Università degli studi di Milano–Bicocca
Scuola di Economia e Statistica

Corso di Laurea in
Scienze Statistiche ed Economiche

Approccio bayesiano applicato a modelli di regressione Probit

Relatore: Dott. Tommaso Rigon
Correlatore: Dott. Roberto Ascari

Tesi di laurea di:
Tommaso Menghini
Matricola N. 864946

Anno Accademico 2022/2023

Indice

Ringraziamenti

Introduzione

Questa tesi prevede un'introduzione all'inferenza bayesiana, per poi concentrarsi sugli algoritmi MCMC, fondamentali per campionare da distribuzioni a posteriori complesse o non trattabili. In seguito si approfondiscono i due paper Albert & Chib (1993) e Durante (2019), con particolare attenzione alla trattazione della classe di variabili casuali *unified skew-normal*. Infine viene proposta una applicazione ad un dataset disponibile in R. La tesi consiste di quattro capitoli. Nel primo capitolo si approfondiscono i concetti elementari dell'inferenza bayesiana come il Teorema di Bayes e le distribuzioni a priori coniugate, ma ci si sofferma anche sul perchè studiare e scegliere questo approccio ed il suo rapporto con il paradigma classico frequentista. Il secondo capitolo fornisce una panoramica sui metodi MCMC, si inizia con una contestualizzazione teorica delle catene di Markov, per poi approfondire, anche con esempi, i due algoritmi MCMC più noti: l'algoritmo Metropolis-Hastings e il Gibbs sampler. Il terzo capitolo è il cuore di questa tesi. Si parte con l'illustrare il modello di regressione probit nel contesto bayesiano, per poi passare allo studio di Albert & Chib (1993), nel quale si introducono strategie di *data dugmentation* e il Gibbs sampler per la regressione dati binari. Infine si conclude il capitolo con l'approfondimento in dettaglio della distribuzione *unified skew-normal* e dei risultati ottenuti in Durante (2019) per l'inferenza a posteriori e la previsione per un modello di regressione probit collocato nel paradigma bayesiano. Il quarto e ultimo capitolo presenta un'applicazione dei risultati teorici ottenuti nel capitolo precedente. Si utilizza il dataset preso da R, *Pima.tr*, per confrontare in tempo computazionale e *mixing* prodotto l'algoritmo per campionare indipendentemente dalla distribuzione a posteriori dei coefficienti di regressione con gli algoritmi MCMC approfonditi nel secondo capitolo. Per concludere si discute di alcuni spunti su cui poter continuare a lavorare in futuro. Alla tesi è allegata un'appendice in cui è presentato il codice R, utile per una contestualizzazione più pratica di ciò che si è cercato di spiegare nel corso di questo saggio.

Capitolo 1

Introduzione all'inferenza Bayesiana

"Le differenze tra il paradigma Bayesiano e quello frequentista [...] sono numerose e tali da non esaurirsi nel solo confronto di metodologie statistiche: esse prendono origini dai fondamenti filosofici della teoria della conoscenza e, quindi, coinvolgono l'intera Statistica nel suo momento induttivo."
Domenico Piccolo, Statistica

In questo capitolo si introducono i concetti elementari dell'inferenza bayesiana. Si inzia con il presentare il teorema di Bayes, il quale propone un metodo razionale per aggiornare le proprie conoscenze alla luce di nuove informazioni, per poi illustrare il tema della scelta della distribuzione a priori approfondendo l'approccio coniugato. Si continua quindi con una breve disamina sui concetti di soggettività ed oggettività nelle scienze, con particolare attenzione al campo della statistica. Infine si chiude il capitolo facendo rapide considerazioni su limiti ed ulteriori spunti del paradigma bayesiano. Chiaramente si è trattato solo una piccola parte della conoscenza riguardo questo sottocampo della statistica; per approfondimenti ed ulteriori nozioni si consiglia la consultazione di Robert (1991) e Hoff (2009), i quali sono stati il riferimento principale.

1.1 Teorema di Bayes

Il teorema di Bayes ha assunto un ruolo di notevole importanza per l'intrepretazione statistica di cui se ne fa uso. Questa interpretazione è così estesa che ha dato luce all'impostazione della "Statistica bayesiana" contrapposta a quella detta "classica". Al presbitero Thomas Bayes è attribuità la paternità del teorema omonimo. Questo risultato rappresenta un traguardo rivoluzionario della conoscenza statistica.

Teorema 1.1 (di Bayes). Se $A_1, A_2, ..., A_m$ sono eventi che costituiscono una partizione di Ω, allora per qualsiasi evento $E \in \Omega$ la probabilità di A_i dato E è:

$$P(A_i \mid E) = \frac{P(E \mid A_i)P(A_i)}{\sum_{j=1}^{m} P(A_j)P(E \mid A_j)}, \qquad \forall i = 1, 2, ..., m$$

Questa legge interviene nel problema inverso, cioè inverte il ruolo delle cause e delle conseguenze: dati i risultati si determinano le cause più probabili. Si ha quindi un nuovo punto di vista statistico; il parametro di interesse θ incognito ma fisso nell'approccio classico, ora nell'inferenza bayesiana è caratterizzato da incertezza, diventa quindi una variabile casuale su cui esiste una conoscenza preliminare riassunta dalla distribuzione di probabilità a priori π definita sullo spazio parametrico Θ.

