Università degli Studi di Padova
Dipartimento di Scienze Statistiche
Corso di Laurea Magistrale in
Scienze Statistiche

Stima bayesiana di un modello per la curva di fecondità basato sulla distribuzione normale asimmetrica

Relatore Prof. Bruno Scarpa
Dipartimento di Scienze Statistiche

Laureando: Edoardo Marchi
Matricola N 1062360

Anno Accademico 2014/2015

Indice

1 Introduzione **4**

2 Il tasso di fecondità **6**
2.1 Reperimento dei dati 6
2.2 I nostri dati 7

3 Modelli per i tassi di fecondità per età **12**

4 Strumenti utilizzati **16**
4.1 La distribuzione normale asimmetrica 16
4.1.1 Proprietà e momenti 17
4.1.2 Rappresentazioni alternative della distribuzione normale asimmetrica 18
4.1.3 La distribuzione normale asimmetrica unificata 20
4.2 Metodi MCMC e Gibbs Sampler 21
4.2.1 Catene markoviane 21
4.2.2 Metropolis-Hastings 24
4.2.3 Gibbs Sampler 24
4.3 Julia 25
4.3.1 Julia rispetto a R 25

5 Il modello di base **32**
5.1 Le osservazioni 32
5.2 Variabili latenti 32
5.3 Distribuzione di α 33
5.3.1 Sfruttare l'informazione per l'elicitazione a priori 34
5.4 La variabile latente η 35
5.5 Distribuzione di ξ e ω 35
5.6 Distribuzione di v 36
5.7 I passi dell'algoritmo 36

6 Il nuovo modello proposto **38**
6.1 Le osservazioni 39
6.2 Il parametro R 39
6.2.1 Elicitazione a priori 40
6.3 Gli altri parametri 40
6.4 L'algoritmo finale 40

7 I risultati ottenuti **44**
7.1 Scelta dei parametri iniziali 44
7.2 La stima dei parametri 45

8 Conclusioni **52**

1 Introduzione

In questo lavoro è stato costruito un modello bayesiano per la stima della curva di fecondità, basandosi sul modello definito in *Canale and Scarpa [2013]*. Questi, partendo dall'assunzione che la curva di fecondità possa essere scritta come

$$g(y; R, \theta_1, \ldots, \theta_{r-1}) = R \cdot h(y; \theta_1, \ldots, \theta_{r-1}) \tag{1.1}$$

con $h(y; \theta_1, \ldots, \theta_{r-1})$ funzione di densità, R tasso di fecondità e y età della madre alla nascita del figlio, hanno stimato il modello per la sola $h(y; \theta_1, \ldots, \theta_{r-1})$, ottenuta dividendo la curva di fecondità per una stima esterna del parametro R. Avendo, generalmente, a disposizione dati che ricoprono un intervallo di tempo di diversi anni consecutivi e avendo essi un'evoluzione per lo più graduale e costante, si può pensare di sfruttare l'informazione ottenuta dalle stime dei parametri per l'anno $t-1$ per influenzare positivamente le stime dei parametri dell'anno t, da cui l'utilità dell'approccio bayesiano seguito.

Lo scopo di questo lavoro è stato, quindi, quello di integrare la stima del parametro R nel modello di Canale and Scarpa [2013], ritrovandosi quindi a dover gestire una funzione che non è più una densità propria. Al fine di dare una dimostrazione pratica dell'efficacia del modello, sono stati utilizzati i dati del Giappone, presi dalla banca dati *www.humanfertility.org*, e, in particolare, i dati relativi all'età della madre alla nascita del figlio e alla numerosità per età delle donne dai 12 ai 55 anni, per l'intervallo temporale dall'anno 1980 al 2012.

2 Il tasso di fecondità

Il tasso di fecondità totale, detto anche TFT, è un indicatore statistico, calcolabile su base annuale, che rappresenta il numero di nascite medio che avrebbe una donna se, ipoteticamente, vivesse il suo intero periodo fertile in un unico anno, rispettando i tassi di fecondità specifici per età dell'anno.

Questo valore è quindi definito come

$$TFT = \sum_{x=\alpha}^{\beta} f_x$$

dove α e β sono gli anni o le classi di età estreme del periodo di fertilità di una donna, generalmente fissati a 15-49 oppure a 12-55, e f_x è il tasso di fecondità specifico per l'età, o la classe di età, x.

Il TFT è quindi un indicatore astratto, inosservabile nella realtà, ma che ci permette di avere una misura della fecondità di un paese confrontabile con altri paesi, anche a fronte di una diversa distribuzione di donne per età.

I tassi di fecondità per età, detti TFS, misurano il numero di nascite da donne di una specifica età, o classe di età, per 1000 donne appartenenti a quella classe di età.

Possiamo quindi calcolare questo valore come

$$TFS_x = \frac{B_x}{D_x} 1000 \tag{2.1}$$

dove B_x è il numero di nascite da donne di età x e D_x è il numero di donne di età x.

2.1 Reperimento dei dati

I dati che ci interessano, cioè quelli relativi all'età della madre alla nascita di un figlio e quelli relativi all'età di tutte le donne all'interno del paese, possono essere reperiti da molteplici fonti. Quelle sicuramente più affidabili e complete sono date dai sistemi di registrazione civile che registrano ogni nascita avvenuta. Nei paesi meno sviluppati, che invece non possiedono un sistema di questo tipo, è possibile ricavare queste informazioni da indagini censuarie o campionarie effettuate periodicamente. Dati relativi a queste ultime due fonti, però, sono ovviamente meno affidabili in quanto nell'intervallo di tempo tra un'analisi e l'altra possono accadere innumerevoli cause che provocano dati errati o mancanti, come ad esempio la morte prematura di una donna o l'errata

compilazione da parte dei singoli individui dei questionari o ancora l'incapacità di ricordare eventi accaduti anche svariati anni prima. Il confronto tra i dati dei registri civili e quelli relativi a questo tipo di indagini mostra come generalmente i dati di questi ultimi siano meno affidabili dei primi.

I dati vengono poi resi pubblici, in forma anonima, dagli uffici statistici nazionali del paese di interesse e dalle grandi banche dati mondiali dalle quali è possibile avere accesso ai dati di un gran numero di paesi. Tra le più importanti citiamo *www.worldbank.org* e *www.humanfertility.org* che si impegnano a mantenere dati il più possibile affidabili e confrontabili nel tempo.

2.2 I nostri dati

I dati per questo lavoro sono stati presi dal sito *www.humanfertility.org* e sono relativi al Giappone. I criteri che hanno portato alla scelta di questo paese sono semplici: i dati sono di ottima qualità, al contrario di quelli di paesi meno sviluppati che magari non hanno un sistema censuario e le stime sono spesso incomplete o imprecise; inoltre i paesi europei, tra i quali sarebbe anche potuta ricadere la scelta, hanno tutti la caratteristica di un secondo hump, in almeno qualche anno, che può essere trattata con un modello basato sulla distribuzione normale asimmetrica flessibile, descritta in Mazzuco and Scarpa [2015], ma che non verrà trattata in questo lavoro.

I dati raccolti si riferiscono al periodo che va dal 1980 al 2012 e, per ognuno di questi anni, contengono la numerosità di nuovi nati, divisi per età della madre al momento del parto, e la numerosità delle donne, anche queste suddivise per età. Grazie a queste due tipologie di dati è possibile calcolare il tasso di fecondità, come in (2.1).

Prendiamo, come esempio, cinque anni: 1980, 1988, 1996, 2004, 2012, spaziati equamente lungo tutta la linea temporale dei dati. In Figura 1 sono riportati i tassi di fecondità specifici per età degli anni di esempio.

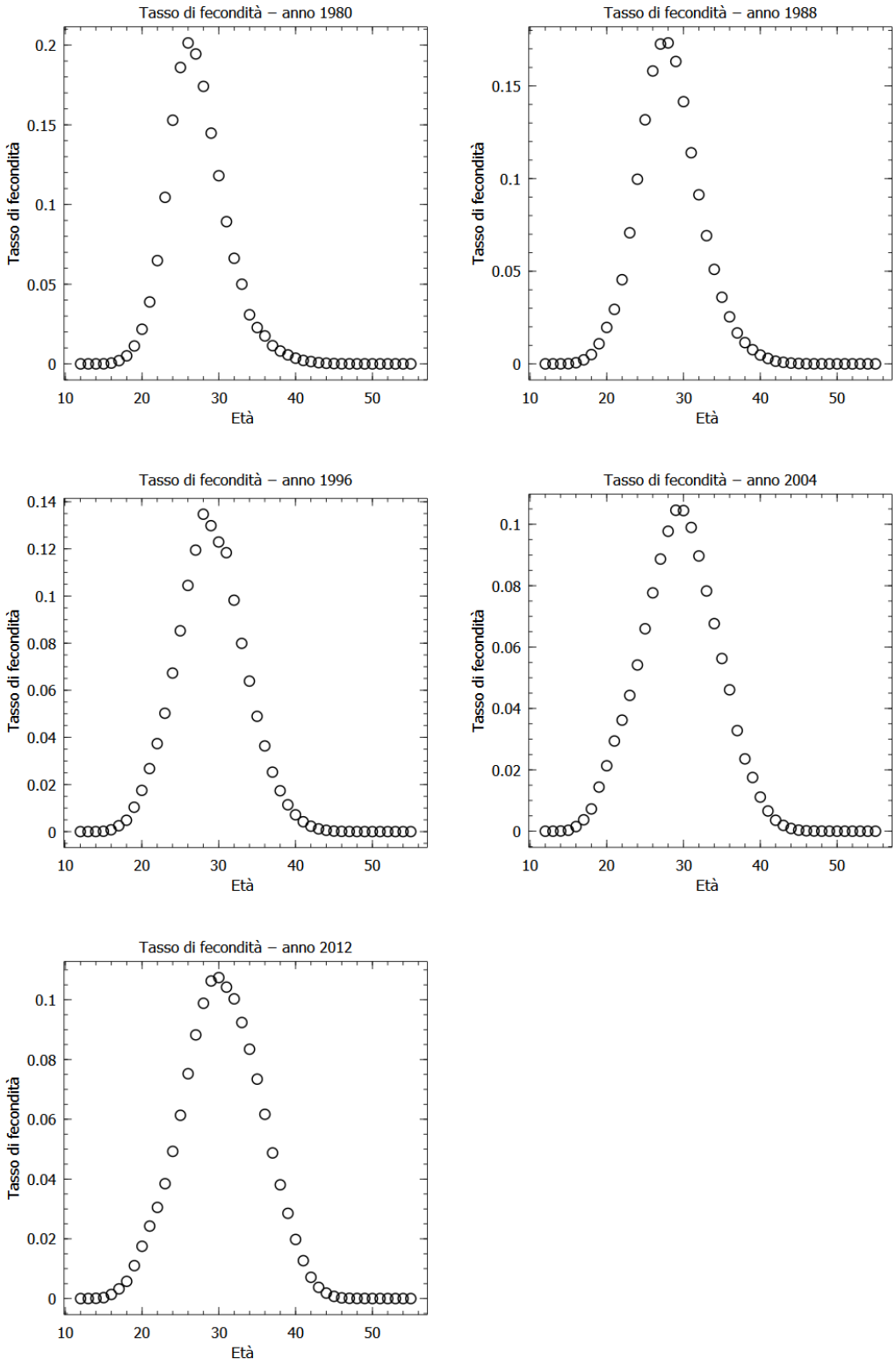

Figura 1: Tassi di fecondità per gli anni 1980, 1988, 1996, 2004 e 2012.

Notiamo subito come l'età media sia aumentata negli anni e la forma della curva risulti sempre più simmetrica. Questo si può anche vedere nella seguente

tabella che riporta alcuni dati del Giappone, calcolati tramite i dati a disposizione. Bisogna fare attenzione al fatto che l'età media delle donne è stata calcolata considerando solo le donne nella fascia di età tra i 12 e i 55 anni! Si può comunque notare, come abbiamo detto prima, che l'età media della madre al momento del parto sia aumentata e che il numero totale di parti e di donne sia diminuito notevolmente nel corso dell'ultimo trentennio.

Anno	**Età media al parto**	**Età media delle donne**	**Totale parti**	**Totale donne**
1980	27.62	33.17	1576888	37206146
1981	27.73	33.23	1529451	37420964
1982	27.83	33.28	1515389	37616801
1983	27.91	33.31	1508686	37821013
1984	27.98	33.31	1489779	38042370
1985	28.07	33.31	1431575	38270844
1986	28.13	33.32	1382944	38458773
1987	28.21	33.34	1346656	38580358
1988	28.30	33.38	1314005	38634102
1989	28.36	33.43	1246799	38635796
1990	28.41	33.50	1221584	38613480
1991	28.40	33.58	1223245	38558702
1992	28.42	33.66	1208987	38466396
1993	28.46	33.76	1188279	38343260
1994	28.52	33.88	1238327	38229300
1995	28.59	34.00	1187063	38143043
1996	28.69	34.11	1206553	38024789
1997	28.77	34.19	1191664	37818235
1998	28.86	34.25	1203146	37549634
1999	28.93	34.32	1177668	37249270
2000	29.06	34.43	1190547	36891522
2001	29.15	34.58	1170659	36635187
2002	29.29	34.76	1153852	36454323
2003	29.50	34.82	1123608	36042180
2004	29.69	34.77	1110719	35426411
2005	29.86	34.72	1062526	34821423
2006	30.04	34.69	1092673	34305465
2007	30.24	34.68	1089812	33865495
2008	30.37	34.72	1091155	33492202
2009	30.53	34.78	1070029	33173013
2010	30.69	34.86	1071299	32907871
2011	30.81	34.94	1050803	32655323
2012	30.99	35.04	1037225	32417016

Tabella 1: Dati sulla fecondità del Giappone: età media delle madri al parto, età media delle donne nel paese, numero totale di parti e numero totale di donne.

3 Modelli per i tassi di fecondità per età

La distribuzione dei tassi di fecondità per età è una funzione definita sull'intervallo di età in cui una donna è fertile. Generalmente questo intervallo è fissato a 15-44 oppure a 15-49 anni, ma a volte i dati comprendono intervalli anche più ampi come 12-55 anni. La curva presenta il massimo in un'età che va dai 20 ai 30 anni e, a seconda del paese e dell'anno, si è visto che può avere un «hump», generalmente in età adolescenziale. Inoltre, una caratteristica importante di questa distribuzione, comune alla maggior parte dei paesi, è la presenza di un'asimmetria a sinistra che rende la modellazione difficoltosa e per questo, oltre che per l'importanza della tematica, la ricerca di un modello adatto alla stima dei tassi di fecondità è stato oggetto di molti studi nel corso degli ultimi decenni. In Figura 1 abbiamo visto un esempio di quanto può essere asimmetrica la curva in questione, soprattutto nei primi anni.

Seguendo la definizione di Hoem et al. [1981] possiamo definire la funzione dei tassi di fecondità per età come

$$g(y; R, \theta_1, \ldots, \theta_{r-1}) = R \cdot h(y; \theta_1, \ldots, \theta_{r-1}) \tag{3.1}$$

dove $g(y; R, \theta_1, \ldots, \theta_{r-1})$ è il tasso di fecondità per l'età y, R è il tasso di fecondità totale e $h(y; \theta_1, \ldots, \theta_{r-1})$ è una funzione di densità di probabilità con i restanti $r-1$ parametri.

La funzione $h(y; \theta_1, \ldots, \theta_{r-1})$, in particolare è stata oggetto di studi e ne sono state date molte specificazioni. Gli stessi Hoem et al. [1981] hanno proposto la Hadwiger (Gaussiana inversa), la distribuzione gamma, beta, Coale-Trussel, Brass e Gompertz. Hanno inoltre definito altri due modelli basati sulle spline di regressione e su funzioni polinomiali e hanno visto come quello basato sulle spline di regressione è quello che si adatta meglio ai dati, anche se richiede la stima di ben 10 parametri. Tra i modelli più parsimoniosi in termini di parametri, quelli che hanno dato migliori risultati sono gamma, Coale-Trussel e, in alcuni casi, la funzione di Hadwiger. Quest'ultima funzione è stata definita dallo stesso Hadwiger [1940] come

$$g(y; a, b, c) = \frac{ab}{c}\left(\frac{c}{y}\right)^{3/2} exp\left\{-b^2\left(\frac{c}{y} + \frac{y}{c} - 2\right)\right\} \tag{3.2}$$

Il modello (3.2) è riconducibile alla forma descritta dal modello (3.1) ponendo $R = a\sqrt{\pi}$ e $g(\cdot)$ la funzione di densità di Hardwiger.

Un'estensione del modello (3.2) è data dal modello mistura proposto da

Chandola et al. [1999] che permette di cogliere anche il caso di una distribuzione con due mode. Tale modello è

$$g(y; a, m, b_1, b_2, c_1, c_2) = m\frac{ab_1}{c_1}\left(\frac{c_1}{y}\right)^{3/2} exp\left\{-b_1^2\left(\frac{c_1}{y} + \frac{y}{c_1} - 2\right)\right\} + (1-m)\frac{ab_2}{c_2}\left(\frac{c_2}{y}\right)^{3/2} exp\left\{-b_2^2\left(\frac{c_2}{y} + \frac{y}{c_2} - 2\right)\right\} \quad (3.3)$$

dove $0 \leqslant m \leqslant 1$ è il parametro mistura che determina la dimensione delle due popolazioni sottostanti. Il numero di parametri, però, è più del doppio del modello (3.2), anche se Chandola et al. [1999] hanno mostrato che è possibile dare un'interpretazione demografica ai parametri.

