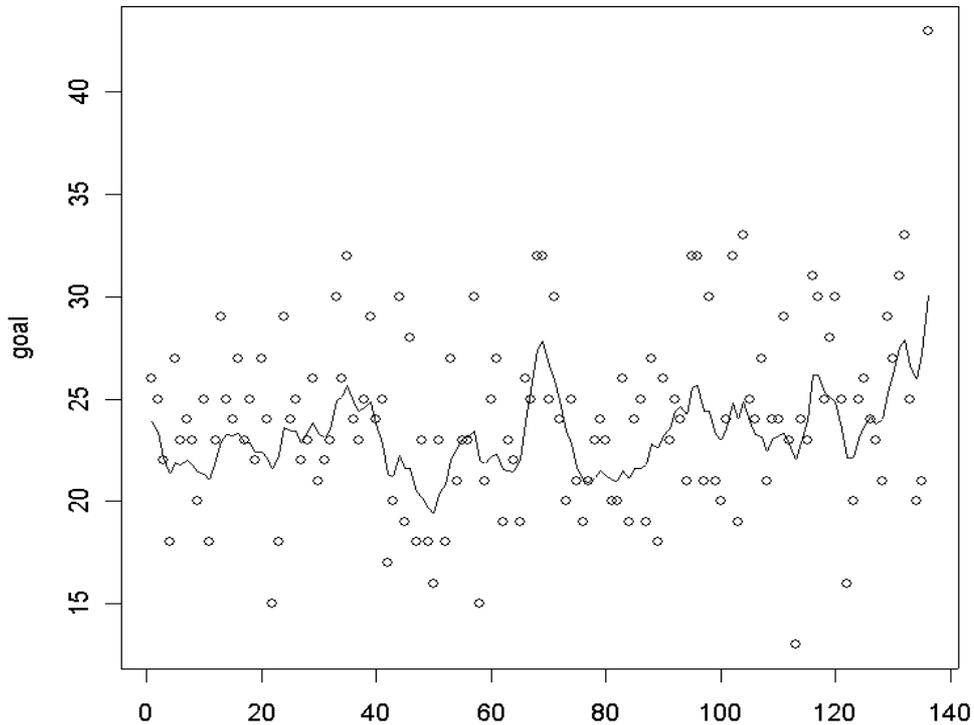


Figura 5.8

Si vede chiaramente in figura 5.8 come il liscio sembri abbastanza buono e come la serie storica dei parametri  $\theta_1^*, \theta_2^*, \dots, \theta_{136}^*$  (curva) approssimi bene i valori campionari  $y_1, y_2, \dots, y_{136}$  (pallini).

Nella figura 5.9, invece, riportiamo le stime dei parametri di liscio ottenute con IIS ponendo sempre  $\alpha=.99$  e  $\sigma=.1$  (anche in questo caso è risultata la combinazione dei 2 parametri migliore). Per realizzare l'algoritmo e per dare confrontabilità diretta ai risultati abbiamo poi tenuto  $M=2000$  e  $n=136$  come quanto fatto in figura 5.8.