L'inferenza su θ è basata sulla distribuzione di θ condizionatamente ai dati X chiamata distribuzione a posteriori e definita come:

$$\pi(\theta|X) = \frac{\pi(X|\theta)\pi(\theta)}{\int_\Theta \pi(X|\theta)\pi(\theta)d\theta}.$$

L'obiettivo dell'inferenza bayesiana è quantificare quanto l'incertezza sulle caratteristiche della popolazione cambi in base alle informazioni di cui si viene in possesso. Si hanno a disposizione la distribuzione a priori $\pi(\theta)$ e la distribuzione del modello $\pi(X|\theta)$ che descrive la convinzione di X come esito dello studio se θ rappresentasse veramente le caratteristiche della popolazione. Una volta che si ottengono i dati X, allora si può aggiornare ciò che si pensa nei riguardi di θ, infatti la distribuzione a posteriori $\pi(\theta|X)$ descrive l'idea che θ rappresenti le vere caratteristiche della popolazione avendo osservato i dati X. Al denominatore della frazione nell'equazione sopra è presente la costante di normalizzazione, che è la distribuzione marginale dei dati X. Quell'integrale spesso è intrattabile, cioè non ha soluzioni analitiche, aldilà dei casi coniugati. É importante, infine, rimarcare che il procedimento sopra descritto non afferma quali debbano essere le nostre convinzioni su θ, ma come esse cambino nel momento in cui si hanno nuove informazioni.

1.2 Scelta della distribuzione a priori

La distribuzione a priori è per definizione una scelta soggettiva, tuttavia non per questo deve essere arbitraria. In generale π deve essere considerata come uno strumento che sintetizza le informazioni e l'incertezza sulle conoscenze a priori

che si hanno a disposizione, che in un contesto scientifico potrebbero rappresentare proprio la conoscenza della comunità scientifica. L'analisi bayesiana, però, può anche essere estesa in situazioni non-informative, dimostrando che l'utilizzo di una distribuzione a priori non introduce necessariamente un condizionamento nel processo statistico.

In questa sezione si approfondiscono le distribuzioni a priori coniugate e alcuni concetti legati ad esse. Per ulteriori approfondimenti si consiglia Robert (1991).

1.2.1 Distribuzione a priori Coniugata

Nelle situazioni in cui le informazioni sul modello sono limitate, vaghe o inaffidabili lo statistico bayesiano è forzato a pensare ad una distribuzione a priori tale per cui l'inferenza a posteriori sia basata per la maggior parte sul modello stesso. Le distribuzioni a priori coniugate possono essere considerate un buon metodo d'approccio per situazioni descritte come sopra, ma non solo. Il loro vantaggio primario è la possibilità di ottenere una distribuzione a posteriori trattabile semplificando il calcolo di momenti e quantità funzionali all'inferenza successiva.

Definizione 1.1 (Famiglia Coniugata). Una famiglia parametrica $\mathcal{F}$ di distribuzioni a priori su Θ è detta coniugata per una verosimiglianza $\pi(X|\theta)$ se, per ogni $\pi \in \mathcal{F}$, anche la distribuzione a posteriori $\pi(X|\theta)$ appartiene a $\mathcal{F}$.

Le motivazioni che spingono all'utilizzo di questo approccio sono diverse. Per prima cosa le informazioni trasmesse dai dati sono limitate, pertanto nell'aggiornamento da distribuzione a priori a quella a posteriori non ci dovrebbe essere un cambiamento radicale della struttura di $\pi(\theta)$, ma solamente un cambiamento dei suoi parametri. Questo aspetto è caratterizzante dell'approccio coniugato in quanto il passaggio da a priori ad a posteriori è ridotto ad un semplice aggiornamento di parametri. Tuttavia è chiaro che la giustificazione più rilevante per l'utilizzo di distribuzioni a priori coniugate è la loro convenienza matematica. Scegliere una distribuzione a priori trattabile comporta che lo sia anche quella a posteriori portando a espressioni in forma chiusa per il calcolo di quantità a posteriori.

1.2.2 Famiglia esponenziale e a priori coniugate

Si approfondisce una specifica famiglia di distribuzioni fortemente correlata con l'approccio coniugato.

Definizione 1.2 (Famiglia esponenziale). Sia μ una misura σ-finita su $\mathcal{X}$, e sia Θ lo spazio parametrico. Siano C e h funzioni, rispettivamente da $\mathcal{X}$ e da Θ a $\mathbb{R}_+$, e siano R e T funzioni da Θ e $\mathcal{X}$ ad $\mathbb{R}^k$. La famiglia di distribuzioni con densità

$$\pi(X|\theta) = C(\theta)h(X)\exp\{R(\theta)T(X)\}$$

è chiamata famiglia esponenziale di dimensione k. Nel caso particolare in cui $\Theta \subset \mathbb{R}^k$, $\mathcal{X} \subset \mathbb{R}^k$, e

$$\pi(X|\theta) = C(\theta)h(X)\exp\{\theta^T X\}$$

allora la famiglia è detta naturale.