Un altro modello, sviluppato in questo caso in ambito semi-parametrico, è quello di Schmertmann [2003] basato su spline quadratiche

$$g(y; R, \alpha, \beta, \theta_0, \ldots, \theta_4, t_0, \ldots, t_4) = R \cdot I(\alpha \leqslant y \leqslant \beta) \sum_{k=0}^{4} \theta_k (y - t_k)_+^2 \quad (3.4)$$

dove $I(\cdot)$ è una funzione indicatrice, α e β sono i limiti inferiore e superiore di età, θ_k i parametri e t_k i nodi delle splines. Grazie a questa proposta è possibile modellare le più svariate strutture, ma essa ha lo svantaggio di richiedere addirittura 13 parametri e l'interpretazione della formula non è immediata.

Un'altra proposta è quella di Mazzuco and Scarpa [2015] che propongono un modello basato sulla distribuzione Normale asimmetrica definita come

$$h(y; \xi, \omega^2, \alpha) = \frac{2}{\omega}\phi\left(\frac{x-\xi}{\omega}\right)\Phi\left\{\alpha\left(\frac{x-\xi}{\omega}\right)\right\} \quad (3.5)$$

dove ϕ e Φ sono, rispettivamente, la funzione di densità e di ripartizione di una distribuzione Normale standard e il parametro α permette di gestire il grado di asimmetria della distribuzione. Questa distribuzione, di cui parleremo più approfonditamente nel seguito, è stata studiata da Azzalini [1985] e da altri. È interessante vedere che se $\alpha = 0$ la funzione si riduce a una funzione di densità Normale. Essendo questa distribuzione unimodale, Ma and Genton [2004] hanno proposto una loro versione che permettesse di modellare anche le distribuzioni bimodali e che hanno chiamato FGSN (Flexible Generalized

Skew Normal). Questa estensione è definita come

$$h(y, \xi, \omega^2, \alpha, \beta) = \frac{2}{\omega}\phi\left(\frac{y-\xi}{\omega}\right)\Phi\left\{\alpha\left(\frac{y-\xi}{\omega}\right) + \beta\left(\frac{y-\xi}{\omega}\right)^3\right\} \tag{3.6}$$

Ma and Genton [2004] hanno dimostrato che la funzione (3.6) può avere al massimo due mode, ma che all'aumentare del grado del polinomio argomento di $\Phi(\cdot)$, si può aumentare il numero delle possibili mode. E' stato visto, in ogni caso, che due mode siano generalmente sufficienti per quanto riguarda la distribuzione dei tassi di fecondità specifici.

Una precisazione del precedente modello arriva da Zenere [2015] che, vista la natura dei dati, utilizza una distribuzione

$$Multinom(y; \psi_1, \ldots, \psi_K) \tag{3.7}$$

con K classi di età, dove ciascuna probabilità è data da

$$\psi_k = \int_k^{k+t} h(y; \theta)dy \tag{3.8}$$

con $k = 1, \ldots, K$ e t il numero di anni della classe di età. Alternativamente si può usare in (3.8) la funzione (3.6), per trattare anche il caso bimodale.

In seguito Griggio [2014] ha rielaborato il problema in chiave Bayesiana e, sfruttando l'algoritmo Gibbs Sampler costruito da Canale and Scarpa [2013], ha stimato i parametri del modello utilizzando delle distribuzioni a priori informative basate sulle stime a posteriori dei parametri dell'anno precedente. Le distribuzioni per $h(\cdot)$ da lui usate sono, ancora una volta la (3.5) e la (3.7).

Federico Cauduro [2015] utilizza il modello definito da Griggio per costruire un modello bayesiano gerarchico per aggiornare la stima della varianza del parametro di posizione ξ e, infine, Peruzzo [2015] propone un modello per stimare la funzione $g(\cdot)$ attraverso la verosimiglianza, anche in presenza di valori mancanti.

Un problema di tutte queste ultime proposte, a parte l'ultima, è che studiano solo la funzione $h(\cdot)$ e non $g(\cdot)$ nella sua interezza. Questo fa sì che per stimare il parametro R si utilizzi una procedura in due passi, dove prima si stima R in modo da ottenere la funzione di densità $h(\cdot) = \frac{g(\cdot)}{R}$ e poi si stima quest'ultima con i modelli sopracitati. Visto che l'onerosità computazionale aggiuntiva per la stima dell'r-esimo parametro non è incisiva, non c'è motivo

di accettare l'inaccuratezza aggiuntiva portata dalla stima in due passi.

Obiettivo di questo lavoro è incorporare la stima di R all'interno del modello di Griggio [2014] e, utilizzando l'algoritmo Gibbs Sampler di Canale and Scarpa [2013], opportunamente modificato, stimare tutti gli r parametri in un unico passo, senza l'utilizzo di stime esterne per il parametro R.

4 Strumenti utilizzati

Di seguito vengono riportati i principali strumenti utilizzati in questo lavoro, con una breve trattazione per ciascuno di essi e le motivazioni alla base della loro scelta.

4.1 La distribuzione normale asimmetrica

Questa famiglia di distribuzioni, ideata ed elaborata da Azzalini [1985], si basa su un importante Lemma.

Lemma 1. *[Azzalini, 1985] Sia f_0 una funzione di densità di probabilità in $\mathbb{R}^d$, sia $G_0(\cdot)$ una funzione di una distribuzione continua nell'asse reale e sia $\omega(\cdot)$ una funzione a valori reali in $\mathbb{R}^d$, tale che*

$$f_0(-x) = f_0(x), \quad \omega(-x) = -\omega(x), \quad G_0(-y) = 1 - G_0(y)$$

per tutti i valori di $x \in \mathbb{R}^d$, $y \in \mathbb{R}$. Allora

$$f(x) = 2f_0(x)G_0\{\omega(x)\} \tag{4.1}$$

è una funzione di densità in $\mathbb{R}^d$.

Con questo Lemma è quindi possibile unire ad una funzione di densità simmetrica f_0 una funzione $G_0\{\omega(x)\}$ che rispetti le proprietà viste nel Lemma e ottenere ancora una funzione di densità. Scegliendo $f_0 = \phi$, $G_0 = \Phi$ e $\omega(x) = \alpha x$, con ϕ e Φ, funzioni di densità e di ripartizione di una normale standard e $\alpha \in \mathbb{R}$, Azzalini [1985] ha definito la funzione di densità della distribuzione normale asimmetrica, abbreviata SN (dall'inglese, Skew-Normal) che assume quindi la forma

$$\phi(x;\alpha) = 2\phi(x)\Phi(\alpha x) \tag{4.2}$$

con una notazione che riprende quella della distribuzione normale standard dove viene aggiunto il parametro di forma che regola l'asimmetria.

Nella figura (2) si può vedere come si distribuisce la SN al variare di α.

Analogamente alla distribuzione normale, possiamo introdurre i parametri di posizione, $\xi \in \mathbb{R}$, e di scala, $\omega \in \mathbb{R}^+$, nella distribuzione normale asimmetrica. Pertanto avremo che, data una variabile casuale $Z \sim \phi(x;\alpha)$,

$$Y = \xi + \omega Z \tag{4.3}$$

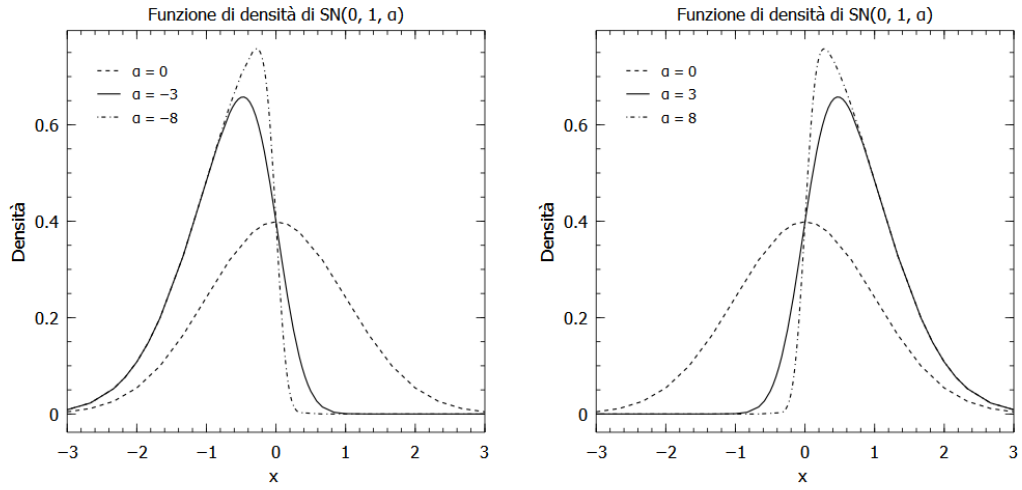

Figura 2: Grafici della funzione di densità di una SN con media nulla e varianza unitara al variare di α

sarà una variabile che si distribuisce come una distribuzione normale asimmetrica. Riconducendoci alla (4.2), possiamo scrivere che la funzione di densità di Y, come definita in (4.3), è

$$\phi\left(\frac{x-\xi}{\omega};\alpha\right) = \frac{2}{\omega}\phi\left(\frac{x-\xi}{\omega}\right)\Phi\left(\alpha\frac{x-\xi}{\omega}\right) \tag{4.4}$$

e che

$$Y \sim SN(\xi, \omega^2, \alpha) \tag{4.5}$$

dove ω^2 è usato per analogia con la notazione $N(\mu, \sigma^2)$.

4.1.1 Proprietà e momenti

Definiamo alcune proprietà definite in Azzalini and Capitanio [2014].

Proposizione 2. *Sia Z una variabile casuale $SN(0,1,\alpha)$ con funzione di densità $\phi(x;\alpha)$. Allora valgono le seguenti proprietà:*

1. $\phi(x;0) = \phi(x), \quad \forall x;$
2. $\phi(0;\alpha) = \phi(0), \quad \forall \alpha;$
3. $-Z \sim SN(0,1,-\alpha)$, e quindi $\phi(-x;\alpha) = \phi(x;-\alpha), \quad \forall x;$
4. $\lim_{\alpha\to\infty} \phi(x;\alpha) = 2\phi(x)I_{(0,\infty)}(x), \quad \forall x;$
5. $Z^2 \sim \chi^2_1, \quad \forall \alpha;$

6. se $Z^{'} \sim SN(0,1,\alpha^{'})$, con $\alpha^{'} < \alpha$, allora $Z^{'} <_{st} Z$.

Riportiamo, infine, i primi 4 momenti della normale asimmetrica che ci saranno utili in seguito per ricavare delle stime per i parametri delle distribuzioni a priori.

$$E(Y) = \xi + \omega\sqrt{2/\pi}\delta \tag{4.6}$$

$$var(Y) = \omega^2(1 - (2/\pi)\delta^2) \tag{4.7}$$

$$\gamma_1(Y) = \frac{4-\pi}{2}\left(\frac{\mu_Z^2}{\sigma_Z^2}\right)^2 \tag{4.8}$$

$$\gamma_2(Y) = 2(\pi-3)\left(\frac{\mu_Z^2}{\sigma_z^2}\right)^2 \tag{4.9}$$

4.1.2 Rappresentazioni alternative della distribuzione normale asimmetrica

Una caratteristica molto interessante della distribuzione normale asimmetrica è quella di poter essere rappresentata in modo stocastico in una varietà di modi. Tale caratteristica è molto utile per la generazione casuale di valori da questa distribuzione e verrà impiegata in seguito nella costruzione del Gibbs Sampler che permetterà di stimare il modello finale.

Sia $Z \sim SN(0,1,\alpha)$ e X_0 e U due variabili normali standard indipendenti fra loro. Z può essere rappresentata come

$$Z = \begin{cases} X_0 & se\, U < \alpha X_0 \\ -X_0 & altrimenti \end{cases} \tag{4.10}$$

o alternativamente come

$$Z = (X_0 | U < \alpha X_0) \tag{4.11}$$

Di queste due rappresentazioni, la (4.10) è sicuramente la migliore dal punto di vista della generazione casuale di numeri, in quanto possiamo generare n valori in esattamente n iterazioni, grazie al fatto che tutti i valori vengono sempre accettati. Nella (4.11), invece, i valori vengono accettati solo con probabilità $\mathbb{P}(U < \alpha X_0)$, comportando un numero di iterazioni maggiori o uguali a n. Il pregio della (4.11) è che è più adatta a considerazioni teoriche.

Possiamo anche rappresentare Z attraverso una variabile normale bivariata (X_0, X_1), con le marginali standardizzate, dove

$$X_1 = \frac{\alpha X_0 - U}{\sqrt{1+\alpha^2}}$$

tale che $cor(X_0, X_1) = \delta(\alpha)$, dove

$$\delta(\alpha) = \frac{\alpha}{\sqrt{1+\alpha^2}}.$$

In questo modo possiamo definire Z come

$$Z = \begin{cases} X_0 & se\ X_1 > 0 \\ -X_0 & altrimenti \end{cases} \tag{4.12}$$

o alternativamente come

$$Z = (X_0 | X_1 > 0) \tag{4.13}$$

Per definire l'ultima rappresentazione alternativa di Z, che è anche quella che ci servirà nel seguito di questo lavoro, dobbiamo introdurre la seguente affermazione:

Proposizione 3. *[Chiogna, 1998] Se* $Y_1 \sim SN(\xi, \omega^2, \alpha)$ *e* $Y_2 \sim N(\mu, \sigma^2)$ *sono variabili casuali indipendenti, allora*

$$Y_1 + Y_2 \sim SN(\xi + \mu, \omega^2 + \sigma^2, \tilde{\alpha}), \quad con\ \tilde{\alpha} = \frac{\alpha}{\sqrt{1 + (1+\alpha^2)\sigma^2/\omega^2}} \tag{4.14}$$

Inoltre

$$\lim_{\alpha \to \pm\infty} \tilde{\alpha} = \pm \frac{\omega}{\sigma} \tag{4.15}$$

Ora, utilizzando la (4.14) nel caso limite (4.15), ponendo $\omega = |\delta|$, dove $\delta \in (-1, 1)$, e $\sigma^2 = 1 - \delta^2$ e, infine, considerando U_0, U_1 due variabili $N(0,1)$ indipendenti, posso rappresentare Z nel modo seguente:

$$Z = \delta |U_0| + \sqrt{1-\delta^2} U_1 \sim SN(0, 1, \alpha) \tag{4.16}$$

dove

$$\alpha = \alpha(\delta) = \frac{\delta}{\sqrt{1-\delta^2}}$$

La (4.16) è facilmente estendibile al caso con parametro di posizione diverso da 0 e parametro di scala diverso da 1. Infatti, sia $Y \sim SN(\xi, \omega^2, \alpha)$, per la

(4.3), possiamo scrivere

$$Y = \xi + \omega Z = \xi + \omega \left(\delta |U_0| + \sqrt{1 - \delta^2} U_1 \right) \sim SN(\xi, \omega^2, \alpha) \tag{4.17}$$

4.1.3 La distribuzione normale asimmetrica unificata

Al fine di effettuare la stima a posteriori del parametro di forma del nostro modello è utile considerare anche un'estensione della distribuzione normale asimmetrica che prende il nome di *normale asimmetrica unificata*. Tale distribuzione è stata introdotta da Arellano-Valle and Azzalini [2006].

Proposizione 4. *[Arellano-Valle and Azzalini, 2006] Siano V_0 e V_1 due variabili indipendenti tali che*

$$V_{0\gamma} \sim LTN_m(-\gamma; 0, \Gamma), \quad V_1 \sim N_d(0, \Psi)$$

con Γ e Ψ matrici di correlazione e indicando con la notazione $LTN_m(c; \mu, \Sigma)$ una variabile normale m-variata troncata sotto c, e si consideri la seguente trasformazione

$$Y = \xi + \omega(B_0 V_{0\gamma} + B_1 V_1) \tag{4.18}$$

con $B_0 = \Delta\Gamma^{-1}$ e B_1 matrice d x d tale che

$$B_1 \Psi B_1^T = \bar{\Omega} - \Delta\Gamma^1\Delta^T$$

dove $\Gamma, \Delta, \bar{\Omega}$ sono gli elementi ottenuti dalla partizione della matrice di correlazione

$$\Omega^* = \begin{pmatrix} \Gamma & \Delta^T \\ \Delta & \bar{\Omega} \end{pmatrix}$$

allora la funzione di densità di Y, per come definita in (4.18), è pari a

$$f(y) = \phi_d(y - \xi; \omega\bar{\Omega}\omega) \frac{\Phi_m(\gamma + \Delta^T\bar{\Omega}^{-1}\omega^{-1}(y - \xi); \Gamma - \Delta^T\bar{\Omega}^{-1}\Delta)}{\Phi_m(\gamma; \Gamma)} \tag{4.19}$$

per $y \in \mathbb{R}^d$, con $\phi_d(\cdot)$ e $\Phi_d(\cdot)$, rispettivamente, funzione di densità e di ripartizione di una normale d-variata e con ω matrice diagonale d x d.

La distribuzione di Y è allora normale asimmetrica unificata (SUN) ed è definita dalla seguente notazione

$$Y \sim SUN_{m,d}(\xi, \gamma, \omega, \bar{\Omega}, \Delta, \Gamma) \tag{4.20}$$

Grazie a questa definizione possiamo introdurre un'estensione del Lemma (4.17) che risulterà fondamentale per la stima a poseriori dei parametri del modello.

Lemma 5. *[Arellano-Valle and Azzalini, 2006] Sia* $V_0 \sim LTN_q(-\gamma; 0, \Gamma), V_1 \sim N(0,1)$ *con* V_0 *indipendente da* V_1 *e dove la notazione* $LTN_d(\tau; \mu, \Sigma)$ *denota una distribuzione normale d-variata con media* μ *e matrice di varianza e covarianza* Σ *troncata sotto* τ*. Se*

$$Y = \xi + \omega(\Delta\Gamma^{-1}V_0 + \sqrt{1 - \Delta^T\Gamma^{-1}\Delta}V_1)$$

allora

$$Y \sim SUN_{1,q}(\xi, \gamma, \omega, 1, \Delta, \Gamma)$$

4.2 Metodi MCMC e Gibbs Sampler

Il Gibbs Sampler è l'algoritmo fondamentale che ci ha permesso di stimare i parametri del modello. Iniziamo dunque con il definire la base su cui si poggia questo algoritmo: le catene markoviane.