### IIS con sigma=.1 e alpha=0.99

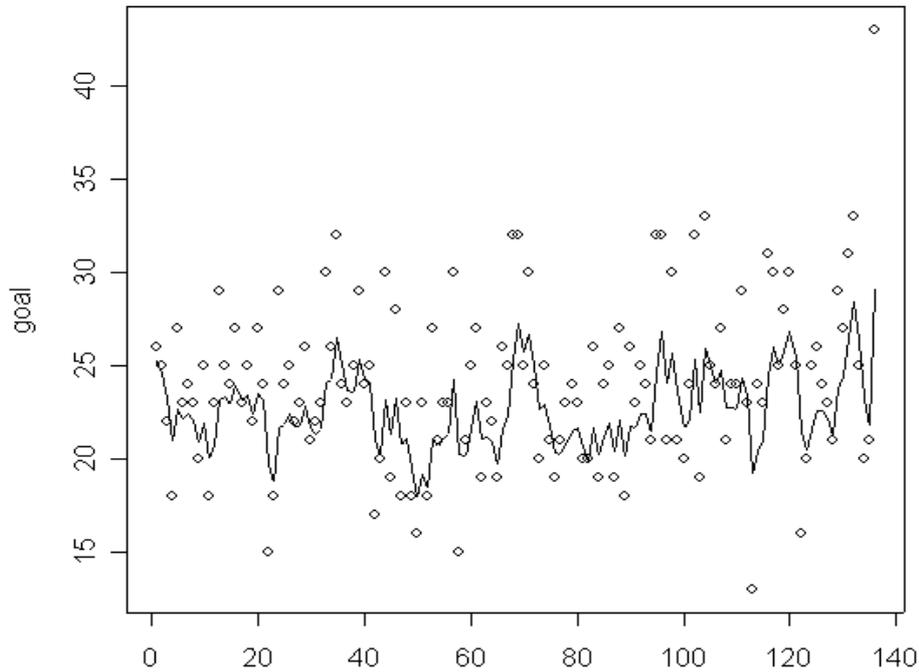


Figura 5.9

Si vede bene anche in figura 5.9 (così come avveniva, del resto, per la figura 5.8) come il liscio sembra buono (anche se qui, sembra esservi più variabilità): infatti, ad un picco campionario ne corrisponde uno nel parametro di liscio; ad un valore campionario basso, corrisponde un valore  $\theta_t^*$  basso. MCMC e IIS sembrano portare alle stesse conclusioni e a stime comparabili nella serie dei parametri di liscio  $\theta_1^*, \theta_2^*, \dots, \theta_{136}^*$ ; anche se, a parità di precisione, forse IIS dovrebbe portare a stime più efficienti per quanto visto nel paragrafo 5.3.4. Tornando al nostro problema di partenza (il valutare se a fine stagione si segni di più) si vede in ambo le figure come, alle giornate 34, 68, 102 e 136 (cioè nelle ultime giornate di campionato), ci siano dei picchi che stanno ad indicare il fatto che nelle ultime giornate di campionato si tenda a segnare di più.



## CAPITOLO 6

### CONCLUSIONI

In ambito Bayesiano, come del resto già ricordato all'inizio di quest'elaborato, è difficile simulare direttamente dalla distribuzione a posteriori vista la presenza della costante di normalizzazione al denominatore della formula (non esistono, infatti, distribuzioni a posteriori già tabulate nei principali pacchetti statistici) :

$$f(\theta | x) = (f(\theta) * f(x | \theta)) / \int (f(\theta) * f(x | \theta)) d\theta ;$$

Per ottenere un campione  $\theta_1, \theta_2, \dots, \theta_m$  della distribuzione a posteriori  $f(\theta | x)$  bisogna quindi ricorrere a metodi che ci permettano di simulare con semplicità sfruttando l'informazione campionaria iniziale  $f(x | \theta)$  e la distribuzione a priori  $f(\theta)$ . In quest'elaborato, abbiamo cercato di simulare dalla "a posteriori" utilizzando metodi come Importance Sampling (IS), Markov Chain of Montecarlo con passeggiata casuale (MCMC) e Iterative Importance Sampling (IIS). Abbiamo scelto questi tre metodi perché oltre ad essere tra i più usati, sono implementabili con relativa facilità. L'obiettivo di fondo di questa tesi era proprio vedere come si comportassero questi tre algoritmi di simulazione in ambito bayesiano cercando di valutare, di situazione in situazione, quale funzionasse meglio.

Nel capitolo 3, si è simulato prima, utilizzando il modello normale (a priori)-normale (a posteriori) con media ignota e varianza nota; poi, attraverso il

modello normale–normale con entrambi i parametri a posteriori non noti. In tutti e due i casi, conoscendo la vere media e varianza a posteriori, abbiamo potuto confrontare quale algoritmo di volta in volta fornisse stime più precise cambiando le caratteristiche di base proprie di ciascun algoritmo (distribuzione a priori, passo di campionamento, numerosità campionarie, ecc...). Non abbiamo individuato univocamente l'algoritmo migliore, ma abbiamo individuato il migliore (o i migliori) situazione per situazione.