La famiglia di distribuzioni esponenziali comprende distribuzioni come la distribuzione binomiale, quella Normale, quella di Posson e altre ancora. Ognuna di queste distribuzioni esponenziali offre un modo flessibile di modellare i dati, consentendo di adattarsi alle caratteristiche specifiche del fenomeno in esame e trovando quindi applicazione in svariati ambiti di studio.

Esempio: Modello binomiale

Si può dimostrare con facilità che la variabile causale $\texttt{Binomiale}(\theta)$ appartiene alla famiglia esponenziale semplicemente prendendo in considerazione la densità di una singola variabile binaria:

$$\begin{aligned}\pi(X|\theta) &= \theta^x(1-\theta)^{1-x} \\ &= (\frac{\theta}{1-\theta})^x(1-\theta) \\ &= e^{\phi x}(1+e^{\phi})^{-1}\end{aligned}$$

con $\phi = \log(\frac{\theta}{1-\theta})$. La distribuzione a priori coniugata per ϕ è quindi $\pi(\phi|n_0, t_0) \propto e^{\phi n_0 t_0}(1+e^{\phi})^{-n_0}$, con t_0 che rappresenta la probabilità a priori che $X = 1$. Questo porta a una distribuzione a priori per θ $\pi(\theta|n_0, t_0) \propto \theta^{n_0 t_0 - 1}(1-\theta)^{n_0(1-t_0)-1}$ che è la distribuzione di una $\texttt{Beta}$ parametrizzata nel modo seguente $\texttt{Beta}(n_0 t_0, n_0(1-t_0))$. Considerando una $\texttt{Beta}(t_0, (1-t_0))$ poco informativa come distribuzione a priori, si dimostra che quella a posteriori sarà $\{\theta|x_1, ..., x_n\} \sim \texttt{Beta}\,(t_0 + \sum y_i, (1-t_0) + \sum(1-y_i))$.

1.3 La scelta bayesiana

L'analisi bayesiana può essere applicata in svariati campi anche e soprattutto grazie ad i metodi computazionali descritti nel capitolo 2. É importante, però, considerare l'analisi statistica bayesiana prima di tutto da una prospettiva teorica e decisionale. In questa sezione ci si concentra, quindi, sulla coerenza interna al paradigma bayesiano anche comparandolo con la teoria classica frequentista. I riferimenti principali sono stati il capitolo 11 di Robert (1991) e Gelman & Hennig (2015).

1.3.1 Oggettività e soggettività

Nella letteratura spesso è stato criticato il passaggio da informazione a priori a distribuzione a priori. Lo sviluppo di una distribuzione a priori si poggia sulla capacità degli individui di rappresentare la loro conoscenza, con i suoi limiti, in termini di probabilità. Questo processo è stato frequentemente contestato, poichè molti studiosi lo hanno definito come soggettivo e arbitrario. É necessario, allora, aprire una parentesi sui concetti di soggettività ed oggettività. É comune pensare che le scienze, tra cui la statistica, debbano essere oggettive, cioè indipendenti da pregiudizi personali. In realtà si può notare come la statistica sia un campo del sapere in cui ci si debba concentrare notevolmente sul giudizio personale. Ad esempio per molti metodi statistici si devono determinare costanti per regolare il processo, oppure quando si deve comparare metodi in studi di simulazione, vi è il bisogno di compiere diverse scelte nei riguardi di processi di generazione dei dati oppure nelle misure utilizzate per confrontare le performance. Non si deve scappare dalla soggettività, la quale può essere, dunque, trasformata in un principio.

L'interpretazione frequentista della probabilità è considerata oggettiva nel senso che alloca probabilità a eventi che si verificano in un mondo oggettivo che esiste indipendentemente dall'osservatore. Gli epigoni del paradigma frequentista considerano l'oggettività di quest'ultimo come una virtù, ma questa proprietà è solo descrittiva, in quanto non implica che le suddette probabilità esistano nel mondo oggettivo, nè che esse siano un oggetto interessante per lo studio scientifico. La statistica Bayesiana, come detto precedentemente, è spesso stata presentata come soggettiva, proprio per la scelta della distribuzione a priori. Questa idea è errata, poichè è ingannevole pensare che la soggettività provenga

solamente dalle conoscenze a priori che si hanno. In effetti le distribuzioni a priori non necessariamente sono più soggettive di altri aspetti di un modello statistico; è quindi uno sbaglio pensare che in una procedura statistica tutta la soggettività derivi dalle scelte a priori compiute. Per approfondire il tema, con note filosofiche che però trovano corrispondenza sia nella teoria matematica che nelle applicazioni pratiche, si consiglia il *discussion paper* Gelman & Hennig (2015).

1.4 Discussioni conclusive e altri riferimenti

Il dibattito che scontra i fedelissimi dell'approccio frequentista contro quelli dell'approccio bayesiano è frequente e stimolante dal punto di vista intellettuale. Oltre al lato filosofico, le possibili applicazioni del metodo bayesiano sono piuttosto ampie ed una vera comprensione dei benefici e dei limiti di quest'ultimo arriva solo con l'accumulo di esperienza. Di frequente l'inferenza bayesiana, nella pratica, richiede calcoli complessi, specialmente per quanto riguarda la costante di normalizzazione. L'integrale al denominatore nella regola di Bayes infatti è spesso intrattabile. L'obiettivo di campionare dalla distribuzione a posteriori può essere comunque raggiunto tramite l'utilizzo di metodi computazionali come quelli Markov Chain Monte Carlo (MCMC).