4.2.1 Catene markoviane

Proposizione 6. *Sia* $X_t, \quad t = 0, 1, 2, \ldots$ *una serie di variabili casuali. Essa si dice catena markoviana di primo ordine se, per* $t = 0, 1, \ldots$ *e per tutti gli insiemi rilevanti* A*,*

$$\mathbb{P}(X_{t+1} \in A | X_0, X_1, \ldots, X_t) = \mathbb{P}(X_{t+1} \in A | X_t) \tag{4.21}$$

ovvero se X_{t+1} *è dipendente dalla variabile immediatamente precedente* X_t *ed è indipendente da tutte le variabili precedenti* $X_0, \ldots, X_{t-1}$*.*

Iniziamo considerando una catena markoviana discreta, ovvero una catena con spazio degli stati discreto e finito. Assumendo omogeneità, la catena viene definita con una matrice di probabilità di transizione da uno stato all'altro

$$P = \begin{bmatrix} p_{1,1} & \ldots & p_{1,n} \\ \vdots & \vdots & \vdots \\ p_{n,1} & \ldots & p_{n,n} \end{bmatrix} \tag{4.22}$$

dove denotiamo gli stati con gli indici $(1, \ldots, n)$ e definiamo la probabilità di passare da uno stato all'altro come

$$p_{i,j} = \mathbb{P}(X_{t+1} = j | X_t = i)$$

e le probabilità degli stati iniziali $\pi_0 = \begin{bmatrix} \pi_{01} & \ldots & \pi_{0n} \end{bmatrix}$ come

$$\pi_{0j} = \mathbb{P}(X_0 = j)$$

Quindi le proprietà stocastiche della catena sono determinate da π_0 e da P. Possiamo quindi scrivere che

$$\pi_t = [\mathbb{P}(X_t = 1), \ldots, \mathbb{P}(X_t = n)] = \pi_{t-1}P = \cdots = \pi_0 P^t$$

Alcune proprietà

- La distribuzione $\pi = [\pi_1, \ldots \pi_n]$ è detta *invariante (o stazionaria)* per la catena markoviana se

  $$\pi P = \pi$$

  cioè se gli stati della catena hanno la stessa probabilità π ad ogni passo della catena.

- Se, al variare di t, la catena oscilla fra due stati essa si dice *periodica*. Sia quindi $\mathbb{P}(X_t = i)$, con $i = 1, \ldots, n$, la probabilità che la catena al passo t assuma lo stato i, avremo che se

  $$\mathbb{P}(X_t = i) = \begin{cases} \mathbb{P}(X_0 = i) & \textit{per t pari} \\ 1 - \mathbb{P}(X_0 = i) & \textit{per t dispari} \end{cases}$$

  la catena è periodica e $\mathbb{P}(X_t = i)$ non converge.

- Se, per $t \to \infty$, $\mathbb{P}(X_t = i) \to \zeta_i$, per ogni i, con $\zeta_i \in (0, 1)$, allora diremo che la catena ha una *distribuzione limite* e questo implica che, asintoticamente, il comportamento della catena non dipende dalle condizioni iniziali. Questo non può verificarsi se la catena è periodica.

- Se la catena ammette una distribuzione limite essa è *invariante*.

- Dati $i, j \in [1, \ldots, n]$, con $i \neq j$, due stati della catena, se il passaggio da i a j è sempre possibile, con un numero sufficiente di passi, allora la catena si dice *irriducibile*; altrimenti essa si dice *riducibile*.

- Se il numero atteso di transizioni di ritorno a un qualsiasi stato è finito, allora la catena di dice *ricorrente positiva*, ovvero sia

$$\tau_{ii} = min\{t > 0 : X_t = i | X_0 = i\}$$

la ricorrenza positiva implica che $E(\tau_{ii}) < \infty, \quad \forall i$.

- Una catena markoviana irriducibile, aperiodica e ricorrente positiva è detta *ergodica* e ha un'unica distribuzione invariante che coincide con la distribuzione limite della catena.

- Se esiste una distribuzione $\pi = [\pi_1, \ldots, \pi_m]$ tale che $\pi_i P_{ij} = \pi_j P_{ji}$, la catena si dice *reversibile*.

- Se una catena è reversibile rispetto a π, allora π è una distribuzione invariante.

Nel caso continuo le transizioni da uno stato all'altro sono definite tramite una transizione kernel

$$K(x, y) = K(X_{t+1} = y | X_t = x) \tag{4.23}$$

che svolge una funzione analoga alla matrice di transizione P, cioè stabilisce la densità di probabilità di passare dallo stato x al tempo t allo stato y al tempo *t+1*.

Anche nel caso continuo valgono le proprietà descritte sopra, sebbene le definizioni formali siano più complicate. E' interessante notare che una condizione sufficiente per l'irriducibilità è che $K(x, \cdot) > 0$ ovunque. Per gli algoritmi che seguono le catene che considereremo hanno avere la proprietà di ergodicità, cioè essere irriducibili, aperiodiche e a ricorrenza positiva.

Una delle utilità delle catene markoviane è che permettono di simulare facilmente valori. Infatti, dato uno stato iniziale e una densità di transizione kernel, possiamo simulare X_0 secondo la distribuzione iniziale π_0 e, poi, via via, utilizzare il valore X_j per simulare il valore X_{j+1} secondo la matrice di transizione. Dopo un numero sufficientemente grande N di passi si arriva a simulare approssimativamente dalla distribuzione limite π, indipendentemente dal valore di partenza scelto, come visto nelle proprietà sopra descritte. Avremo quindi che i valori $X_{N+1}, X_{N+2}, \ldots$ hanno distribuzione approssimativamente pari a π. E' importante notare che i valori $x_{N+1}, x_{N+2}, \ldots$ sono tra loro dipendenti e non costituiscono quindi un campione casuale semplice.

4.2.2 Metropolis-Hastings

I metodi MCMC (Markov Chain Monte Carlo) sono degli algoritmi che permettono di ottenere un campione dalla distribuzione desiderata basandosi sulla costruzione di una catena markoviana che abbia tale distribuzione come distribuzione limite.

Uno dei più conosciuti ed utilizzati è l'algoritmo Metropolis-Hastings, nel quale il generico valore x_{t+1} viene generato utilizzando una funzione di densità $q(\cdot|x_t)$, detta distribuzione *strumentale* o *proposta* che non è, ovviamente, la densità di transizione kernel della catena generata dall'algoritmo. La seconda componente è una funzione di confronto tra il valore x_t e il nuovo valore proposto x^*_{t+1}, definita come

$$\alpha(x_t, x^*_{t+1}) = \min\left\{1, \frac{f(x^*_{t+1})q(x_t|x^*_{t+1})}{f(x_t)q(x^*_{t+1}|x_t)}\right\} \tag{4.24}$$

dove $f(\cdot)$ è la vera funzione da cui vogliamo generare il campione. Grazie ad $\alpha(x_i, x^*_{t+1}) = \alpha$ decidiamo se accettare o rifiutare il nuovo valore proposto. Si noti che $f(\cdot)$ può essere considerata a meno di costanti di proporzionalità in quanto esse si eliminerebbero in (4.24). Avremo quindi che

$$x_{t+1} = \begin{cases} x^*_{t+1} & \textit{con probabilità } \alpha \\ x_t & \textit{con probabilità } 1-\alpha \end{cases}$$

La densità di transizione kernel è quindi definita come

$$K(x_{t+1}, x_t) = q(x_{t+1}|x_t)\alpha(x_t, x_{t+1}) + I(x_{t+1} = x_t)\int q(y|x_t)\left\{1 - \alpha(x_t, y)\right\} dy \tag{4.25}$$

E' semplice dimostrare come una catena markoviana così costruita rispetti le condizioni di reversibilità e stazionarietà della distribuzione, cioè che

- $f(x_t)K(x_t, x_{t+1}) = f(x_{t+1})K(x_{t+1}, x_t)$
- $f(x_t) = f(x_{t+1})$

4.2.3 Gibbs Sampler

Il Gibbs Sampler è un caso particolare di metodo MCMC nel caso multivariato dove si pone la funzione di densità proposta

$$q_j(x_j|\mathbf{x}) = f(x_j|\mathbf{x}_{(j)}) \tag{4.26}$$

dove $\mathbf{x}_{(j)}$ è il vettore $\mathbf{x}$ senza la j-esima componente. Stiamo quindi considerando le distribuzione di densità *full-conditional* delle variabili considerate condizionatamente a tutte le altre. L'idea di base è che, in una distribuzione multivariata è più semplice campionare dalle distribuzioni condizionate che da quelle marginali ottenute tramite integrazione sulla distribuzione congiunta. Avremo quindi che i valori sono generati utilizzando le distribuzioni di densità *full-conditional* a posteriori, dove, ad ogni valore x_j generato, esso viene subito utilizzato per tutte le altre $f(x_i|\mathbf{x}_{(i)})$, con $i \neq j$. Generalmente questo algoritmo produce dei campioni con buone proprietà, anche se la controindicazione è che può essere usato solo se le funzioni di densità condizionate sono note. In ogni caso, se questo non fosse possibile, si può sempre utilizzare un passo di Metropolis-Hastings, un altro algoritmo della famiglia di metodi MCMC, per generare il valore dalla funzione di densità non nota, a meno di una costante di proporzionalità.

4.3 Julia

Julia è un linguaggio di programmazione Open Source di recente sviluppo, ad oggi ancora alla versione beta 0.3.11, ma nonostante questo già molto robusto ed efficiente. E' un linguaggio di alto livello, dinamico e con notevoli performance, in molti casi paragonabili o molto vicine a quelle dei linguaggi compilati come C. Inoltre è stato sviluppato specificatamente per l'ambiente scientifico e, per questo motivo, ha già nella libreria standard tutti i costrutti che tipicamente possono servire per le analisi matematiche e statistiche. Un'altra caratteristica molto importante in questa epoca sempre più incentrata sulla rete e dove la potenza di calcolo richiesta è sempre più comunemente data da grandi cluster di macchine è il fatto che in Julia sono integrati molteplici strumenti per semplificare l'esecuzione parallela e distribuita del codice, permettendo allo sviluppatore di sfruttare la potenza di calcolo di più processori con una facilità estrema e con una conoscenza minima della materia. Infine, una caratteristica fondamentale per un linguaggio, è la *community*. Infatti, nonostante sia ancora un linguaggio estremamente giovane, ha già un'ampia community alle spalle che produce pacchetti e librerie per ogni esigenza.

4.3.1 Julia rispetto a R

Perché quindi usare Julia rispetto a R, che è sicuramente il linguaggio più usato per l'analisi statistica? Vediamo il confronto tra alcune caratteristiche

dei due linguaggi.

Velocità Le simulazioni usate in questo lavoro sono abbastanza complesse da rendere il tempo di calcolo un problema non trascurabile. In R i cicli *for* e *while* sono notoriamente lenti: per ovviare a questo problema è pratica comune sfruttare il calcolo vettoriale e la famiglia di funzione *apply()*. Il problema è che l'algoritmo Gibbs Sampler, come visto nel capitolo (4.2), si basa sulle catene markoviane, dove i valori simulati sono dipendenti dall'iterazione precedente. Questa dipendenza dal passato rende inattuabile l'utilizzo di *apply()* o la vettorializzazione dell'algoritmo in modo tale da eliminare i cicli *for*. In Julia, invece, questo problema non sussiste e, sebbene sia possibile utilizzare la forma vettoriale, è incoraggiato l'uso dei normali cicli che rendono il codice più chiaro e leggibile.

Un altro problema di R è che gli oggetti passati come argomento alle funzioni non vengono passati per riferimento, ma per valore. Per chiarezza, passare un oggetto per riferimento significa passare l'indirizzo di memoria in cui è contenuto l'oggetto, quindi l'oggetto è presente una sola volta in memoria e verrà condiviso dal codice chiamante e dalla funzione. Passare un oggetto per valore, invece, significa che verrà eseguita una copia dell'oggetto stesso, quindi avremo due indirizzi che puntano a due oggetti uguali, ma completamente indipendenti. Questo è un grosso problema nel caso in cui l'oggetto sia molto pesante, come ad esempio una matrice con un numero elevato di elementi, cosa molto comune in un contesto statistico, perché implica che l'oggetto verrà copiato all'interno dell'environment della funzione. La conseguenza di ciò è che in R la memoria utilizzata è molto maggiore, a causa della copiatura degli oggetti e che non è possibile effettuare operazioni *in-place*, cioè effettuare modifiche sull'oggetto di partenza che vengono viste dal codice che ha invocato la funzione.

Parlando più in generale, la velocità di Julia è data da un compilatore JIT estremamente efficiente che compila «Just In Time» il codice in linguaggio macchina, rendendo l'esecuzione molto più veloce, mentre R è un linguaggio interpretato, cioè un linguaggio in cui le istruzioni vengono lette, interpretate ed eseguite nel momento stesso in cui si presentano. Entrambi i linguaggi consentono la chiamata a funzioni di linguaggi di basso livello come C e Fortran per velocizzare l'esecuzione di parti critiche del codice e per sfruttare le librerie scientifiche robuste e collaudate come *LAPACK*, anche se in Julia la libreria non ha nemmeno bisogno di essere caricata e la chiamata alle funzioni delle

	Fortran	Julia	Python	R	Matlab	Java
	gcc 4.8.2	0.3.7	2.7.9	3.1.3	R2014a	1.7.0_75
fib	0.57	2.14	95.45	528.85	4258.12	0.96
parse_int	4.67	1.57	20.48	54.30	1525.88	5.43
quicksort	1.10	1.21	46.70	248.28	55.87	1.65
mandel	0.87	0.87	18.83	58.97	60.09	0.68
pi_sum	0.83	1.00	21.07	14.45	1.28	1.00
rand_mat_stat	0.99	1.74	22.29	16.88	9.82	4.01
rand_mat_mul	4.05	1.09	1.08	1.63	1.12	2.35

Tabella 2: Tempi di benchmark relativi a C (1.00 significa che il tempo di esecuzione è uguale a quello di C).

librerie è assolutamente trasparente.

A causa della lentezza intrinseca di R, generalmente «il vero lavoro» degli algoritmi viene eseguito in C: questo rende difficilmente tracciabile ciò che viene eseguito a runtime, soprattutto in caso di errori. Per contro, la velocità di Julia permette di non affidarsi così tanto a linguaggi di basso livello, se non per librerie come quelle discusse sopra, le quali non avrebbe senso riscrivere da zero. A prova di ciò è interessante notare come il linguaggio e la libreria standard siano quasi completamente definiti e scritti in Julia stesso.

Nella tabella 2, presa dal sito ufficiale di Julia e riguardante alcuni benchmark eseguiti su vari linguaggi comunemente usati in ambiente scientifico, si può vedere la notevole differenza di prestazione tra R e Julia. Segue una breve descrizione degli algoritmi di benchmark utilizzati:

- fib: calcolo della successione di Fibonacci
- parse_int: trasformazioni di stringhe in numeri interi
- quicksort: ordinamento di un vettore numerico
- mandel: prestazioni dei numeri in virgola mobile tramite il calcolo del frattale di Mandelbrot
- pi_sum: prestazione delle somme tramite il calcolo di π
- randmatstat e randmatmult: prestazioni dei calcoli matriciali attraverso il calcolo di statistiche base, nel primo caso, e del prodotto matriciale, nel secondo.

Esempio Vediamo ora un semplice esempio che ci permetterà di vedere le differenze in sintassi, velocità e caratteristiche tra Julia ed R. Visti gli obbiettivi di questo lavoro e visto che Julia verrà utilizzato principalmente per la creazione di un Gibbs Sampler, ne costruiamo uno semplicissimo per simulare da una distribuzione normale bivariata con $\mu = 0$, entrambe le varianze uguali a 1 e correlazione ρ. Si dimostra facilmente che le distribuzioni condizionate sono

$$X|Y = y \sim N(\rho y, 1 - \rho^2)$$

e

$$Y|X = x \sim N(\rho x, 1 - \rho^2)$$

Il codice R e Julia sono riportati di seguito.

```
r_gibbs <- function(nsim, rho, x0, y0)
{
    # creo una matrice di zeri per i risultati
    out <- matrix(0, nrow = nsim, ncol = 2)
    # imposto i valori iniziali per x e y
    x <- x0
    y <- y0
    # per nsim volte genero valori per x e y
    # e li inserisco nella matrice di output
    for (i in 1:nsim)
    {
        x <- rnorm(1, rho*y, sqrt(1 - rho^2))
        y <- rnorm(1, rho*x, sqrt(1 - rho^2))
        out[i,] <- c(x, y)
    }
    # restituisco la matrice di output
    out
}

system.time({res <- r_gibbs(10^7, 0.5, 1, 1)})
```

Il tempo totale di esecuzione è risultato essere 88.62 secondi.

```
using Distributions

function julia_gibbs(nsim, ρ, x0, y0)
    # creo una matrice di zeri per i risultati
```

```
    out = zeros(nsim, 2)
    # imposto i valori iniziali per x e y
    x = x0
    y = y0
    # per nsim volte genero valori per x e y
    # e li inserisco nella matrice di output
    for i = 1:nsim
        x = rand(Normal(ρ*y, sqrt(1 - ρ^2)))
        y = rand(Normal(ρ*x, sqrt(1 - ρ^2)))
        out[i, 1:end] = [x, y]
    end
    # restituisco la matrice di output
    out
end

@time res = julia_gibbs(10^7, 0.5, 1.0, 1.0)
```

In questo caso il tempo totale di esecuzione è risultato di soli 1.52 secondi.