Stessa cosa abbiamo fatto nel capitolo 4 dove, in un'approccio bayesiano a valori estremi, abbiamo confrontato le stime ottenute con i tre metodi di simulazione utilizzando come informazione campionaria  $f(x | \theta)$ , non più un campione normale, ma un campione della distribuzione GEV (Generalized Extreme Value). In quest'ambito i tre metodi potrebbero essere utilizzati indifferente l'uno dall'altro dato che non risultano esserci grosse differenze nelle stime.

Nel capitolo 5, infine, si è vista un'applicazione dei metodi (solo MCMC e IIS) a dati tipo serie storiche: prima abbiamo effettuato un confronto con dati simulati (IIS sembra funzionar meglio di MCMC), poi abbiamo fornito un'applicazione con dati “veri” (verificando la presenza di una stagionalità nei goal segnati nella serie A italiana nella stagioni 2001-2002, 2002-2003, 2003-2004 e 2004-2005).

Molto interessante, sarebbe stato vedere come, altri metodi di simulazione si sarebbero “comportati” nelle situazioni sopra descritte: se avrebbero dato stime più precise, più efficienti e così via.... Tutto ciò o un'analisi di quanto qui visto più dettagliata, potrebbe essere l'idea di partenza per un nuovo elaborato sulla simulazione in ambito bayesiano.

## APPENDICE

In quest'Appendice, riportiamo alcune (per motivi di spazio non abbiamo potuto riportarle tutte) delle funzioni utilizzate in quest'elaborato:

Importance Sampling nel caso normale-normale con media a post ignota e varianza a post nota

```
dati<-rnorm(10,0,1)
normal<-function(x,a,b,c,d,n){
tau<-NULL
ww<-NULL
w<-NULL
sam2<-rgamma(n,c,d)
sam1<-rnorm(n,a,sqrt(1/(sam2*b)))
for(j in 1:n){
  w[j]=prod(dnorm(x,sam1[j],sqrt(1/sam2[j])))}
ww<-w/sum(w)
tau<-sample(sam2,n,replace=T,prob=ww)
nn<-length(x)
astar<- (nn*mean(x) + b*a)/(nn+b)
bstar<- b+ nn
cstar<- c + sum(((x-mean(x))^2)/2) + ((nn*b)/(2*nn+2*b))*((mean(x)-a)^2)
dstar<-d + nn/2
tau<-sort(tau)
qqplot(tau,qgamma(((1:n)/(n+1)),cstar,dstar))
abline(0,1)}
normal(dati,0,0.5,2,2,100)
```

MCMC nel caso normale-normale con media a post ignota e varianza a post nota

```
dens1 <- function(dati,a,b,mu){
  f <- sum(dnorm(dati,mu,1,log=T))+ dnorm(mu,a,b,log=T)
  f}
```

```

mcnorm <- function(dati,a,b,nsim,ep,mu0){
out <- NULL
mu<- mu0
for(i in 1:nsim){
  mus <- mu+runif(1,-ep,ep)
  al <- dens1(dati,a,b,mus)-dens1(dati,a,b,mu)
  u <- log(runif(1))
  if(u<al) {mu <- mus}
  out <- c(out,mu) }
out<-out[100:nsim]
plot(out)
hist(out,prob=T)
v<-1/(length(dati)/1+1/b^2)
m<-(-a/b^2+(length(dati)*mean(dati))/1)*v
cat(mean(out),var(out),m,v,"\n")
y<-seq(-1,1,by=.01)
lines(y,dnorm(y,m,sqrt(v)))
qqnorm(out)
abline(m,sqrt(v),col=2,lwd=2)}
mcnorm(x,0,1,1100,.5,3)

```

IIS nel caso normale-normale con media a post ignota e varianza a post nota

```

x<-rnorm(100,0,1)
normal<-function (x,a,b,n,N,sigma,eps)
{par(mfrow=c(2,2))
th=matrix(0,nr=n,nc=N+10)
ww=matrix(0,nr=n,nc=N+10)
me=NULL
mm=NULL
w=NULL
sam=rnorm(n,a,b)
for(j in 1:n){
  w[j]=prod(dnorm(x,sam[j],sigma))}
ww[,1]=w/sum(w)
th[,1]=sample(sam,n,replace=T,prob=w)
su=sum(sam*ww[,1])
me[1]=su
mm[1]=su
for(i in 2:(N+10)){
sam=th[,i-1]+runif(n,-eps,eps)