Capitolo 2

Metodi Markov Chain Monte Carlo

In questo capitolo si discutono i metodi Markov Chain Monte Carlo, una classe di algoritmi utilizzati per campionare da distribuzioni di probabilità complesse. Per comprendere gli algoritmi MCMC nel loro aspetto pratico è essenziale capirne le basi teoriche, pertanto la sezione inizia trattando i concetti essenziali delle catene markoviane, per poi approfondire i due metodi MCMC più illustri: l'algoritmo Metropolis-Hastings e il Gibbs sampler. Per un maggior approfondimento si consiglia la consultazione di Robert & Casella (1999) e Robert & Casella (2010), i quali sono stati il riferimento principale.

Nel contesto bayesiano spesso è difficoltoso campionare direttamente dalla distribuzione a posteriori $\pi(\theta|x)$ rendendo impossibile l'inferenza a posteriori. Gli algoritmi MCMC risolvono questo problema creando una catena di Markov $(\theta^{(r)})_{r\geqslant 1}$, la cui distribuzione stazionaria coincide con la distribuzione a posteriori, da cui poi derivare come approssimazioni le quantità a posteriori ricercate. Nonostante le prime forme di algoritmi MCMC siano venute alla luce negli anni cinquanta, questi ultimi hanno rivoluzionato la statistica bayesiana solo negli ultimi trent'anni permettendo la proposta e lo studio di modelli sempre più complessi.

2.1 Markov Chains

2.1.1 Introduzione

Le catene di Markov sono modelli matematici che descrivono l'evoluzione di un sistema nel tempo in modo probabilistico.

Definizione 2.1 (Markov Chain). Una sequenza di elementi casuali $Y^{(0)}, ..., Y^{(r)}$ è una catena di Markov se

$$P(Y^{(r+1)} \in A|y^{(0)}, ..., y^{(r)}) = P(Y^{(r+1)} \in A|y^{(r)})$$

In altre parole, la distribuzione condizionale di $Y^{(r+1)}$ dati $y^{(0)}, ..., y^{(r)}$ è uguale alla distribuzione condizionale di $Y^{(r+1)}$ dato $y^{(r)}$, chiamata *transition kernel*. Una catena di Markov si dice omogenea se il *transition kernel* non dipende da r. In casi continui quest'ultimo è una densità condizionale indicata con $k(y^{(r+1)}|y^r)$, mentre nei casi discreti è una matrice detta *transition matrix*.

Esempio: AR(1)

I processi autoregressivi sono un esempio comune di catene di Markov su spazi continui. Sia $Y^{(0)} \sim N(\mu, 1)$ e definiamo un AR(1) come:

$$Y^{(r)} = \rho Y^{(r-1)} + \epsilon^{(r)}, \quad \rho \in \mathbb{R},$$

con il termine $\epsilon^{(r)}$ he si distibuisce come una Normale Standard.

La sequenza delle $Y^{(r)}$ forma una catena di Markov e la *transition density* è derivata come:

$$(y^{(r)}|y^{(r-1)}) \sim N(\rho y^{(r-1)}, 1).$$

Definizione 2.2 (Distribuzione di probabilità Invariante). Una densità di probabilità $\pi(y)$ è invariante per una catena di Markov con kernel k se

$$\pi(y^*) = \int k(y^*|y)\pi(y)dy.$$

Questo risultato equivale ad affermare che la distribuzione marginale di $Y^{(r)}$ e $Y^{(r+1)}$ sono le stesse ed uguali a $\pi(y)$, nonostante $Y^{(r)}$ e $Y^{(r+1)}$ siano dipendenti.

Quando una catena di Markov ammette una distribuzione di probabilità stazionaria più altre condizioni tecniche, per un R abbastanza grande la catena si stabilizza attorno alla legge invariante. Non tutte però ne ammettono una, ma le catene costruite per fini simulativi dovrebbero sempre convergere ad una legge stazionaria. Infatti per un algoritmo MCMC la distribuzione di probabilità stazionaria $\pi(y)$ corrisponde alla densità obiettivo dalla quale si vuole campionare al fine di compiere le seguenti approssimazioni

$$\int g(y)\pi(y)dy \approx \frac{1}{R}\sum_{r=1}^{R} g(y^{(R)}),$$

con $y^{(0)}, ..., y^{(R)}$ generati secondo una catena di Markov con $y^{(0)} \sim \pi(y)$

Prima di iniziare a parlare di come costruire algoritmi MCMC è necessario rivedere certe assunzioni sotto le quali le approssimazioni definite come subito sopra sono ragionevoli.

2.1.2 Condizioni di Regolarità

Si considerano catene di Markov irriducibili, aperiodiche e ricorrenti. In questa sezione le proprietà testè introdotte sono presentate in modo parziale, per un ulteriore approfondimento si consiglia il capitolo 4 di Robert & Casella (1999).

Irriducibilità, Aperiodicità e Ricorrenza

Informalmente, una catena si dice irriducibile se non si blocca in una determinata regione dello spazio di campionamento. Questa proprietà è una misura della sensibilità di una catena di Markov alle condizioni iniziali $y^{(0)}$.