Come si può vedere, anche in un esempio semplice come questo, la differenza di velocità è abissale e la riduzione effettiva del tempo di esecuzione del codice è di più del 98%.

Come abbiamo detto, Julia è stato pensato tenendo conto della computazione parallela e distribuita. Esistono infatti molti strumenti nella libreria standard definiti a tale scopo, come l'oggetto *DArray* che rappresenta un «Distributed Array», cioè un array spezzato in varie parti, ognuna memorizzata e gestita da un diverso processore o calcolatore. Questo consente di gestire array di dimensioni elevate eliminando di fatto i problemi relativi alla gestione di un oggetto così grande in memoria e velocizza le operazioni effettuate sull'oggetto in quanto ogni processore si occuperà dei calcoli eseguiti sulla propria parte di array. Nonostante la complessità della cosa, la costruzione di un *DArray* è banale e la sua gestione è totalmente trasparente.

```
@everywhere using Distributions

@everywhere function julia_gibbs(nsim, ρ, x0, y0)
    # creo una matrice di zeri per i risultati
    out = zeros(nsim, 2)
    # imposto i valori iniziali per x e y
    x = x0
```

```
    y = y0
    # per nsim volte genero valori per x e y
    # e li inserisco nella matrice di output
    for i = 1:nsim
        x = rand(Normal(ρ*y, sqrt(1 - ρ^2)))
        y = rand(Normal(ρ*x, sqrt(1 - ρ^2)))
        out[i, 1:end] = [x, y]
    end
    # restituisco la matrice di output
    out
end

@time resd = DArray(
    I -> julia_gibbs(map(length, I)[1], 0.5, 1.0, 1.0),
    (10^7, 2))
mean(resd, 1) # o altre operazioni sulla matrice...
```

Il tempo di esecuzione è di 0.48 secondi sullo stesso calcolatore utilizzando 4 processori, effettivamente riducendo di 4 volte il tempo di esecuzione, trascurando ovviamente il tempo di creazione dei processi. L'unica accortezza è di utilizzare la macro *@everywhere* davanti ai pacchetti e alle funzioni che si vogliono esportare sugli altri processori, in quanto esse sono definite di default unicamente sul processo principale. Anche se può sembrare inizialmente complicato, l'oggetto DArray vuole come primo argomento una funzione che accetta come argomento una t-upla di intervalli di indici, uno per dimensione, che rappresentano la parte di array concessa a ciascun processo e come secondo argomento una t-upla contenente le dimensioni finali dell'array che vogliamo creare. Nel nostro caso abbiamo una matrice $10^7 \times 2$ e ad ogni processo verrà concessa una matrice $\frac{10^7}{4} \times 2$. Utilizzando la scrittura *x -> f(x)* come argomento di una funzione, si definisce una funzione anonima con argomento x e corpo della funzione *f(x)*. L'equivalente nel linguaggio R si ottiene con la scrittura *function(x) f(x)*.

Il primo argomento nella funzione *julia_gibbs* semplicemente decide quanti valori simulare in base al numero di righe della parte di matrice concessa al processo. Nonostante l'oggetto risultante *resd* sia un array distribuito su più processi si possono effettuare operazioni globali come la media per colonna *mean(resd, 1)* all'ultima riga di codice e l'oggetto DArray si occuperà di passare l'operazione ad ogni processo e restituire il risultato come se l'operazione fosse

stata svolta in locale.

Sintassi La sintassi di Julia cerca di prendere il meglio dai più comuni linguaggi usati per analisi scientifiche come R e Python, MATLAB, C e altri, e grazie a questo riesce ad essere al contempo molto espressiva, ma chiara e semplice da leggere anche per i non esperti.

Come abbiamo visto nell'esempio riportato sopra, la sintassi risulta molto simile o comunque familiare per chi già conosce R e permette anche l'uso di simboli e lettere greche utilizzando la notazione LaTeX che, visto l'ambiente scientifico per cui è stato costruito il linguaggio, consente di scrivere formule matematiche nel codice in modo più chiaro e leggibile. Anche se non obbligatorio è anche possibile specificare il tipo di oggetto nella definizione di nuove variabili, classi o funzioni. Questo ha un duplice vantaggio: in fase di scrittura del codice permette di avere la possibilità di *type-checking* in modo da essere sicuri, almeno per quanto riguarda il tipo degli oggetti che si usano, che il codice faccia ciò che lo sviluppatore intende e in fase di runtime aiuta il compilatore JIT a tradurre la funzione in linguaggio macchina utilizzando i tipi specificati nel codice. Se non specificati, il compilatore deve aspettare che a runtime venga invocata la funzione, capire quali sono i tipi degli argomenti e il tipo di ritorno della funzione e solo allora tradurla.

In definitiva, Julia rappresenta un ottimo compromesso tra la velocità di un linguaggio compilato e la potenza espressiva dei linguaggi moderni di alto livello.

5 Il modello di base

In questo capitolo verrà presentato il modello bayesiano utilizzato così come proposto da Griggio [2014]. Innanzitutto spieghiamo il motivo per cui è stato scelto un approccio bayesiano: i dati a disposizione sono distribuiti su più anni ed è plausibile supporre che i tassi di nascita siano in qualche modo influenzati dal comportamento degli anni precedenti a causa di motivazioni politiche, economiche, sociali o altro. Per questo motivo la scelta di considerare un modello bayesiano dove utilizziamo l'informazione data dalle stime a posteriori dell'anno $t-1$ per definire la distribuzione a priori dei parametri per l'anno t sembra appropriata.

5.1 Le osservazioni

Partiamo dalla definizione delle osservazioni che verranno utilizzate per il modello. Ciò che andremo ad osservare è che un figlio è nato in una certa classe di età della madre, dove nel nostro caso la classe di età corrisponde a un anno. Possiamo quindi pensare che le nostre osservazioni sono dei vettori $\mathbf{x}_i = [x_{i,12}, \ldots, x_{i,j}, \ldots, x_{i,55}]$, dove

$$x_{i,j} = \begin{cases} 1 & se\ y_i = j \\ 0 & altrimenti \end{cases}$$

dove $y_i = j$ significa che nell'i-esima nascita osservata la madre ha età j. In altre parole abbiamo che

$$\mathbf{x}_i \sim MN(1, p_1, \ldots, p_J) \tag{5.1}$$

con $p_1, \ldots, p_J$ le probabilità delle J classi di età, da 12 a 55, e dove il generico p_j è uguale a

$$p_j = \mathbb{P}(y_i = j) = \int_{j-1/2}^{j+1/2} f_{SN}(t; \xi, \omega^2, \alpha)dt \tag{5.2}$$

dove f_{SN} indica la funzione di densità di una variabile normale asimmetrica.

5.2 Variabili latenti

Sono state introdotte delle variabili latenti continue $v_1, \ldots, v_n$ aventi distribuzione $SN(\xi, \omega^2, \alpha)$ e rispetto a queste osserviamo y_i dove $y_i = j$ se $j - 1/2 \leq v_i < j + 1/2$. Queste variabili latenti hanno permesso di trovare delle distribu-

zioni *full conditional* per le a priori e a posteriori dei parametri del modello ξ, ω, α che vediamo qui di seguito.

Date queste variabili latenti, possiamo definire la distribuzione a posteriori congiunta del vettore di parametri $(\xi, \omega^2, \alpha, V)$ nel modo seguente:

$$\pi(\xi, \omega^2, \alpha, V|Y) \propto f_\xi(\cdot) f_{\omega^2}(\cdot) f_\alpha(\cdot) \prod_{i=1}^{n} f_{SN}(v_i; \xi, \omega^2, \alpha) x \left\{ \sum_{i=12}^{55} I(y_i = j) I\left(j - 1/2 \leq v_i < j + 1/2\right) \right\} \quad (5.3)$$

dove f_ξ, f_{ω^2} e f_α sono le funzioni di densità a priori dei relativi parametri che verranno definiti qui di seguito, $f_{SN}(v_i; \xi, \omega^2, \alpha)$ è la funzione di densità normale asimmetrica di parametri ξ, ω^2, α e $I(\cdot)$ indica una funzione indicatrice.

5.3 Distribuzione di α

Al fine di ottenere la distribuzione α condizionatamente a tutti gli altri parametri, in questo paragrafo consideriamo i parametri ξ, ω noti e, senza perdita di generalità, poniamo $\xi = 0$ e $\omega = 1$. Nell'algoritmo finale i parametri verranno ovviamente considerati tutti ignoti. La verosimiglianza del campione *iid* $v = (v_1, \ldots, v_n)$ possiamo quindi scriverla come

$$f(v; \alpha) = \prod_{i=1}^{n} 2\phi(v_i)\Phi(\alpha v_i) \quad (5.4)$$

Canale and Scarpa [2013] hanno proposto due distribuzioni per la a priori di α: una normale, che può essere scelta con l'obiettivo di centrare la a priori su una particolare ipotesi di α, e una normale asimmetrica, che è la distribuzione che verrà qui considerata.

Assumiamo quindi che α sia distribuito a priori come una normale asimmetrica

$$\alpha \sim SN(\alpha_0, \psi_0^2, \lambda_0) = \frac{2}{\psi_0} \phi\left(\frac{\alpha - \alpha_0}{\psi_0}\right) \Phi\left(\lambda_0 \frac{\alpha - \alpha_0}{\psi_0}\right) \quad (5.5)$$

dove α_0 è l'iperparametro di posizione, ψ_0 è l'iperparametro di scala e λ_0 è l'iperparametro di forma che regola la direzione e l'intensità dell'asimmetria. Nella simulazione pratica porremo $\alpha_0 = 0$. La conseguenza di ciò è che l'informazione a priori del parametro α riguarderà così solo il lato dell'asimmetria: la distribuzione a priori di α sarà centrata in zero e l'iperparametro di scala sarà impostato con un valore alto per non limitare troppo, a priori, la scelta dei va-

lori generati casualmente dal Gibbs Sampler. Avremo quindi che il parametro che influenzerà maggiormente la distribuzione a priori è quello di forma che ci permetterà di fare assunzioni a priori sul lato dell'asimmetria della normale asimmetrica utilizzando valori positivi o negativi per λ_0.

Di conseguenza la distribuzione a posteriori di α diventa

$$\begin{aligned}\pi(\alpha|v) &\propto \phi\left(\frac{\alpha-\alpha_0}{\psi_0}\right)\Phi\left(\lambda_0\frac{\alpha-\alpha_0}{\psi_0}\right)\prod_{i=1}^{n}\phi\left(\alpha v_i\right)\\ &\propto \phi\left(\frac{\alpha-\alpha_0}{\psi_0}\right)\Phi_{n+1}\left(\begin{bmatrix}v\alpha_0\\0\end{bmatrix}+\begin{bmatrix}v\\ \lambda_0/\psi_0\end{bmatrix}(\alpha-\alpha_0);I_{n+1}\right)\end{aligned} \tag{5.6}$$

dove I_d è una matrice identità di dimensione d. La densità a posteriori di α appartiene quindi alla classe di distribuzioni SUN e può essere scritta come

$$\alpha|v \sim SUN_{1,n+1}(\alpha_0,\gamma,\psi_0,1,\Delta,\Gamma) \tag{5.7}$$

con $\Delta = [\delta_i]_{i=1,\dots,n+1}$, dove $\delta_i = \psi_0 z_i\left(\psi_0^2 z_i^2+1\right)^{-1/2}$ e $z = \left[v^T, \lambda_0\psi_0^{-1}\right]^T$, $\gamma = \left[\Delta_{1:n}\alpha_0\psi_0^{-1}, 0\right]$ e $\Gamma = I - D(\Delta)^2 + \Delta\Delta^T$, dove con la scrittura $D(V)$ si intende una matrice diagonale costruita con gli elementi di V.

5.3.1 Sfruttare l'informazione per l'elicitazione a priori

Come abbiamo detto all'inizio di questo capitolo, uno dei motivi principali dell'utilizzo di un approccio bayesiano è la possibilità di sfruttare l'informazione dell'anno $t-1$ per la stima dei parametri dell'anno t. Come si è visto, però, per il parametro α non abbiamo una coniugazione tra le distribuzioni a priori e a posteriori, rendendo impossibile l'utilizzo della semplice stima a posteriori come parametro della a priori dell'anno successivo. Un metodo alternativo è utilizzare le relazioni che intercorrono tra i momenti della famiglia di distribuzioni normali asimmetriche e i suoi parametri. Come visto in (4.1.1), i primi quattro momenti della distribuzione $SN(\xi,\omega^2,\alpha)$ sono

$$E(Y) = \xi + \omega\sqrt{2/\pi}\delta$$

$$var(Y) = \omega^2(1-(2/\pi)\delta^2)$$

$$\gamma_1(Y) = \frac{4-\pi}{2}\left(\frac{\mu_Z^2}{\sigma_Z^2}\right)^2$$

$$\gamma_2(Y) = 2(\pi-3)\left(\frac{\mu_Z^2}{\sigma_z^2}\right)^2$$

con $\delta = \alpha/\sqrt{1+\alpha^2}$, $\mu_Z = \sqrt{2/\pi}\delta$ e $\sigma_Z^2 = 1 - (2/\pi)\delta^2$. Utilizzando i momenti della distribuzione a posteriori di α per l'anno $t-1$ possiamo ottenere il parametro α per la a priori dell'anno t semplicemente invertendo la terza equazione.

5.4 La variabile latente η

Per ottenere una distribuzione coniugata per i parametri di posizione e di scala, condizionatamente a tutti gli altri, dobbiamo introdurre delle ulteriori variabili latenti normali standard $\eta_1, \ldots, \eta_n$. Utilizzando l'equazione (4.17) che definisce la rappresentazione stocastica di una distribuzione normale asimmetrica attraverso due variabili normali standard indipendenti fra loro, notiamo che

$$Y|\eta = \xi + \omega Z = \xi + \omega\left(\delta|\eta| + \sqrt{1-\delta^2}U\right) \sim N(\xi + \omega\delta|\eta|, \omega^2(1-\delta^2)) \quad (5.8)$$

dove $Y \sim SN(\xi, \omega^2, \alpha)$, $Z \sim SN(0, 1, \alpha)$ e $U \sim N(0,1)$ e con $\xi \in \mathbb{R}$, $\omega \in \mathbb{R}^+$, $\delta = \frac{\alpha}{\sqrt{1-\alpha^2}}$ e η un valore generato da una normale standard. Di conseguenza possiamo vedere la generica variabile latente v_i condizionata all'osservazione i-esima distribuita normalmente con media $\xi + \omega\delta|\eta|$ e varianza $\omega^2(1-\delta^2)$.

La distribuzione a posteriori di η condizionatamente agli altri parametri è dunque

$$\pi(\eta|\xi, \omega^2, \alpha, V, Y) \sim TN_0(\delta(v_i - \xi), \omega^2(1-\delta^2)) \quad (5.9)$$

dove con la scrittura $TN_\gamma(\mu, \sigma^2)$ si intende una distribuzione normale con media μ e varianza σ^2 troncata a sinistra di γ.

5.5 Distribuzione di ξ e ω

Grazie all'introduzione delle variabili casuali η_i è stato possibile trovare una coniugazione per i parametri ξ e ω. Infatti, ponendo come a priori per i due parametri le distribuzioni

$$\xi \sim N(\xi_0, \kappa\omega^2) \quad (5.10)$$

$$\omega^2 \sim GammaInv(a, b) \quad (5.11)$$

è semplice ricavare le distribuzioni a posteriori condizionate, cioè

$$\pi(\xi|\omega^2, \alpha, \eta, V, Y) = N(\hat{\mu}, \hat{\kappa}\omega^2) \quad (5.12)$$

$$\pi(\omega^2|\xi, \alpha, \eta, V, Y) = GammaInv(a + (n+1)/2, b + \hat{b}) \quad (5.13)$$

dove

$$\hat{\mu} = \frac{\kappa \sum_{i=1}^{n}(v_i - \delta\eta_i) + (1-\delta^2)\xi_0}{n\kappa + (1-\delta^2)}$$

$$\hat{\kappa} = \frac{\kappa(1-\delta^2)}{n\kappa + (1-\delta^2)}$$

$$\hat{b} = \frac{1}{2(1-\delta^2)} \left\{ \delta^2 \sum_{i=1}^{n} \eta_i^2 - 2\delta \sum_{i=1}^{n} \eta_i (v_i - \xi) + \sum_{i=1}^{n} (v_i - \xi)^2 + \frac{1-\delta^2}{\kappa} (\xi - \xi_0)^2 \right\}.$$

5.6 Distribuzione di v

Infine possiamo ricavare semplicemente la distribuzione a posteriori delle variabili latenti v_i in quanto

$$\pi(v_i|\xi, \omega^2, \alpha, y_i = j) \propto f_{SN}(v_i; \xi, \omega^2, \alpha) \left\{ \sum_{j=12}^{55} I(y_i = j) I\left(j - 1/2 \leq v_i < j + 1/2\right) \right\}$$

da cui si vede che $v_i|\xi, \omega^2, \alpha, y_i = j$ si distribuisce come una normale asimmetrica troncata a sinistra di $j - 1/2$ e a destra di $j + 1/2$. Quindi la funzione di densità a posteriori di v_i, condizionatamente agli altri parametri, può essere scritta come

$$\pi(v_i|\xi, \omega^2, \alpha, y_i = j) = \begin{cases} C f_{SN}(v_i; \xi, \omega^2, \alpha) & se\ j - 1/2 \leq v_i < j + 1/2 \\ 0 & altrimenti \end{cases} \tag{5.14}$$

dove $C^{-1} = \int_{j-1/2}^{j+1/2} f_{SN}(t; \xi, \omega^2, \alpha) dt$ è la costante di normalizzazione della distribuzione normale asimmetrica troncata.