```

```

for(j in 1:n){
  w[j]=dnorm(sam[j],a,b)*prod(dnorm(x,sam[j],sigma))}
ww[,i]=w/sum(w)
th[,i]=sample(sam,n,replace=T,prob=w)
su=su+sum(sam*ww[,i])
me[i]=su/i
mm[i]=sum(sam*ww[,i])
th=th[-(1:10)]
th1=as.vector(th)
ww=ww[-(1:10)]
hist(th1,prob=T)
v=1/(length(x)/sigma^2+1/b^2)
m=(a/b^2+(mean(x)*length(x))/sigma^2)*v
cat(mean(th1),var(th1),m,v,"\n")
y=seq(-6,6,by=.01)
lines(y,dnorm(y,m,sqrt(v)))
qqnorm(th1)
abline(m,sqrt(v),col=2,lwd=2)
#hist(ww[,N])
#plot(sam,ww[,N])
plot(me)
abline(h=m,col=2)
plot(mm)
abline(h=m,col=2)}
normal(x,0,1,1000,1000,1,.5)

```

## IS per valori estremi

```

gev.sim<- function(n,p){
  p[1] + (p[2] * (( - log(runif(n)))^( - p[3]) - 1))/p[3]
}
gev.lik <- function(a,xdat) {
### log veros. negata per gev con parametri in a e dati in xdat
  mu <- a[1]
  sc <- a[2]
  xi <- a[3]
  n <- length(xdat)
  y <- (xdat - mu)/sc
  y <- 1 + xi * y
  if (any(y <= 0) || any(sc <= 0))
    return(Inf)

```

```

      n*(log(sc)) + sum(y^(-1/xi)) + sum(log(y) * (1/xi + 1))}
dati<-gev.sim(10,c(0,1,0.2))
normal<-function(a,b,c,va,vb,vc,n){
par(mfrow=c(3,2))
sam1<-NULL
sam2<-NULL
sam3<-NULL
mu<-NULL
sigma<-NULL
csi<-NULL
phi<-NULL
ww<-NULL
w<-NULL
d1<-NULL
d2<-NULL
d3<-NULL
sam1<-rnorm(n,a,(va))
sam2<-exp(rnorm(n,b,(vb)))
sam3<-rnorm(n,c,(vc))
for(j in 1:n){
  w[j]=gev.lik(c(sam1[j],sam2[j],sam3[j]),dati)}
w<-exp(-w)
ww<-w/sum(w)
mu<-sample(sam1,n,replace=T,prob=ww)
sigma<-sample(sam2,n,replace=T,prob=ww)
csi<-sample(sam3,n,replace=T,prob=ww)
hist(mu)
d1<-density(mu)
plot(d1)
hist(sigma)
d2<-density(sigma)
plot(d2)
hist(csi)
d3<-density(csi)
plot(d3)}
normal(0,0,0,.5,.5,.5,1000)

```

## MCMC per valori estremi

```

gev.sim<- function(n,z){
  z[1] + (z[2] * (( - log(runif(n)))^( - z[3]) - 1))/z[3]}

```

```

gev.lik <- function(a,b,xdat) {
ndata<-length(xdat)
mu <- a[1]
sc <- a[2]
xi <- a[3]
y <- (xdat - mu)/sc
y <- 1 + xi * y
if (any(y <= 0) || any(sc <= 0))
return(-Inf)
nlv<- ndata*log(sc) + sum(y^(-1/xi)) + sum(log(y) * (1/xi + 1))
log.prior<- log(dnorm(a[1],b[1],b[2]))+log(dlnorm(a[2],b[3],b[4]))+log(dnorm(a[3],b[5],b[6]))-nlv+log.prior}
mcnorm <- function(med,w,nsim,p0,b,dati){
mu<-NULL
sigma<-NULL
csi<-NULL
out <- NULL
p <- p0
for(i in 1:nsim){
for(j in 1:3){
ps <- p
ps[j] <- p[j]+rnorm(1,med[j],w[j])
al <- gev.lik(ps,b,dati)-gev.lik(p,b,dati)
if(is.na(al))al<- -Inf
u <- log(runif(1))
if(u<al) p[j] <- ps[j]}
out <- rbind(out,p)}
mu<-out[,1]
sigma<-out[,2]
csi<-out[,3]
plot(mu)
plot(sigma)
plot(csi)
cbind(mu,sigma,csi)
}
par(mfrow=c(3,1))
dati <- gev.sim(100,c(0,1,.2))
res2<-mcnorm(c(0,0,0),c(.1,.1,.1),1000,c(1,2,1),c(0,.5,0,.5,0,.5),dati)