Il comportamento di $(Y^{(r)})$ a volte potrebbe essere ristretto da vincoli deterministici nel passaggio da $Y^{(r)}$ a $Y^{(r+1)}$. Una catena di Markov è aperiodica se non presenta un ciclo deterministico. La formalizzazione di tali vincoli deterministici non è ulteriormente trattata, ma si può dimostrare che i metodi MCMC non mostrano questo tipo di comportamento e non soffrono degli svantaggi ad essi collegati.

Una catena di Markov è ricorrente se visita ogni regione dello spazio di campionamento 'sufficientemente spesso'. Nei casi continui, la ricorrenza è definita in termini di numero di passaggi medio in un insieme di Borel, che deve essere divergente.

Misure Invarianti, Detailed Balance Condition e Teorema Ergodico

Una catena di Markov che soddisfa le proprietà di aperiodicità e di ricorrenza mostra un comportamento piuttosto stabile, ma può non ammettere comunque una distribuzione invariante; un esempio può essere il *Random Walk* Gaussiano. Si dice ricorrente positiva una catena che è ricorrente e che ammette una distribuzione di probabilità invariante.

In generale per quanto riguarda l'esistenza di misure invarianti vale il seguente risultato.

Teorema 2.1 Se $(Y^{(r)})_{r\geqslant 1}$ è una catena ricorrente, esiste una misura σ-finita invariante che è unica proporzionalmente ad un fattore moltiplicativo.

Non sempre, però, una misura invariante è una distribuzione di probabilità. Una condizione sufficiente affinchè una catena ricorrente sia anche ricorrente positiva è il risultato del teorema seguente, ma prima è necessario introdurre la definizione di *detailed balance condition*.

Definizione 2.4 (*Detailed Balance Condition*). Una catena di Markov $(Y^{(r)})_{r\geqslant 1}$ con *transition kernel* k soddisfa la detailed balance condition se esiste una funzione f tale per cui

$$k(y|y^*)f(y) = k(y^*|y)f(y^*).$$

Teorema 2.2 Se $(Y^{(r)})_{r\geqslant 1}$ soddisfa la *detailed balance condition* con π, una funzione di densità di probabilità, allora π è la densità stazionaria della catena.

L'ultimo teorema da trattare per quanto riguarda le catene di Markov è quello ergodico. Quest'ultimo è la principale giustificazione dell'utilizzo di metodi MCMC, come la legge dei grandi numeri lo è per l'integrazione Monte Carlo. Il risultato seguente, inoltre, vale a prescindere dalle condizioni iniziali $Y^{(0)}$.

Teorema 2.3 (Teorema Ergodico). Sia la catena di Markov $(Y^{(r)})_{r\geqslant 1}$ ricorrente positiva con una distribuzione stazionaria π. Sia la funzione g integrabile per π. Allora

$$\frac{1}{R}\sum_{r=1}^{R} g(Y^{(R)}) \to \int g(y)\pi(y)dy, \quad R \to \infty,$$

quasi certamente.

2.1.3 Costruzione di una catena di Markov

Avendo presentato la definizione di catena di Markov e , superficialmente, le sue condizioni di regolarità, la sua costruzione è semplice. Si inizia con il simulare $Y^{(0)} \sim \pi_0$, si continua poi simulando i valori seguenti $(Y^{(r+1)}|Y^{(r)})$ in accordo con il *transition kernel* k. Se una catena di Markov ammette una distribuzione stazionaria π, significa che, nella pratica, i valori della catena è come se fossero campionamenti provenienti da quella distribuzione.

Tutte le considerazioni fatte nei paragrafi precedenti comportano che la distribuzione π_r di $Y^{(r)}$ alla fine convergerà alla legge stazionaria π, da cui vogliamo

simulare. Pertanto $Y^{(B)}$, con B grande abbastanza, può essere considerato un campione proveniente da π; lo stesso vale anche per i valori seguenti a questo.

I valori $Y^{(1)}, ..., Y^{(B)}$ rappresentano il periodo di burn-in, cioè il numero di valori di cui la catena ha bisogno per raggiungere la convergenza. Questi ultimi devono essere scartati, ma la scelta di B nella pratica non è scontata. Le approssimazioni di interesse sono quindi calcolate come

$$\int g(y)\pi(y)dy \approx \frac{1}{R-B}\sum_{r=B+1}^{R} g(y^{(r)}).$$

2.2 Metropolis-Hastings

In questo paragrafo s'introduce l'algoritmo Metropolis-Hastings che è considerato come uno degli algoritmi MCMC più generali, ma anche come uno dei più facili da spiegare e comprendere, rendendosi perfetto come punto di partenza per lo studio dei temi trattati in questo capitolo. Il riferimento principale è stato Robert & Casella (2010), che si consiglia di approfondire per uno studio completo dell'argomento.

2.2.1 Algoritmo Metropolis-Hastings

L'algoritmo Metropolis-Hastings è basato sul proporre nuovi valori y campionati da una distribuzione condizionale $q(y^*|y)$ da cui è facile simulare, detta *proposal distribution*. La scelta di q può essere quasi arbitraria; è importante però, per non far fallire l'algoritmo, sceglierne una che riesca a ben esplorare il supporto di $\pi(y)$ che è la densità obiettivo nonchè quella stazionaria. I valori così proposti saranno poi accettati con una certa probabilità α che riflette la verosimiglianza che essi provengano proprio da $\pi(y)$.