5.7 I passi dell'algoritmo

L'algoritmo Gibbs Sampler completo proposto da Griggio [2014], ispirandosi a quello costruito da Canale and Scarpa [2013] per il caso di osservazioni continue distribuite come una normale asimmetrica, è composto dai seguenti passi che permettono di aggiornare le stime iterativamente come descritto nel capitolo (4.2.3):

- Aggiornamento di v_i dalla distribuzione

$$v_i|\xi, \omega^2, \alpha, y_i = j \sim TSN_{j-1/2, j+1/2}(\xi, \omega^2, \alpha)$$

dove con $TSN_{a,b}(\cdot)$ si intende una distribuzione normale asimmetrica troncata a sinistra di a e a destra di b.

- Aggiornamento di η_i dalla distribuzione a posteriori condizionata

$$\eta_i|\xi, \omega^2, \alpha, v_i \sim TN_0(\delta(v_i - \xi), \omega^2(1-\delta^2))$$

- Campionare (ξ, ω^2) da

$$N(\hat{\mu}, \hat{\kappa}\omega^2)GammaInv(a + (n+1)/2, b + \hat{b})$$

dove

$$\hat{\mu} = \frac{\kappa \sum_{i=1}^{n}(v_i - \delta\eta_i) + (1-\delta^2)\xi_0}{n\kappa + (1-\delta^2)}$$

$$\hat{\kappa} = \frac{\kappa(1-\delta^2)}{n\kappa + (1-\delta^2)}$$

$$\hat{b} = \frac{1}{2(1-\delta^2)}\left\{\delta^2\sum_{i=1}^{n}\eta_i^2 - 2\delta\sum_{i=1}^{n}\eta_i(v_i - \xi) + \sum_{i=1}^{n}(v_i-\xi)^2 + \frac{1-\delta^2}{\kappa}(\xi - \xi_0)^2\right\}$$

- Campionare α da

$$\alpha \sim \pi(\alpha|v^*)$$

dove $v^* = \frac{v_i - \xi}{\omega}$ per $i = 1, \ldots, n$ e $\pi(\alpha|v)$ è la distribuzione a posteriori (5.7) definita per $\xi = 0$ e $\omega^2 = 1$ alla quale ci siamo ricondotti tramite la trasformazione v^* delle variabili latenti v_i.

6 Il nuovo modello proposto

In questo capitolo verrà spiegato il nuovo modello che permette di includere la stima del parametro R, che indica il tasso di fecondità totale, insieme a quella dei parametri ξ, ω^2, α, dove questi quattro, ricordiamo, sono i parametri che definiscono la funzione di fecondità

$$g(x; R, \xi, \omega^2, \alpha) = R \cdot h(x; \xi, \omega^2, \alpha)$$

cercando di ricondurci ai risultati ottenuti nel capitolo 4.

Se consideriamo la funzione g invece che la funzione h, come nel modello definito nel capitolo 4, ci si accorge subito del problema principale: non stiamo più trattando una funzione di densità. Volendo considerare le osservazioni così come definite nel capitolo 4, ovvero del tipo $\mathbf{x}_i = [x_{i,12}, \ldots, x_{i,j}, \ldots, x_{i,55}]$, con $x_{i,j}$ variabili dicotomiche, e volendo utilizzare anche in questo caso una distribuzione multinomiale avremmo che le probabilità p_j, con $j = 12, \ldots, 55$, di quest'ultima sarebbero pari a

$$p_j = \mathbb{P}(y_i = j) = R \cdot \int_{j-1/2}^{j+1/2} f_{SN}(t; \xi, \omega^2, \alpha) dt$$

il che non è plausibile in quanto $\sum_{j=12}^{55} p_j \neq 1$, proprietà richiesta dalla distribuzione multinomiale.

Possiamo allora pensare di rilassare questa limitazione considerando, anziché un'unica distribuzione multinomiale, una distribuzione binomiale per ogni età x dove la probabilità della generica età x_0 è ancora una volta

$$\pi_{x_0} = R \cdot \int_{x_0-1/2}^{x_0+1/2} f_{SN}(t; \xi, \omega^2, \alpha) dt. \tag{6.1}$$

Poiché stiamo considerando delle distribuzioni binomiali, non dobbiamo più preoccuparci della relazione che le probabilità π_x hanno tra di loro, ma solo che esse siano effettivamente delle probabilità. Le π_x saranno sicuramente non negative, per come sono state costruite ed è ragionevole supporre che siano anche minori di 1: affinché ciò accada, R dovrebbe avere valori altissimi, impossibili da raggiungere nella nostra realtà attuale.

6.1 Le osservazioni

Avendo modificato la distribuzione, anche le osservazioni prese in considerazione sono cambiate. La stima del parametro R rende necessario avere anche informazione sulla quantità di donne presenti nel Paese e non solo sul numero di nati: mentre i parametri ξ, ω^2, α sono relativi alla sola distribuzione dei nati, il parametro R é legato al rapporto fra numero di nati e numero di donne in quanto esprime il tasso di fecondità. Ciò che consideriamo adesso è quindi:

- N_x, il numero totale di figli nati da donne di età x
- D_x, il numero totale di donne di età x

dove $x \in [12, 55]$, l'intervallo di età considerato.

Avremo quindi che N_x sono realizzazioni di variabili casuali binomiali, con indice D_x e probabilità π_x. Possiamo quindi scrivere la funzione di densità di N_x come

$$f(N_x) = \binom{D_x}{N_x} \pi_x^{N_x} (1 - \pi_x)^{D_x - N_x} \tag{6.2}$$

o anche, utilizzando la (6.1),

$$f(N_x) = \binom{D_x}{N_x} R^{N_x} \cdot \left(\int_{x_0 - 1/2}^{x_0 + 1/2} f_{SN}(t; \xi, \omega^2, \alpha) dt \right)^{N_x} \left(1 - R \cdot \int_{x_0 - 1/2}^{x_0 + 1/2} f_{SN}(t; \xi, \omega^2, \alpha) dt \right)^{D_x - N_x}. \tag{6.3}$$

6.2 Il parametro R

Poiché vogliamo utilizzare un approccio bayesiano, dobbiamo definire una distribuzione a priori per R. Data la funzione di verosimiglianza delle nostre nuove osservazioni N_x:

$$L(R, \xi, \omega, \alpha; N_x) = \prod_{x=12}^{55} \binom{D_x}{N_x} R^{N_x} \cdot \left(\int_{x - 1/2}^{x + 1/2} f_{SN}(t; \xi, \omega^2, \alpha) dt \right)^{N_x} \left(1 - R \cdot \int_{x - 1/2}^{x + 1/2} f_{SN}(t; \xi, \omega^2, \alpha) dt \right)^{D_x - N_x} \tag{6.4}$$

la ricerca di una distribuzione coniugata non risulta una strada praticabile. Si è scelto quindi di utilizzare, come a priori per R, una distribuzione Gamma che rispetta il dominio del parametro, $\mathbb{R}^+$, e consente di sfruttare l'elicitazione

a priori in modo semplice per dare informazione su R basandosi sui risultati degli anni precedenti.

Non essendo stato possibile reperire una distribuzione condizionata a posteriori nota per R, utilizziamo un passo di Metropolis-Hastings per generare valori dalla distribuzione a posteriori

$$\pi(R) = Gamma(a_r, b_r) \cdot L(R, \xi, \omega, \alpha; N_x) \tag{6.5}$$

6.2.1 Elicitazione a priori

Per sfruttare l'informazione data dalla stima del parametro R per l'anno successivo non possiamo contare sul fatto di avere una distribuzione a posteriori coniugata, ma possiamo utilizzare i valori generati per avere informazioni su tale distribuzione. Per questo modello si è scelto di impostare i parametri a_r, b_r in modo tale che il valore medio della distribuzione Gamma sia pari al valore stimato per l'anno precedente e la varianza sia un valore fissato, sufficientemente elevato da consentire ai dati del nuovo anno di esprimersi, ma abbastanza piccolo da rendere la distribuzione a priori informativa.

6.3 Gli altri parametri

Per ricondurci ai risultati del capitolo 4, possiamo fare la seguente considerazione, tralasciata nel paragrafo (6.1) per chiarezza: possiamo considerare, in realtà, le stesse osservazioni utilizzate nel capitolo 4, cioè quelle del tipo $\mathbf{x}_i = [x_{i,12}, \ldots, x_{i,j}, \ldots, x_{i,55}]$. Questo implica ridefinire la (6.2) sostituendo a N_x il valore $\sum_{i=1}^{n} x_{i,x}$ che ha lo stesso identico valore: i risultati ottenuti nei precedenti paragrafi rimangono quindi invariati, con la differenza che ora possiamo reintrodurre le variabili latenti $v_1, \ldots, v_n$ definite in (5.2). Condizionatamente a R, parametro che di fatto regola solo la «altezza» della curva di fecondità, le considerazioni fatte nel capitolo 4 valgono ancora. Osserviamo $y_i = j$ quando $j - 1/2 \leq v_i < j + 1/2$ e, attraverso queste variabili latenti, possiamo utilizzare le stesse distribuzioni del capitolo precedente per i parametri ξ, ω^2, α.

6.4 L'algoritmo finale

Vengono quindi riportati di seguito i passi dell'algoritmo Gibbs completo utilizzato per le simulazioni relative ad ogni anno:

- Generazione di un valore dalla distribuzione a posteriori condizionata di R

$$Gamma(a_r, b_r) \cdot L(R, \xi, \omega, \alpha; N_x)$$

tramite un algoritmo Metropolis-Hastings

- Aggiornamento di v_i dalla distribuzione

$$v_i|\xi, \omega^2, \alpha, y_i = j \sim TSN_{j-1/2, j+1/2}(\xi, \omega^2, \alpha)$$

dove con $TSN_{a,b}(\cdot)$ si intende una distribuzione normale asimmetrica troncata a sinistra di a e a destra di b.

- Aggiornamento di η_i dalla distribuzione a posteriori condizionata

$$\eta_i|\xi, \omega^2, \alpha, v_i \sim TN_0(\delta(v_i - \xi), \omega^2(1 - \delta^2))$$

- Campionare (ξ, ω^2) da

$$N(\hat{\mu}, \hat{\kappa}\omega^2) GammaInv(a + (n+1)/2, b + \hat{b})$$

dove

$$\hat{\mu} = \frac{\kappa \sum_{i=1}^{n}(v_i - \delta\eta_i) + (1 - \delta^2)\xi_0}{n\kappa + (1 - \delta^2)}$$

$$\hat{\kappa} = \frac{\kappa(1 - \delta^2)}{n\kappa + (1 - \delta^2)}$$

$$\hat{b} = \frac{1}{2(1 - \delta^2)} \left\{ \delta^2 \sum_{i=1}^{n} \eta_i^2 - 2\delta \sum_{i=1}^{n} \eta_i (v_i - \xi) + \sum_{i=1}^{n} (v_i - \xi)^2 + \frac{1 - \delta^2}{\kappa} (\xi - \xi_0)^2 \right\}$$

- Campionare α da

$$\alpha \sim \pi(\alpha|v^*)$$

dove $v^* = \frac{v_i - \xi}{\omega}$ per $i = 1, \ldots, n$ e $\pi(\alpha|v)$ è la distribuzione a posteriori (5.7) definita per $\xi = 0$ e $\omega^2 = 1$ alla quale ci siamo ricondotti tramite la trasformazione v^* delle variabili latenti v_i.

Al termine dell'anno vengono calcolati gli iperparametri per l'anno successivo:

- $\xi_0 = \hat{\xi}_{t-1}$
- $b_r = \frac{\hat{R}_{t-1}}{V_r}$ e $a_r = V_r \cdot b_r^2$, dove V_r è il valore fissato per la varianza della distribuzione Gamma

- $\lambda_0 = sign(\gamma) \cdot \sqrt{z/(1-z)}$, dove $z = \frac{\gamma^{2/3}}{\sqrt{\left(\frac{2}{\pi}\right)} \cdot \left(\frac{4-\pi}{2}\right)^{2/3} \cdot (1+\gamma^{2/3})}$ e γ è l'indice di asimmetria della distribuzione a posteriori condizionata di α.

7 I risultati ottenuti

Vediamo ora i risultati ottenuti con il modello proposto. I dati a disposizione sono relativi alla numerosità delle donne e dei nuovi nati, suddivisi per età delle donne e per anno, come già visto nel paragrafo (2.2). A partire da questi è stata creata, per ogni anno, la variabile $\mathbf{y} = (y_1, \ldots, y_N)$, dove $y_i = j$ dove j è pari all'età della madre alla nascita del figlio, ripetendo N_j volte il valore j, per tutti i j da 12 a 55, dove N_j è il numero di nati da madri di età j. Avremo quindi un vettore $\mathbf{y}$ con un numero di elementi pari a $\sum_{j=12}^{55} N_j$, da cui estraiamo un campione casuale semplice di 1000 unità che rappresenta bene l'insieme totale delle osservazioni e, allo stesso tempo, permette di alleggerire un po' le simulazioni. Basti pensare che per la generazione di valori dalla distribuzione *SUN* utilizzata per α bisogna lavorare con matrici $(n+1)\times(n+1)$ che diventano rapidamente ingestibili al crescere della numerosità campionaria.

Le simulazioni sono state eseguite per gli anni dal 1980 al 2012 e per ogni anno il numero di iterazioni è stato pari a 10000, al lordo del *burn-in*, fissato a 500 iterazioni.

7.1 Scelta dei parametri iniziali

Per ogni anno imposteremo gli iperparametri del modello basandoci sulle stime ottenute nell'anno precedente, come specificato in (6.4). Per il primo anno, però, non abbiamo nessuna informazione precedente e per questo motivo si potevano scegliere due strade: utilizzare delle a priori non informative, oppure reperire l'informazione in altro modo, utilizzando un approccio bayesiano empirico. È stata scelta questa seconda opzione ed è stata utilizzata una stima del modello del primo anno, il 1980, utilizzando il metodo dei minimi quadrati, tramite il quale sono stati impostati gli iperparametri a_r e b_r relativi alla distribuzione a priori del parametro R, a e b, iperparametri della Gamma Inversa utilizzata per ω^2 e l'iperparametro λ_0 che gestisce l'asimmetria della distribuzione a priori di α. I parametri κ, ϕ_0 e α_0 sono stati mantenuti costanti in tutte le simulazioni. Ai primi due, che regolano la variabilità del parametro ξ e α, rispettivamente, sono stati assegnati valori sufficientemente elevati da consentire ai dati di esprimersi, in particolare utilizziamo $\kappa = 30$ e $\phi_0 = 10$. L'iperparametro α_0, invece, è stato impostato a 0, come discusso in Canale and Scarpa [2013]: questa soluzione ci permette di avere un'informazione a priori solo sul lato dell'asimmetria della distribuzione dei dati.

7.2 La stima dei parametri

In questo paragrafo verranno riportati i risultati ottenuti dall'algoritmo Gibbs costruito secondo il modello proposto in (6.4). Il numero totale di iterazioni per ogni anno è stato impostato a 10000, con un *burn-in* iniziale di 500 iterazioni, in quanto la convergenza veniva raggiunta piuttosto rapidamente. Visto il grosso carico computazionale, abbiamo deciso di utilizzare un algoritmo che, dopo il periodo di burnin iniziale, veniva fatto girare in parallelo su più processori utilizzando come valore di partenza l'ultimo valore generato nel burnin.

Trattiamo ora, a titolo di esempio, l'anno 1996, ma i grafici di tutti gli anni possono essere reperiti nell'appendice. Per questo anno, le stime e i relativi standard error sono riportati nella tabella seguente.

ξ		ω^2		α		R	
stima	std. err.	stima	std. err.	stima	std. err.	stima	std. err.
26.294	1.307	25.984	4.797	0.755	0.465	1.419	0.00641

Tabella 3: Stime dei parametri per l'anno 1996.

Come si può notare, lo standard error è molto concentrato per il parametro R, ma non per ω^2 e soprattutto per α, considerato il peso di questo parametro, per il quale anche una piccola variazione apporta una notevole modifica alla forma della curva ottenuta.

Vediamo dai trace plot in Figura 3 che le stime non sembrano avere un andamento particolare con il passare delle iterazioni.

Vediamo anche il grafico delle distribuzioni dei parametri a posteriori dei quattro parametri in figura 4. Tutte le distribuzioni hanno delle code molto pesanti, le prime due a destra, le seconde due a sinistra e il parametro ω^2 non ha un chiaro picco, ma piuttosto una densità abbastanza costante in tutti i valori tra 20 e 26.

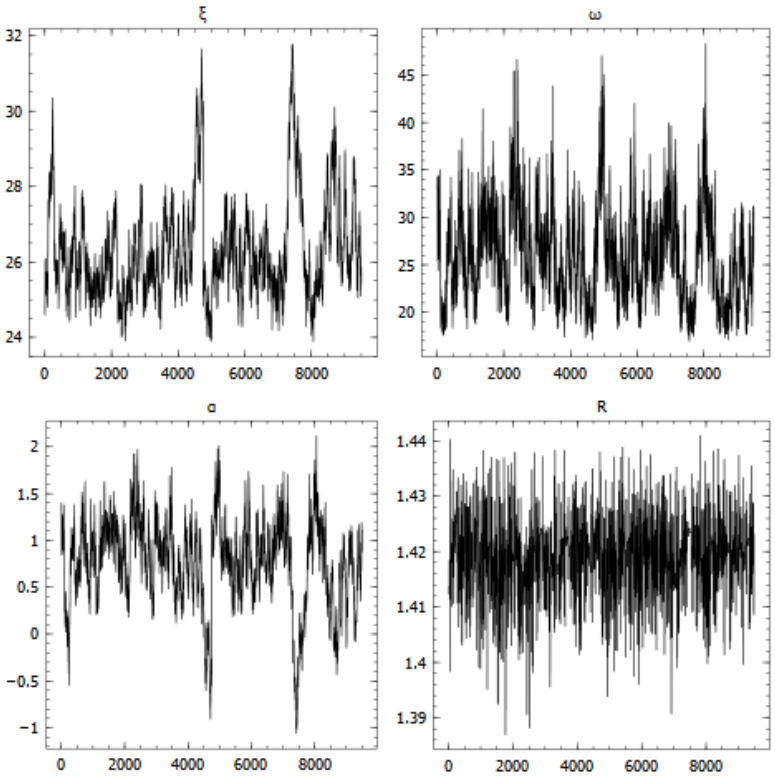

Figura 3: Trace plot delle stime dei quattro parametri nell'anno 1996.