```

## IIS per valori estremi

```

gev.sim<- function(n,p){

```

```

      p[1] + (p[2] * (( - log(runif(n)))^( - p[3] - 1))/p[3])
gev.lik1 <- function(a,xdat) {
  mu <- a[1]
  sc <- a[2]
  xi <- a[3]
  n<-length(xdat)
  y <- (xdat - mu)/sc
  y <- 1 + xi * y
  if (any(y <= 0) || any(sc <= 0))
    return(Inf)
  n*(log(sc)) + sum(y^(-1/xi)) + sum(log(y) * (1/xi + 1))}
dati<-gev.sim(100,c(0,1,.2))
normaliis<-function(a,b,c,va,vb,vc,n,N,eps,dati){
par(mfrow=c(3,2))
ww=matrix(0,nr=n,nc=N)
sigma<-matrix(0,nr=n,nc=N)
mu<-matrix(0,nr=n,nc=N)
csi<-matrix(0,nr=n,nc=N)
sam1<-NULL
sam2<-NULL
sam3<-NULL
w=NULL
me1=NULL
mm1=NULL
me2=NULL
mm2=NULL
me3=NULL
mm3=NULL
su1=NULL
su2=NULL
su3=NULL
sam1=rnorm(n,a,sqrt(va))
sam2=exp(rnorm(n,b,sqrt(vb)))
sam3=rnorm(n,c,sqrt(vc))
for(j in 1:n){
  w[j]=gev.lik1(c(sam1[j],sam2[j],sam3[j]),dati)}
w<-exp(-w)
ww[,1]=w/sum(w)
indici<-sample(1:n,n,replace=T,prob=ww[,1])
mu[,1]<-sam1[indici]
sigma[,1]<-sam2[indici]

```

```

csi[,1]<-sam3[indici]
su1=sum(sam1*ww[,1])
me1[1]=su1
mm1[1]=su1
su2=sum(sam2*ww[,1])
me2[1]=su2
mm2[1]=su2
su3=sum(sam3*ww[,1])
me3[1]=su3
mm3[1]=su3
for(i in 2:N){
w<-NULL
indici<-NULL
sam1<-NULL
sam2<-NULL
sam3<-NULL
sam1<-mu[,i-1]+runif(n,-eps,eps)
sam2<-sigma[,i-1]+runif(n,-eps,eps)
sam3<-csi[,i-1]+runif(n,-eps,eps)
for(j in 1:n){
  w[j]= gev.lik1(c(sam1[j],sam2[j],sam3[j]),dati) }
w<-exp(-w)
ww[,i]=w/sum(w)
indici<-sample(1:n,n,replace=T,prob=ww[,i])
mu[,i]<-sam1[indici]
sigma[,i]<-sam2[indici]
csi[,i]<-sam3[indici]
su1=su1+sum(sam1*ww[,i])
me1[i]=su1/i
mm1[i]=sum(sam1*ww[,i])
su2=su2+sum(sam2*ww[,i])
me2[i]=su2/i
mm2[i]=sum(sam2*ww[,i])
su3=su3+sum(sam3*ww[,i])
me3[i]=su3/i
mm3[i]=sum(sam3*ww[,i])}
hist(mu[,N])
d1<-density(mu)
plot(d1)
hist(sigma[,N])
d2<-density(sigma)

```

```

plot(d2)
hist(csi[,N])
d3<-density(csi)
plot(d3)
list(mean(dati))}
normaliis(0,0,0,.5,.5,.5,1000,100,.01,dati)