Algoritmo 1: Metropolis-Hastings

Inizializzazione: Si pone $y^{(0)}$ come primo valore della catena.
Iterazione r: Sia $y = y^{(r)}$ il valore corrente della catena.

a. Si campiona y^* dalla *proposal distribution* $q(y^*|y)$

b Si calcola la probabilità di accettazione come

$$\alpha = \min\{1, \frac{\pi(y^*)}{\pi(y)}\frac{q(y|y^*)}{q(y^*|y)}\} = \min\{1, \frac{\pi(y^*)\pi(y^*)}{\pi(y)\pi(y)}\frac{q(y|y^*)}{q(y^*|y)}\}$$

c. Si aggiorna il valore corrente della catena con probabilità α.

Un risultato chiave è che per eseguire l'algoritmo non serve conoscere la constante di normalizzazione, dato che essa si semplifica nella frazione sopra. Il *transition kernel* dell'algoritmo MH è il seguente:

$$k(y^*|y) = \alpha(y^*, y)q(y^*|y) + \delta_y(y^*) \int q(s|y)\{1 - \alpha(s|y)\}ds,$$

con $\delta_y(y^*)$ punto di massa che rappresenta la probabilità, nell'iterazione dell'algoritmo di stare fermi. L'integrale sopra, invece corrisponde alla probabilità di rifiutare y^* in media. E' allora semplice dimostrare che la *detailed balance condition* è soddisfatta e che $\pi(y)$ è la legge stazionaria dell'algoritmo. Quest'ultimo è un risultato teorico davvero significativo.

L'algoritmo MH è sempre quasi convergente in teoria, il reale problema è la sua implementazione pratica che potrebbe comportare un tempo di convergenza estremamente lungo o peggio ancora potrebbe dare l'impressione di un'apparente convergenza.

Esempio: Distribuzione Gaussiana

Si vuole simulare da una distribuzione Normale $N(\mu, \sigma^2)$ utilizzando l'algoritmo MH. Per la *proposal distribution* si può utilizzare un *random walk* uniforme, definito come

$$y^* = y + u, \quad u \sim Unif(-\epsilon, \epsilon).$$

La scelta di $\epsilon > 0$ ha un impatto sull'algoritmo. I *random walk* sono delle *proposal distribution* simmetriche, pertanto $q(y^*|y) = q(y|y^*)$. Questa considerazione comporta che α sia uguale a

$$\alpha(y^*, y) = min\{1, \frac{\pi(y^*)}{\pi(y)}\}.$$

Algoritmo 1: Codice R

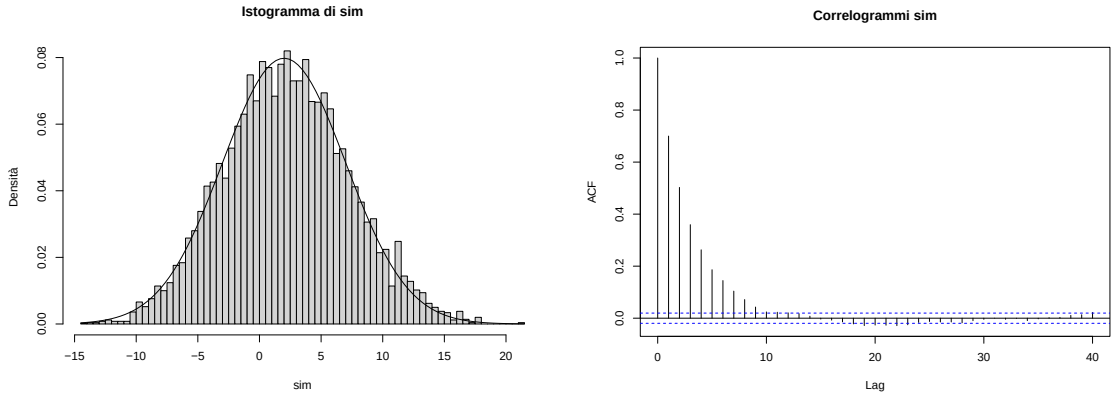
```
norm_mcmc <- function(R, mu, sig, ep, y0, burn_in){

#R  numero di iterazioni, out catena di Markov,
#y0 il primo valore della catena, ep epsilon

  #Inizializzazione
  out <- matrix(0, R, 1)
  y <- y0
  #Algoritmo Metropolis-Hastings
```

```
  for (r in 1:(R+burn_in)){
    #valore proposto
    ys <- y + runif(1, -ep, ep)
    #calcolo dell'acceptance-reject ratio
    alpha <- min(1, dnorm(ys, mu, sig)/dnorm(y, mu, sig))
    #accept-reject step
    accept <- rbinom(1, size = 1, prob = alpha)
    if (accept == 1){
      y <- ys
    }
    if (r > burn_in){
    out[r - burn_in] <- y
    }
  }
  out
}
```