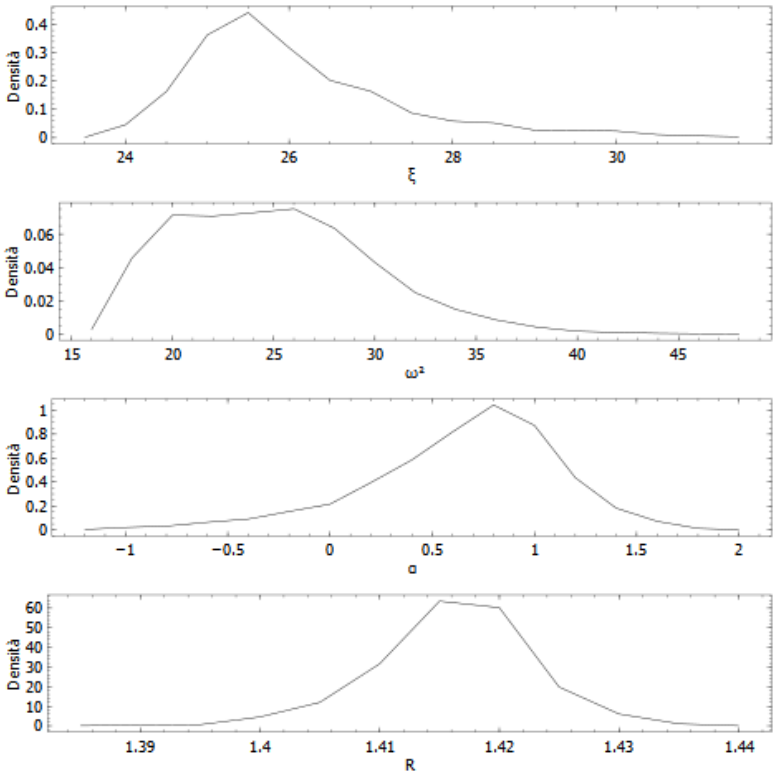

Figura 4: Densità a posteriori dei parametri ξ, ω^2, α e R

Nella Figura 5 possiamo vedere i dati relativi all'anno 1996, raffigurati come pallini e la curva ottenuta con i valori stimati dei parametri, rappresentata dalla linea continua. Vediamo come i parametri stimati permettano alla curva di seguire abbastanza bene l'andamento dei dati, anche se non in modo perfetto.

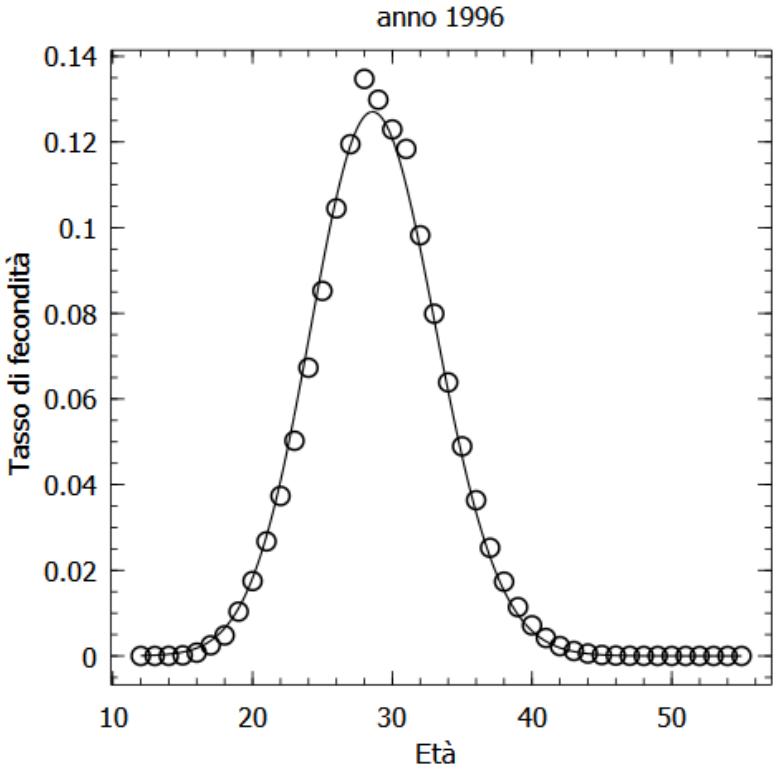

Figura 5: La curva di fecondità stimata per l'anno 1996

Per questo motivo confrontiamo anche la curva del modello con quella ottenuta tramite la stima ai minimi quadrati riportata in Figura 6 come una linea tratteggiata, in cui si può vedere che, effettivamente la più semplice stima ai minimi quadrati sembri approssimare meglio i dati di quanto faccia il più complesso modello bayesiano, quando invece ci si aspetterebbe un'efficacia uguale se non maggiore, vista anche la presenza di informazione a priori in quest'ultimo.

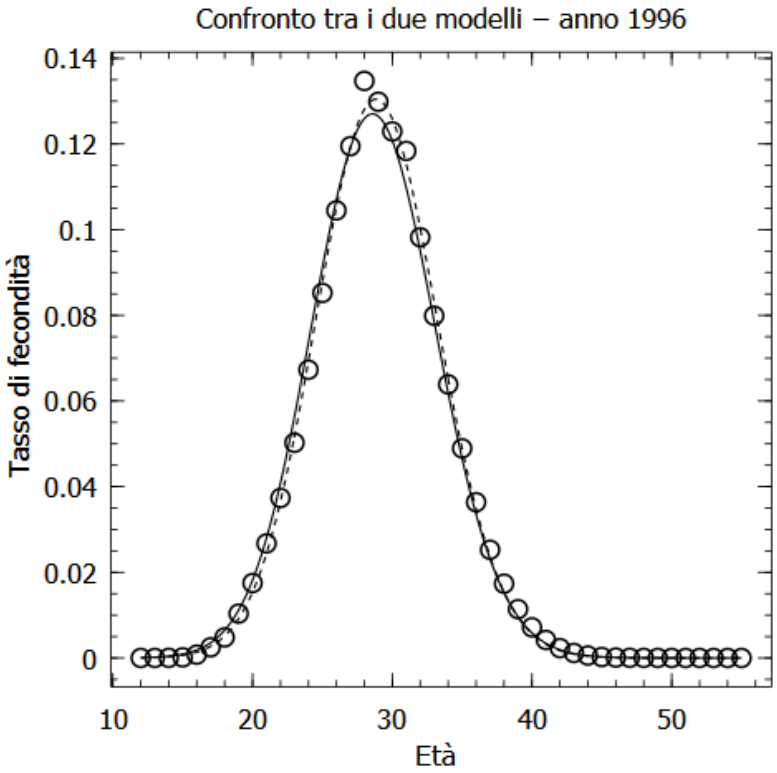

Figura 6: Confronto fra il modello proposto e quello con le stime ai minimi quadrati per l'anno 1996.

Riportiamo in Figura 7 anche i grafici del confronto tra i due tipi di stima del modello per gli anni 1998 e 2004, dove si ha ancora maggior evidenza della diversità di prestazione.

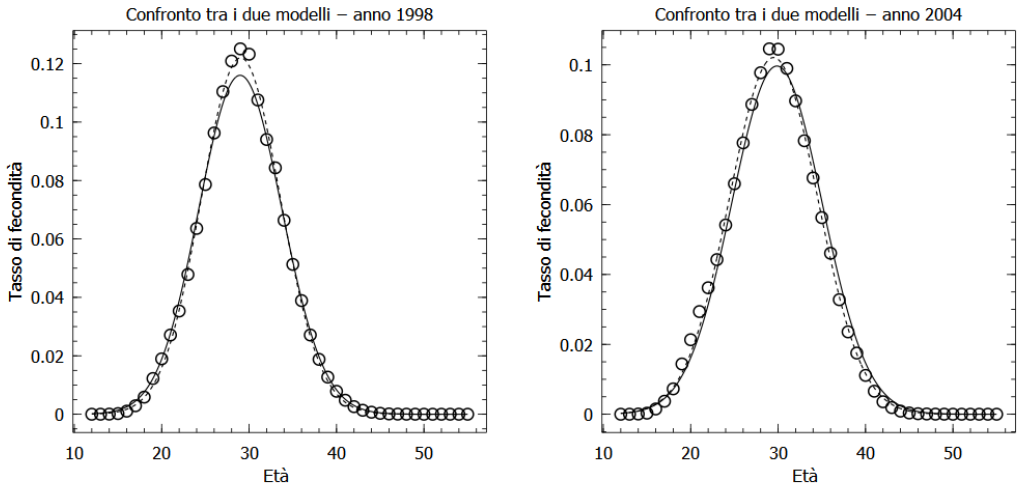

Figura 7: Confronto fra il modello proposto e quello con le stime ai minimi quadrati per gli anni 1998 e 2004.

Questi problemi potrebbero forse derivare da qualche imperfezione nella scelta delle distribuzioni a priori o nel procedimento di stima utilizzato in quanto, dopo un'attenta analisi dei risultati, le imprecisioni maggiori sembrerebbero provenire dai parametri ω^2 e α. Sarebbe quindi interessante, in un lavoro futuro, riverificare la validità complessiva delle stime e capire la causa delle imprecisioni. La stima di ξ sembra buona, ma probabilmente la generazione di valori casuali dalla sua distribuzione è influenzata negativamente dai problemi degli altri due parametri. La stima di R sembra invece ottima: ha una variabilità bassissima ed è sempre estremamente vicina al valore del tasso di fecondità, cioè il valore di R ottenuto tramite il rapporto tra numero di nati e numero di donne, e alla stima ottenuta utilizzando i minimi quadrati.

In ogni caso il modello ha prodotto buoni risultati ed è interessante vedere come il valore dei parametri si evolve nel corso degli anni perché ci permette di trarre informazioni sull'aspetto demografico del Giappone. Analizziamo quindi i seguenti grafici riportati in Figura 8. Vediamo il trend nettamente positivo del parametro ξ ad indicare come l'età media in cui le donne hanno un figlio è aumentata molto negli anni. Il grafico di ω^2 ha un trend leggermente positivo, ma per lo più costante a parte qualche picco sporadico. Il parametro α è calato verso lo zero e tra gli anni 2003 - 2005 lo ha addirittura superato: questo significa che in questi ultimi anni la curva di fecondità è diventata molto più simmetrica che in passato e che nel biennio 2003 - 2005 l'asimmetria era addirittura invertita. Dal grafico di R, invece, notiamo come il tasso di fecondità sia diminuito molto nel ventennio che va dal 1985 al 2005, ma negli

ultimi anni ha ripreso a crescere. Da tutti i grafici possiamo comunque vedere due picchi importanti nel 2003 e nel 2005 che sembrano aver dato inizio a una nuova fase demografica in Giappone in quanto i valori dei parametri di posizione, scala e forma sembrano avere un andamento più costante, privo di trend, e il tasso di fecondità ha ripreso a crescere. Vediamo inoltre un altro picco molto pronunciato negli anni 1990 e 1991 che ha portato ad un aumento improvviso del parametro di posizione e una forte diminuzione dei parametri di scala e di forma, lasciando quasi inalterato il parametro R, ma che non sembra aver influenzato l'evoluzione generale dei 4 parametri.

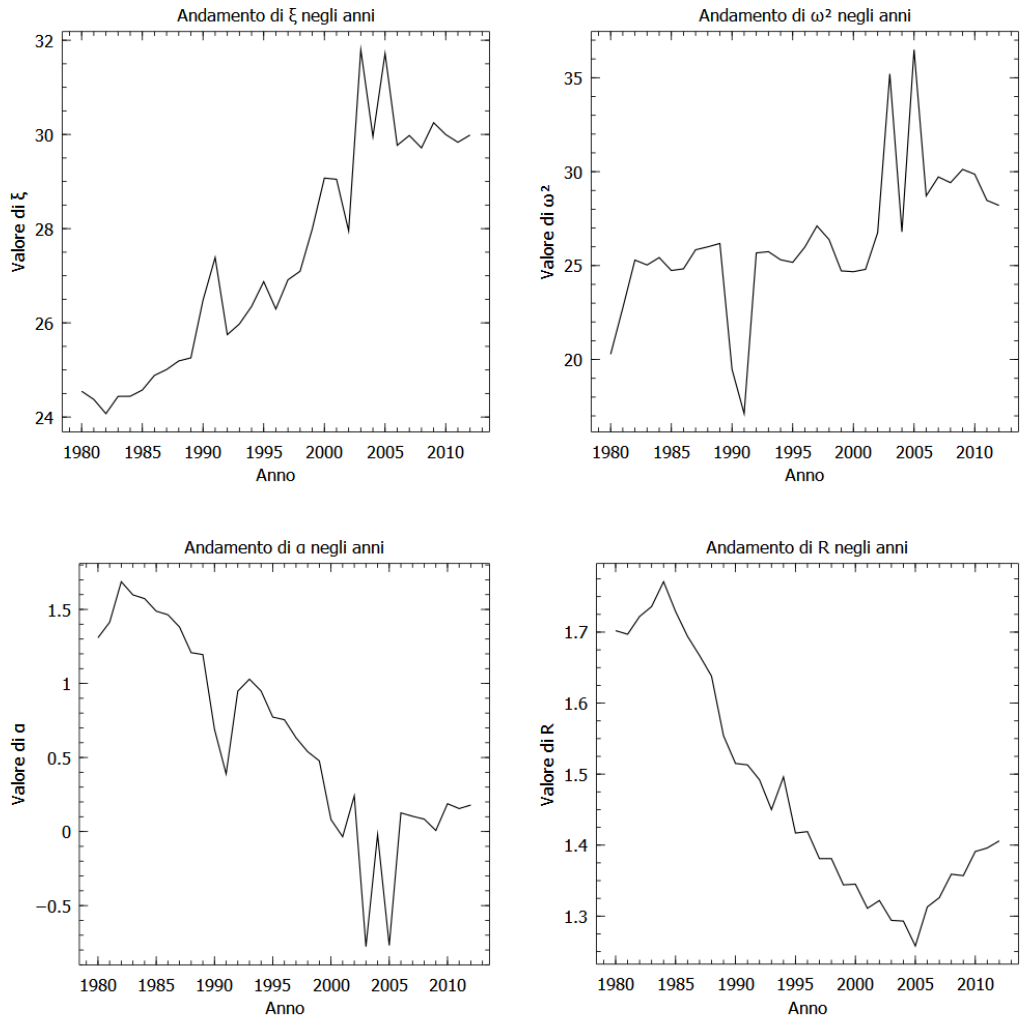

Figura 8: Evoluzione del valore dei parametri dal 1980 al 2012.

8 Conclusioni

In questo lavoro è stato studiato un modello bayesiano per la stima dei parametri della curva di fecondità e, in particolare, l'introduzione del parametro relativo al tasso di fecondità all'interno del modello per fare sì che la stima dell'intero vettore di parametri avvenisse in un unico passo anziché in due. I problemi derivanti da una non perfetta aderenza della curva stimata ai dati osservati potrebbero derivare da imperfezioni del modello base da cui è partito questo lavoro, sui quali potrebbe essere necessario indagare in lavori futuri o, forse, dai sempre possibili bug nel codice, potenzialmente dovuti all'implementazione di varie funzioni, anche complesse e non qui riportate, ancora mancanti nelle librerie di Julia.

In ogni caso questo lavoro ha portato risultati abbastanza soddisfacenti e proseguono un lavoro di più ampia portata nell'identificare un modello ideale per la trattazione dei dati sul tasso di fecondità e offre anche uno spunto interessante per costruire un modello bayesiano su una funzione che non è una densità propria.

Riferimenti bibliografici

Reinaldo B. Arellano-Valle and Adelchi Azzalini. On the unification of families of skew-normal distributions. *Scandinavian Journal of Statistics*, 33(3):561–574, 2006. ISSN 0303-6898. doi: 10.1111/j.1467-9469.2006.00503.x.

A. Azzalini. A class of distributions which includes the normal ones. *Scandinavian Journal of Statistics*, 12(2):p 171–178, 1985. ISSN 03036898. URL `http://www.jstor.org/stable/4615982`.

Adelchi Azzalini and Antonella Capitanio. *The skew-normal and related families*. Cambridge University Press, Cambridge, 2014. ISBN 1107029279.

Antonio Canale and Bruno Scarpa. *Informative Bayesian inference for the skew-normal distribution*. 2013/05/14 edition, 2013.

T. Chandola, David A. Coleman, and R. W. Hiorns. *Recent European fertility patterns*. 1999.

Monica Chiogna. Some results on the scalar skew-normal distribution. *Journal of the Italian Statistical Society*, 7(1):1–13, 1998. ISSN 1121-9130. doi: 10.1007/BF03178918.

Federico Cauduro. Un modello gerarchico bayesiano per la stima della fecondità regionale. *Tesi, Università di Padova*, 2015.

Riccardo Griggio. Tassi di fecondità per età del costa rica. un'analisi bayesiana. *Tesi, Università di Padova*, 2014.