```

## MCMC per filtraggio

```

autoregressive<-function(alpha,sigma,n){
theta<-NULL
yt<-NULL
theta[1]<-rnorm(1,0,sigma)
for(i in 2:(n+1)){
theta[i]<-(alpha*theta[i-1])+rnorm(1,0,sigma)
yt[i-1]<-rpois(1,exp(theta[i]))}
list(yt,theta[1:n])}
passage<-autoregressive(0.95,1,200)
thetateorico<-passage[[2]]
dati<-passage[[1]]
par(mfrow=c(4,2))
mcnorm <- function(dati,alpha,n,sigma,nsim,ep){
theta.init<-matrix(1,nr=1,nc=nsim)
theta<-matrix(0,nr=nsim,nc=n)
theta[,1]<-theta.init
thetas<-NULL
for(j in 2:n){
  for(i in 1:nsim){
    thetas<-theta[i,j-1]+runif(1,-ep,ep)
    if(i==1){
      al<- (dnorm(thetas,0,sigma,log=T) + dpois(dati[i],exp(thetas),log=T) + dnorm(theta[i+1,j-1],(alpha*thetas),sigma,log=T))- (dnorm(theta[i,j-1],0,sigma,log=T) + dpois(dati[i],exp(theta[i,j-1]),log=T) + dnorm(theta[i+1,j-1],(alpha*theta[i,j-1]),sigma,log=T))}
    if(i>1 & i<nsim){
      al <- (dnorm(thetas,(alpha*theta[i-1,j]),sigma,log=T) + dpois(dati[i],exp(thetas),log=T) + dnorm(theta[i+1,j-1],(alpha*thetas),sigma,log=T))- (dnorm(theta[i,j-1],(alpha*theta[i-1,j]),sigma,log=T) + dpois(dati[i],exp(theta[i,j-1]),log=T) + dnorm(theta[i+1,j-1],(alpha*theta[i,j-1]),sigma,log=T))}
    if(i==nsim){
      al <- (dnorm(thetas,alpha*theta[i-1,j],sigma,log=T) + dpois(dati[i],exp(thetas),log=T))- (dnorm(theta[i,j-1],alpha*theta[i-1,j],sigma,log=T) + dpois(dati[i],exp(theta[i,j-1]),log=T))}
    u <- log(runif(1))

```

```

if(u<al)
  theta[i,j] <- thetas
else
  theta[i,j] <- theta[i,j-1] }}
plot(theta[1,200:n],main="theta1")
plot(theta[50,200:n],main="theta50")
plot(theta[125,200:n],main="theta125")
plot(theta[200,200:n],main="theta200")
hist(theta[1,200:n],xlab="theta1",main="")
hist(theta[50,200:n],xlab="theta50",main="")
hist(theta[125,200:n],xlab="theta125",main="")
hist(theta[200,200:n],xlab="theta200",main="")
mcnorm(dati,.95,2200,1,200,2)

```

## IIS per filtraggio

```

autoregressive<-function(alpha,sigma,n){
theta<-NULL
yt<-NULL
theta[1]<-rnorm(1,0,sigma)
for(i in 2:(n+1)){
theta[i]<-(alpha*theta[i-1])+rnorm(1,0,sigma)
yt[i-1]<-rpois(1,exp(theta[i]))}
list(yt,theta[1:n])}
passage<-autoregressive(0.95,1,200)
thetateorico<-passage[[2]]
dati<-passage[[1]]
par(mfrow=c(4,2))
iisnorm <- function(dati,alpha,n,sigma,nsim){
theta.trans<-NULL
theta<-matrix(0,nr=nsim,nc=n)
ww<-matrix(0,nr=nsim,nc=n)
theta.trans<-rnorm(n,0,sigma)
thetas<-NULL
w<-NULL
for(j in 1:n){
w[j]=dpois(dati[1],exp(theta.trans[j]))}
ww[1,]=w/sum(w)
theta[1,]=sample(theta.trans,n,replace=T,prob=ww[1,])
for(i in 2:nsim){
w<-NULL