Questo è un esempio giocattolo, poichè nella pratica si userebbe la funzione di R `rnorm`. Considerando una $N(2, 5^2)$ ed usando la *proposal distribution* sopra presentata con $\epsilon = 10$, si esegue il codice sopra presentato ed ottenuta la catena di Markov si verifica se la convergenza è avvenuta attraverso la library `coda` di R. L'obiettivo è quello di simulare da una normale, quindi si verificano i propri risultati sovrapponendo all'istogramma della catena prodotta la curva della funzione di densità di probabilità della Normale che abbiamo preso a riferimento. Come si vede dalla figura 2.1a l'algoritmo MH converge ad una densità stazionaria che coincide con quella obiettivo. Il grafico in figura 2.1b è molto utile in quanto mostra un concetto cardine della catene di Markov. Chiaramente per come è costruita una catena, i valori da essa prodotti non sono indipendenti; essi hanno una certa correlazione fra loro, specialmente quelli vicini in termini di iterazione in cui sono stati simulati. Ciò comporta che non tutte le simulazioni della catena porteranno informazioni, elaborando *mixing* di minor qualità rispetto ad un campionamento iid.



(a) Istogramma vs Curva di una normale **(b)** Autocorrelazione

Figura 2.1

2.2.2 Mh ibrido

I veri vantaggi dei metodi MCMC sui metodi di campionamento classici si osservano quando le dimensioni $Y^{(r)} = (Y_1^{(r)}, ..., Y_p^{(r)})$ diventano elevate. Nonostante ciò quando $p > 2$ pensare ad una *proposal distribution* può non essere scontato. Un'alternativa può essere quella di utilizzare un algoritmo Metropolis-Hastings detto ibrido. L'idea è quella di applicare iterativamente l'Algoritmo 1 per ogni elemento $Y_j^{(r)}$, secondo la *proposal distribution* $q_j(y_j^*|j)$. A volte è anche conveniente considerare blocchi di coordinate piuttosto che solo componenti univariate. Questa strategia computazionale è anche chiamata Metropolis-within-Gibbs.

2.3 Gibbs Sampler

Questo paragrafo approfondisce l'ambito d'applicazione degli algoritmi MCMC studiando una nuova classe di quest'ultimi metodi chiamata Gibbs Sampler. Il grande fascino di questi schemi è sicuramente rappresentato dal loro funzionamento che dipende in maniera particolare dalla densità obiettivo, e dal fatto che riescono a suddividere problemi complessi, come ad esempio densità di dimensioni elevate per le quali l'algoritmo Metropolis-Hasting è praticamente impossibile da applicare, in problemi di semplice risoluzione.

Questo paragrafo ha l'obiettivo di fornire una panoramica generale del Gibbs Sampler, illustrando i suoi fondamenti teorici, i concetti chiave e le sue applicazioni principali. Per un maggior approfondimento si consiglia il capitolo 7 di Robert & Casella (1999) o il capitolo 7 di Robert & Casella (2010); quest'ultimo è stato il riferimento principale.

2.3.1 Full conditional e Gibbs Sampler

Il concetto di Gibbs sampler è strettamente relazionato con quello di *full-conditional*. Si supponga che per $p > 1$ la variabile casuale Y possa essere scritta come $Y = (Y_1, ..., Y_P)$, con gli Y_i che possono essere componenti unidimensionali o multidimensionali. Inoltre si ipotizzi che si possa simulare facilmente dalla densità condizionali $\pi_1, ..., \pi_p$ corrispondenti. In questo caso con la notazione $\pi(Y_i|-)$ qualifichiamo la cosidetta *full conditional* di Y_i, definita come

$$\pi(Y_p|-) = \pi(Y_p|Y_1, ..., Y_{p-1}, Y_{p+1}, ..., Y_P),$$

cioè la distribuzione di Y_p condizionatamente a tutte le altre componenti di Y. Campionando da $\pi(Y_p|-)$ per p da 1 a P si compone il Gibbs sampler che porta infine a simulare da $\pi(Y)$, la densità obiettivo. Pertanto, anche in un problema a dimensioni elevate vi è la possibilità di compiere simulazioni che possono essere univariate; ciò comporta solitamente un vantaggio.

Algoritmo 2: Gibbs Sampler

Inizializzazione: Si pone $y^{(0)}$ come primo valore della catena.

Iterazione r: Sia $y^{(r)} = y_1^{(r)}, ..., y_P^{(r)}$ il valore corrente della catena.

1. $Y_1^{(r+1)} \sim \pi(y_1|y_2^{(r)}, ..., y_P^{(r)})$;
2. $Y_2^{(r+1)} \sim \pi(y_2|y_1^{(r+1)}, y_3^{(r)} ..., y_P^{(r)})$;

P. $Y_P^{(r+1)} \sim \pi(y_P|y_1^{(r+1)}, ..., y_{P-1}^{(r+1)})$;

La distribuzione di $Y^{(r)}$, nel momento in cui $r \to \infty$, si avvicina alla densità obiettivo, indipendentemente dai valori $y^{(0)}$ con cui si è inizializzato algoritmo. Ovviamente valori iniziali migliori comportano un minor tempo computazionale per arrivare a convergenza. Il risultato cardine è che per la maggior parte delle funzioni g di interesse si possono compiere le seguenti approssimazioni,

$$\frac{1}{R} \sum_{r=1}^{R} g(y^{(r)}) \to \mathbb{E}[g(y)] = \int g(y)\pi(y)dy \quad \text{con } R \to \infty.$$

Nonostante possa essere difficoltoso simulare dalla distribuzione di Y, potrebbe essere più facile invece farlo dalle *full conditionals* $\pi(Y_p|-)$. Questa è l'importantissima capacità del Gibbs Sampler, che però comporta una qualche

conoscenza a priori su π, infatti non è sempre verificata la disponibilità di *full conditional* in forma chiusa da cui è facile campionare.