H. Hadwiger. Eine analytische reprodutionsfunktion fur biologische gesamtheiten. *Skandinavisk Aktuarietidskrift*, 23:101–113, 1940.

J. M. Hoem, D. Madsen, J. L. Nielsen, E. Ohlsen, H. O. Hansen, and B. Rennermalm. Experiments in modelling recent danish fertility curves. *Demography*, 18(2):231–244, 1981. doi: 10.2307/2061095.

Jeff Bezanson, Stefan Karpinski, Viral B. Shah, Alan Edelman. Julia, 2012. URL `http://julialang.org/`.

Yanyuan Ma and Marc G. Genton. Flexible class of skew-symmetric distributions. *Scandinavian Journal of Statistics*, 31(3):459–468, 2004. ISSN 0303-6898. doi: 10.1111/j.1467-9469.2004.03_007.x.

Stefano Mazzuco and Bruno Scarpa. Fitting age-specific fertility rates by a flexible generalized skew normal probability density function. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 178(1):187–203, 2015. ISSN 09641998. doi: 10.1111/rssa.12053.

Silvia Peruzzo. Stima dei tassi di fecondità del cile tramite verosimiglianza e applicazione di un algoritmo em. *Tesi, Università di Padova*, 2015.

Carl Schmertmann. A system of model fertility schedules with graphically intuitive parameters. *Demographic Research*, 9:81–110, 2003. ISSN 1435-9871. doi: 10.4054/DemRes.2003.9.5.

Angelica Zenere. Analisi dei tassi di fecondità tramite la distribuzione normale asimmetrica flessibile. *Tesi, Università di Padova*, 2015.

Appendice A - Grafici

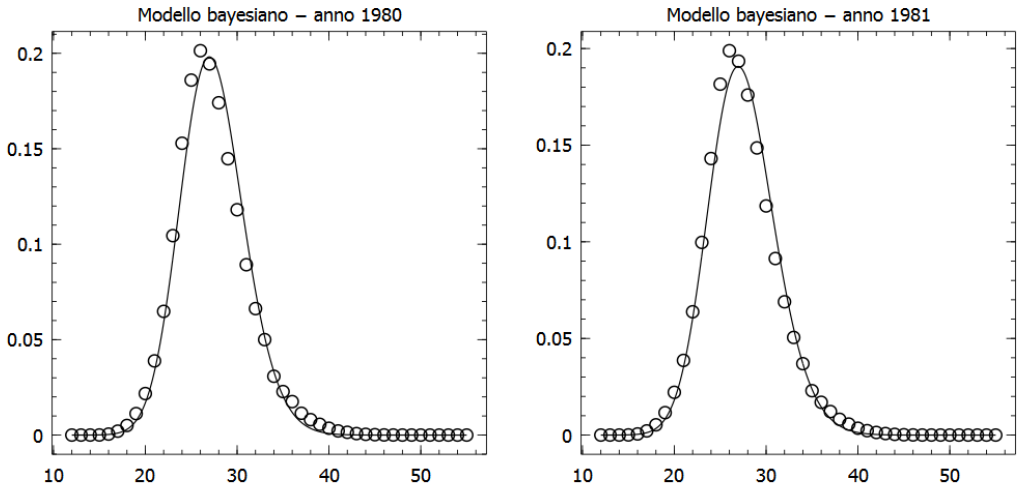

Figura 9: La curva di fecondità stimata per gli anni 1980 e 1981.

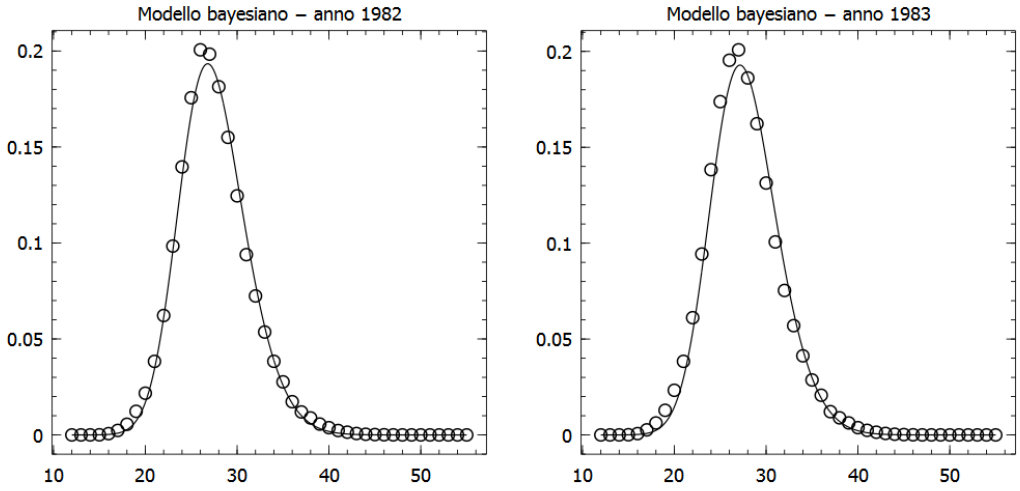

Figura 10: La curva di fecondità stimata per gli anni 1982 e 1983.

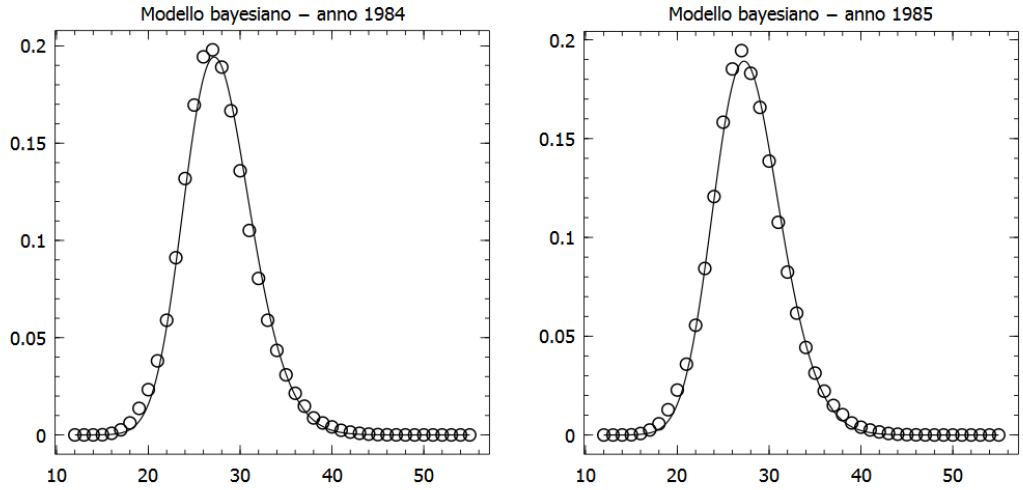

Figura 11: La curva di fecondità stimata per gli anni 1984 e 1985.

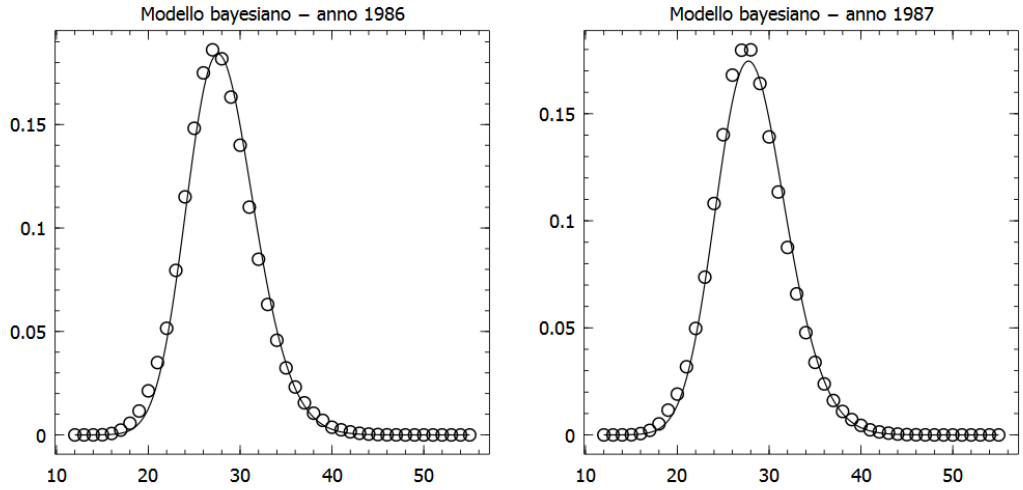

Figura 12: La curva di fecondità stimata per gli anni 1986 e 1987.

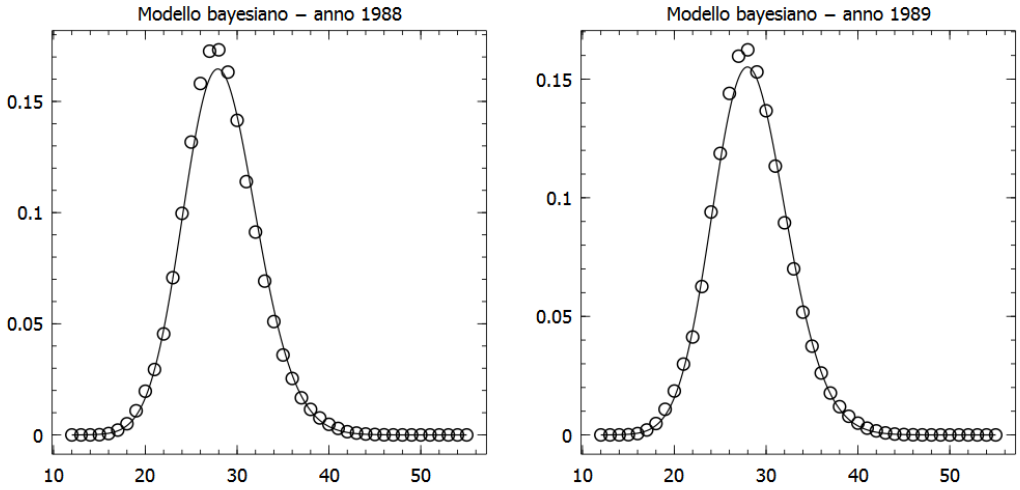

Figura 13: La curva di fecondità stimata per gli anni 1988 e 1989.

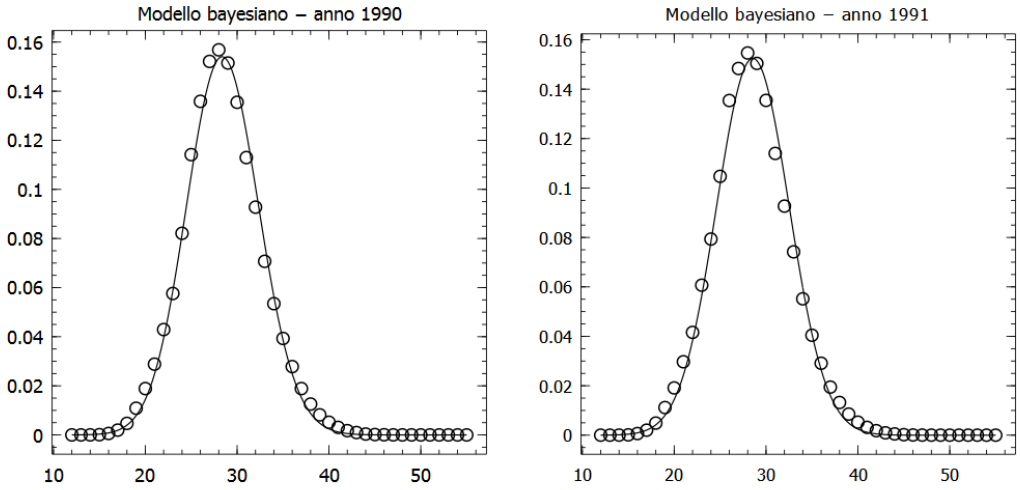

Figura 14: La curva di fecondità stimata per gli anni 1990 e 1991.

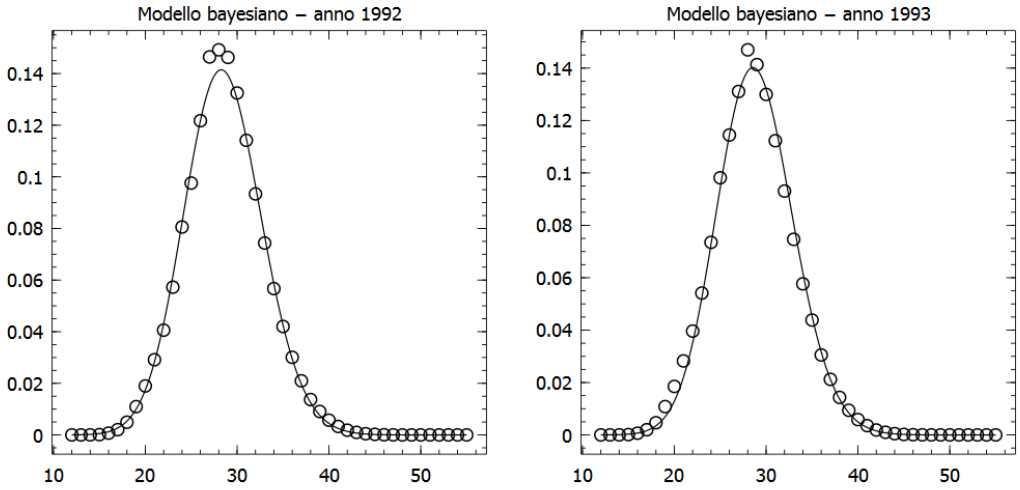

Figura 15: La curva di fecondità stimata per gli anni 1992 e 1993.

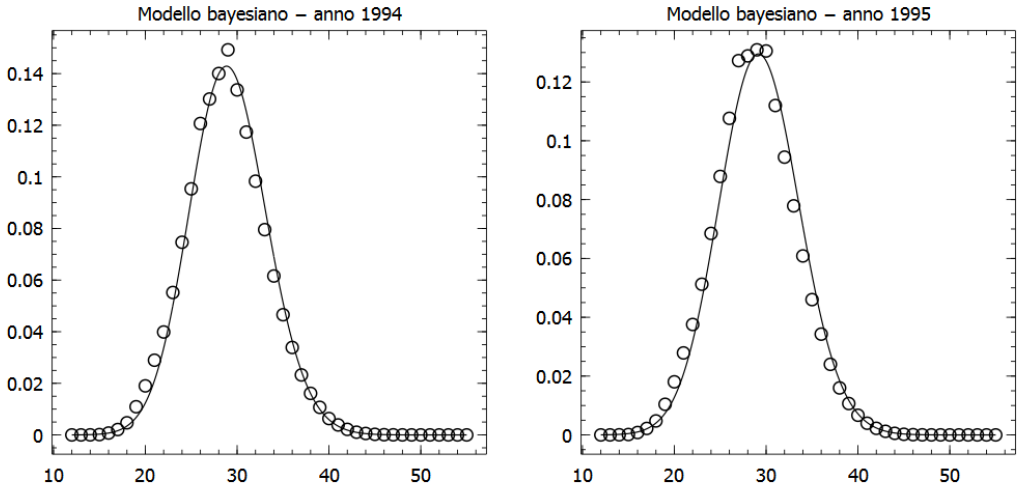

Figura 16: La curva di fecondità stimata per gli anni 1994 e 1995.

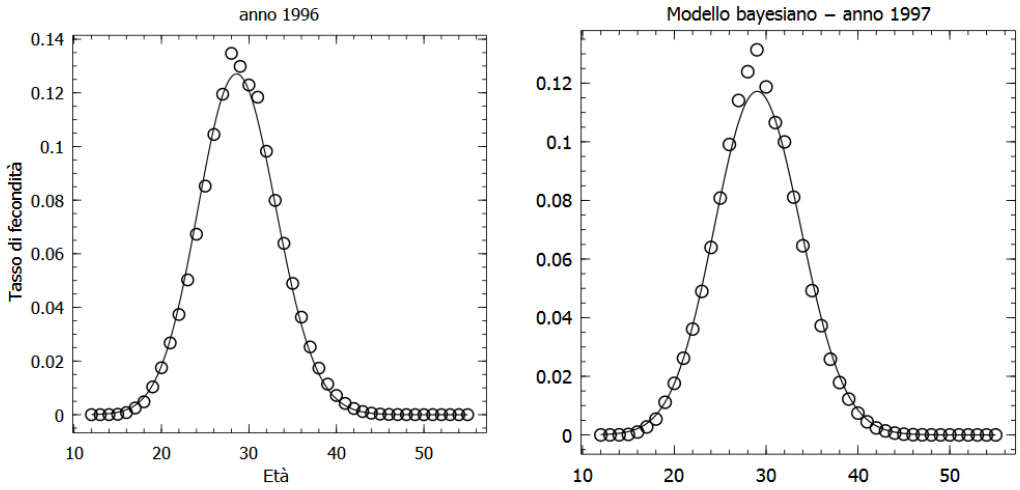

Figura 17: La curva di fecondità stimata per gli anni 1996 e 1997.