```

```

theta.trans<-NULL
theta.trans<-(alpha*theta[i-1,]) + rnorm(n,0,sigma)
for(z in 1:n){
w[z]=dpois(dati[i],exp(theta.trans[z]))}
ww[i,]=w/sum(w)
theta[i,]=sample(theta.trans,n,replace=T,prob=ww[i,])}
plot(theta[1,],main="theta1")
plot(theta[50,],main="theta50")
plot(theta[125,],main="theta125")
plot(theta[200,],main="theta200")
hist(theta[1,],main="theta1",xlab="")
hist(theta[50,],main="theta50",xlab="")
hist(theta[125,],main="theta125",xlab="")
hist(theta[200,],main="theta200",xlab="")}
iisnorm(dati,.95,2000,1,200)

```

## Bibliografia

ANDRIEU C., DOUCET A. e C.P. ROBERT C.P., *Computational Advances for and from Bayesian Analysis*, *Statistical Science* 2004, vol. 19, No. 1, 118–127;

AZZALINI A., *Inferenza Statistica: una Presentazione basata sul Concetto di Verosimiglianza*, Springer, 2001;

BERNARDO J. M. e SMITH A. F. M., *Bayesian Theory*, Wiley, 2004;

CHRISTIAN P. R., *The Bayesian Choice*, Springer, 2001;

COLES S., *Statistical Inference (Bayesian Inference)*, Dispensa utilizzata nel corso di *Statistica Computazionale II*, 10 giugno 1999;

COLES S., *An introduction to Statistical Modelling of Extreme Values*, Springer, 2001;

COLES S., *Statistica computazionale I*, Appunti del corso, Anno accademico 2003/2004 ;

COLES S., *Statistica computazionale II, Appunti del corso, Anno accademico 2003/2004* ;

DE FREITAS N. e DOUCET A., *Sequential Monte Carlo Methods in practice*, Springer, 2001 ;

DIXON M. J. e COLES S., *Modelling Association Football Scores and Inefficiencies in the Football Betting Market*, *Appl. Statist.* (1997) 46, No. 2, pp 265-280;

HARVEY A. C., *Forecasting structural time series models and the Kalman filter*, Cambridge, Cambridge University 1990;

LISEO B., *Statistica bayesiana, Dispensa utilizzata nel corso di Statistica (Corso progredito) Anno Accademico 2003/2004*;

MING -HUI CHEN, QI-MAN SHAO e IBRAHIM J., *Monte Carlo Methods in Bayesian Computation*, Springer 2000;

PACE L. e SALVAN A., *Introduzione alla Statistica II: Inferenza, Verosimiglianza, Modelli*, CEDAM, 2001;

ROBERT C. e CASELLA G., *Monte Carlo Statistical Methods*, Springer, 2004;

SMITH A. F. M. e GELFAND A. E., *Bayesian Statistics Without Tears: A Sampling – Resampling Perspective*, *The American Statistician*, May 1992, Vol. 46, No.2;

TANCREDI, *Importance Sampling con R: due esempi, appunti lezione del dottorato*, 30 giugno 2004;

## Ringraziamenti

In primis volevo ringraziare la mia famiglia per avermi sostenuto in questa bellissima esperienza all'università sia economicamente che moralmente. Non dimenticherò mai tutti gli amici qui conosciuti: Irene, Alessandro, Maurizio, Antonio (che peraltro già conoscevo), Stefania, Corrado, Valentina, Marco, Elisa, Claudio e tutte quelle persone che sicuramente sto dimenticando: ho condiviso con loro cinque anni bellissimi.

Un ringraziamento particolare va al professor Stuart Coles per l'estrema pazienza e competenza dimostratami in questi mesi e a tutti quei docenti che, con la loro disponibilità, mi hanno permesso di raggiungere quest'importante traguardo.

Non dimentico, infine, neanche le segretarie della presidenza (Fiorenza, Delfina, Paola), per il sorriso e la simpatia con cui mi hanno sempre accolto; e neanche tutti gli amici e persone del mio paese per avermi accompagnato e, in certi momenti, motivato.

Sono contento di aver portato a termine questa stupenda esperienza in questa facoltà: speriamo di poter raccogliere nella vita quanto seminato in questi, non mi stancherò mai di ripeterlo, bellissimi anni.