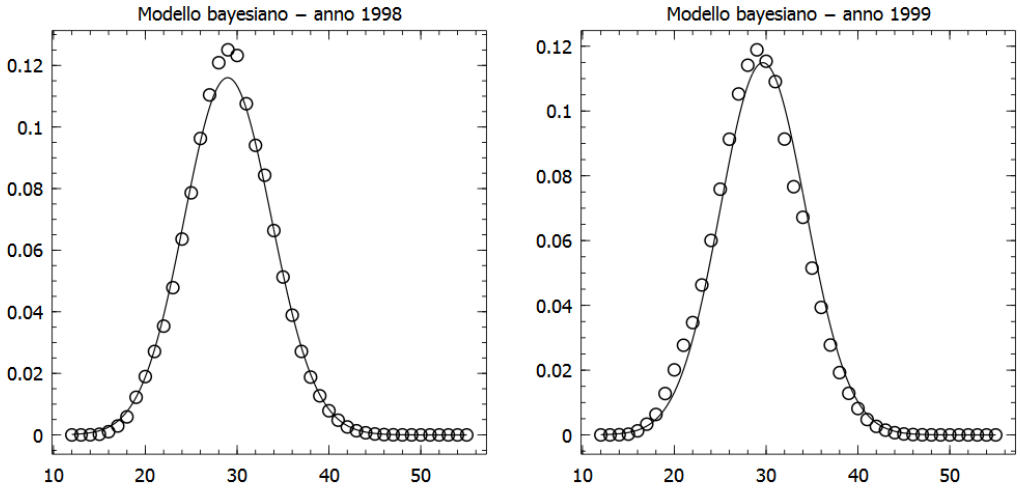

Figura 18: La curva di fecondità stimata per gli anni 1998 e 1999.

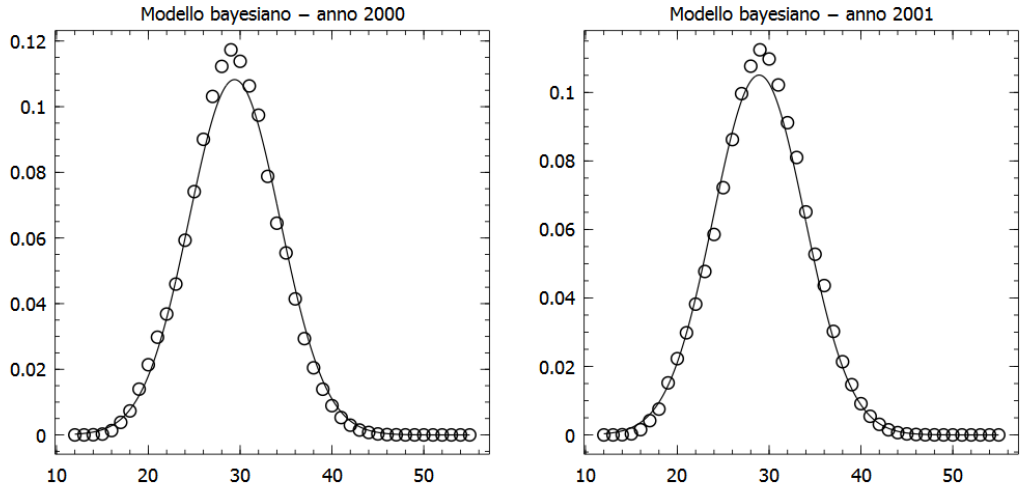

Figura 19: La curva di fecondità stimata per gli anni 2000 e 2001.

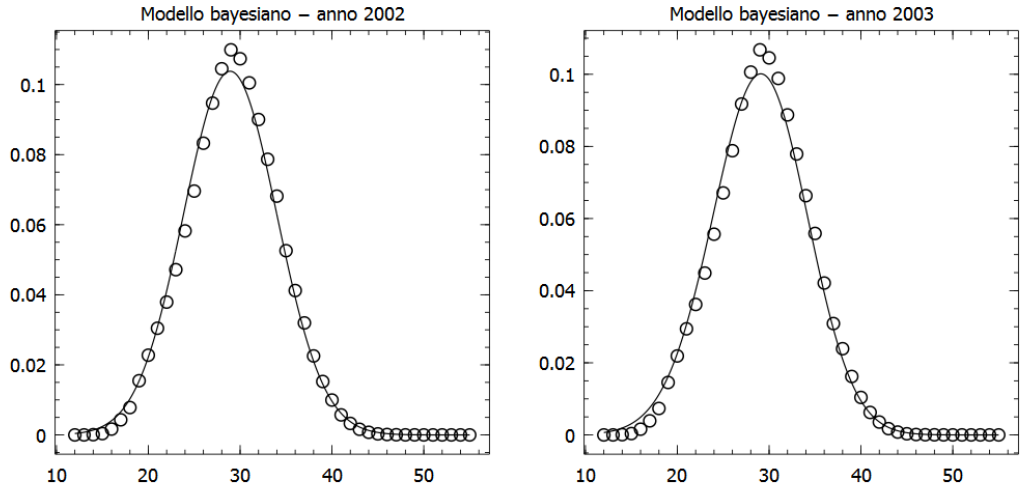

Figura 20: La curva di fecondità stimata per gli anni 2002 e 2003.

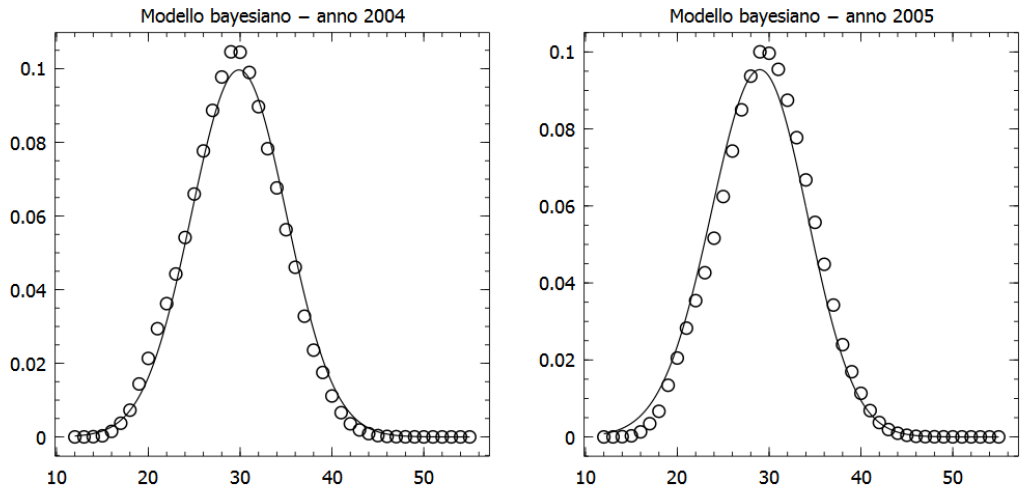

Figura 21: La curva di fecondità stimata per gli anni 2004 e 2005.

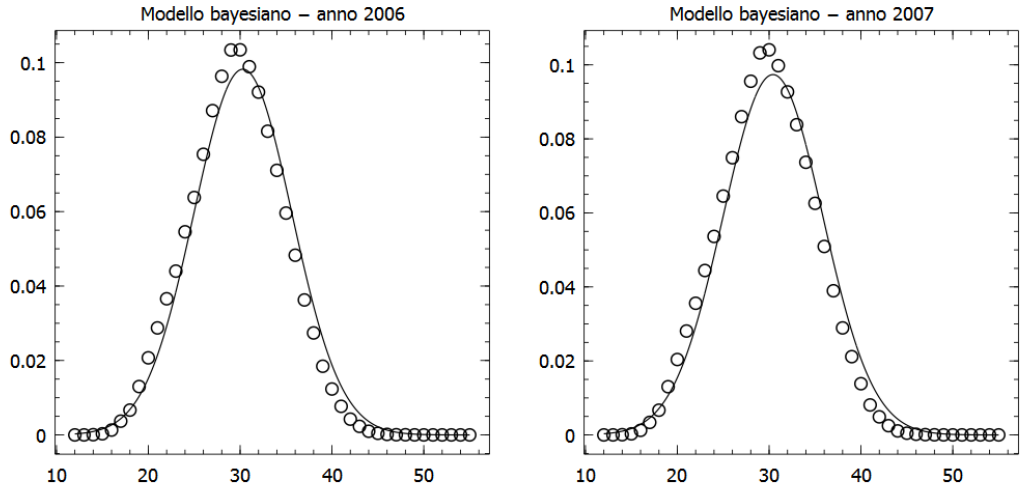

Figura 22: La curva di fecondità stimata per gli anni 2006 e 2007.

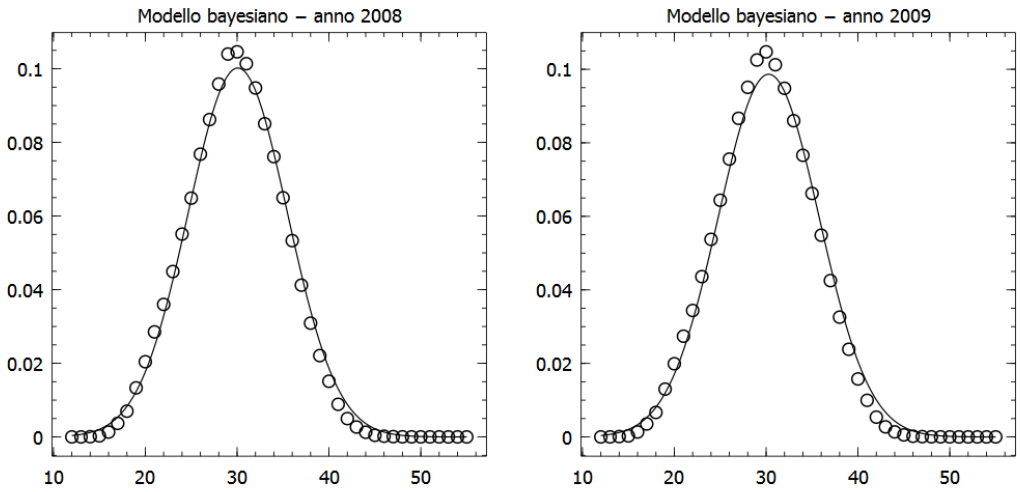

Figura 23: La curva di fecondità stimata per gli anni 2008 e 2009.

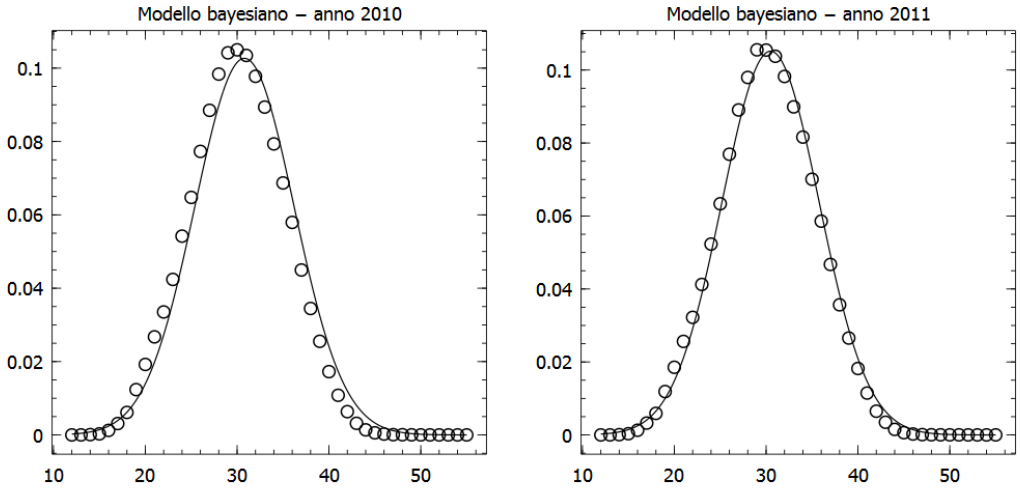

Figura 24: La curva di fecondità stimata per gli anni 2010 e 2011.

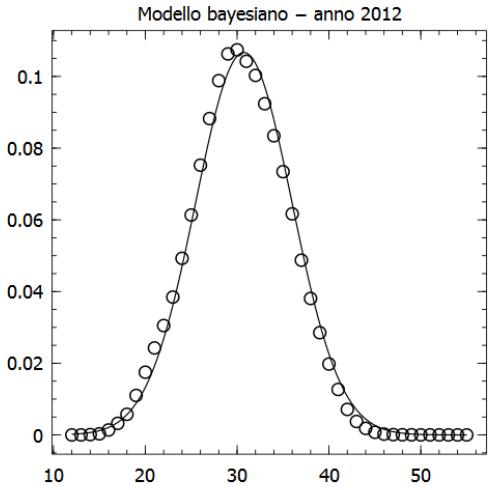

Figura 25: La curva di fecondità stimata per l'anno 2012.

Appendice B - Codice

Codice per il calcolo della distribuzione a posteriori di R.

```
function lprior(R::Float64, a::Float64, b::Float64)
  logpdf(Gamma(a, 1/b), R)
end

function llik(R::Float64, x::Int, B::Int, D::Int,
      ξ::Float64, ω::Float64, α::Float64)
  if R < 0 || ω < 0 || B < 0 || D < 0
    return -Inf
  else
    π = R * quadgk(y -> dsn(y, ξ, ω, α),
                   x - 0.5, x + 0.5)[1]
    return logpdf(Binomial(D, π), B)
  end
end

function llik(R::Float64, x::Array{Int}, B::Array{Int},
      D::Array{Int}, ξ::Float64, ω::Float64, α::Float64)
  out = 0.0
  for i = 1:length(x)
    out += llik(R, x[i], B[i], D[i], ξ, ω, α)
  end
  return out
end

function lposterior(R::Float64, x::Array{Int},
      B::Array{Int}, D::Array{Int}, ξ::Float64,
      ω::Float64, α::Float64,
      a::Float64, b::Float64)
  return llik(R, x, B, D, ξ, ω, α) + lprior(R, a, b)
end

function R_Metropolis(x::Array{Int}, B::Array{Int},
      D::Array{Int}, ξ::Float64, ω::Float64, α::Float64,
      a::Float64, b::Float64, eps::Float64, R0::Float64)
```

```
  R = R0
  Rs = R + rand(Uniform(-eps, eps))
  prob = exp(lposterior(Rs, x, B, D, ξ, ω, α, a, b) -
             lposterior(R, x, B, D, ξ, ω, α, a, b))
  if rand() < prob
    R = Rs
  end
  return R
end
```

Codice per la generazione da una distribuzione Normale asimmetrica troncata.

```
function dsn(x::Float64, ξ::Float64,
      ω::Float64, α::Float64)
  return (2/ω)*pdf(Normal(), (x-ξ)/ω)*
          cdf(Normal(), α*(x-ξ)/ω)
end

function psn(x::Float64, ξ::Float64,
      ω::Float64, α::Float64)
  return quadgk(x -> dsn(x, ξ, ω, α), -Inf, x)[1]
end

function dsn(x::Array{Float64}, ξ::Float64,
      ω::Float64, α::Float64)
  out = zeros(length(x))
  for (i, elem) in enumerate(x)
    out[i] = dsn(elem, ξ, ω, α)
  end
  return out
end

type SimResult
  values::Array{Float64}
  accepted::Float64
end

function fAR(n::Int, ξ::Float64, ω::Float64,
```

```
        α::Float64, j::Int)
    out = Array(Float64, n)
    total = 0
    denomConst = psn(j+0.5, ξ, ω, α)-psn(j-0.5, ξ, ω, α)
    f(x, ξ, ω, α, j) = dsn(x, ξ, ω, α) / denomConst
    b = -optimize(x -> -f(x, ξ, ω, α, j),
                  j-0.5, j+0.5).f_minimum
    for i = 1:n
      xs = rand(Uniform(j-0.5, j+0.5))
      total += 1
      while rand() * b > f(xs, ξ, ω, α, j)
        xs = rand(Uniform(j-0.5, j+0.5))
        total += 1
      end
      out[i] = xs
    end
    return SimResult(out, n/total)
end
```

Codice del Gibbs Sampler.

```
function pgibbs(N::Int, y::Array{Int}, ξ::Float64,
        ω::Float64, α::Float64, R::Float64,
        B_vec::Array{Int}, D_vec::Array{Int},
        eps::Float64, ar::Float64, br::Float64,
        a::Float64, b::Float64, ξ0::Float64,
        κ::Float64, α0::Float64, ϕ0::Float64, λ0::Float64)
    n = length(y)
    η = zeros(n)
    V = zeros(n)
    A = zeros(N)
    B = zeros(N)
    C = zeros(N)
    D = zeros(N)
    W = zeros(n, N)
    Js = 12:55
    for i = 1:N
      R = R_Metropolis(collect(12:55), B_vec, D_vec, ξ,
                       sqrt(ω), α, ar, br, eps, R)
```

```
    for j = 1:n
      V[j] = fAR(1, ξ, sqrt(ω), α, y[j]).values[1]
    end
    δ = α / sqrt(α^2 + 1.0)
    for r in 1:n
      η[r] = rand(TruncatedNormal(δ * (V[r] - ξ),
                          sqrt(ω * (1.0 - δ^2)), 0, Inf))
    end
    δ_const = 1.0 - δ^2
    μ_hat = (κ * sum(V - δ * η) + (δ_const * ξ0)) /
            ((n * κ) + δ_const)
    κ_hat = (κ * δ_const) / ((n * κ) + δ_const)
    b_hat = (1.0 / (2.0 * δ_const)) * ((δ^2 * sum(η.^2)) +
            (2δ * sum(η .* (V - η))) + sum((V - η).^2) +
            ((δ_const / κ) * ((ξ - ξ0)^2)))
    ξ = rand(Normal(μ_hat, sqrt(κ_hat * ω)))
    ω = 1.0 / rand(Gamma(a + ((n + 1.0) / 2.0),
                         1.0 / (b + b_hat)))
    y_star = (V - ξ) / sqrt(ω)
    z = [y_star, λ0 / ϕ0]
    δ_hat = ϕ0 * z ./ sqrt(ϕ0^2 * z.^2 + 1)
    γ = [δ_hat[1:n] * α0 / ϕ0, 0]
    α = rsun(1, α0, γ, ϕ0, δ_hat)[1]
    A[i] = α
    B[i] = ξ
    C[i] = ω
    D[i] = R
    W[1:end, i] = V
  end
  return [B C A D]
end
```