L'applicazione del Gibbs sampler è basata sul Teorema di Hammersley-Clifford; quest'ultimo, infatti, garantisce che le distribuzioni condizionali $\pi(Y_p|-)$ contengano sufficiente informazione per produrre un campione da $\pi(Y)$.

Esempio: Modello Gaussiano

Si assuma che le osservazioni $(y_1, ..., y_n)$ siano simulazioni da

$$(y_i|\mu, \sigma^2) \overset{iid}{\sim} N(\mu, \sigma^2), \;\; i = 1, ..., n.$$

con distribuzioni a priori $\mu \sim N(\mu_\mu, \sigma^2_\mu)$ e $\sigma^{-2} \sim Gamma(a_\sigma, b_\sigma)$ indipendenti.

Si può dimostrare che la distribuzione *full-conditional* per la media μ è:

$$(\mu|-) \sim N\left(\left(\tfrac{n}{\sigma^2} + \tfrac{1}{\sigma^2_\mu}\right)^{-1}\left(\tfrac{\mu_\mu}{\sigma^2_\mu} + \tfrac{1}{\sigma^2}\textstyle\sum_{i=1}^n y_i\right), \;\; \left(\tfrac{n}{\sigma^2} + \tfrac{1}{\sigma^2_\mu}\right)^{-1}\right).$$

Mentre per quanto riguarda la precisione σ^{-2}:

$$\left(\sigma^{-2}|-\right) \sim Gamma\left(a_\sigma + \tfrac{n}{2}, \;\; b_\sigma + \tfrac{1}{2}\textstyle\sum_{i=1}^n (x_i - \mu)\right)^2.$$

Avendo a disposizione le *full conditionals* in forma chiusa, si può impostare un Gibbs Sampler; il codice R per eseguire questo schema di campionamento è il seguente:

Algoritmo 1: Gibbs Sampler

```
gibbs_R <- function(y, mu_mu, sigma_mu, a_sigma,
                    b_sigma, R, burn_in){
  #Inizializzazione
  n <- length(y); xbar <- mean(y)
  out <- matrix(0, M, 2)
  #Valori iniziali per mu e sigma
  sigma <- var(x); mu <- xbar
  for (i in 1:(burn_in + R)){

    #Campiono mu
    sigma_n <- 1/(n/sigma + 1/sigma_mu)
    mu_n <- sigma_n*(mu_mu/sigma_mu + n/sigma*xbar)
    mu <- rnorm(1, mu_n, sqrt(sigma_n))

    #Campiono sigma
    a_n <- a_sigma + n/2
    b_n <- b_sigma + 1/2*sum((x - mu)^2)
```

```
    sigma <- 1/rgamma(1, a_n, b_n)

    #Salvo i risultati dopo il burn-in
    if (i > burn_in){
    out[i - burn_in, ] <- c(mu, sigma)
    }
  }
  out
}
```

Per la verifica della convergenza dell'algoritmo si utilizza, come nell'esempio precendente, il pacchetto `R coda`. La figura 2.2 è l'output della funzione `plot` della libreria, la quale quando è applicata ad un oggetto `mcmc` produce il grafico del percorso della catena nel corso delle varie iterazioni e una stima non parametrica della densità di ogni parametro. In questo caso non si riscontrano segnali di comportamenti non stazionari.

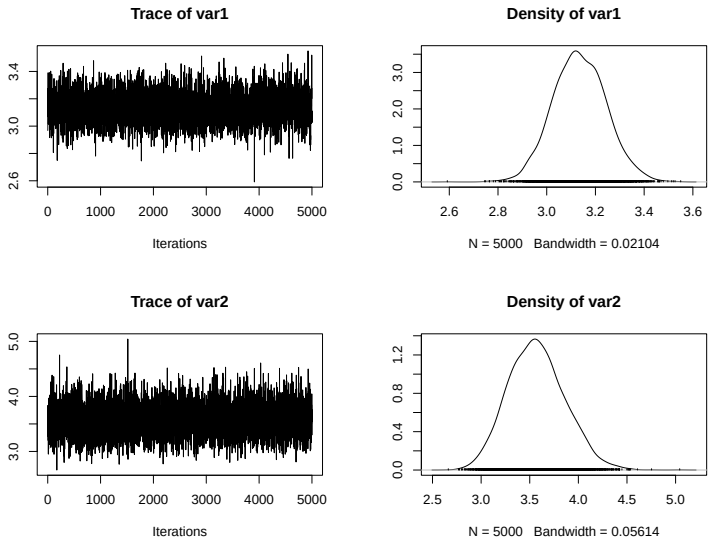


Figura 2.2: Output della funzione `plot` applicata alla catena prodotta dal codice `R` sopra riportato.

2.3.2 Confronto Gibbs Sampler e Metropolis-Hastings

Nonostante il Gibbs sampler sia, almeno formalmente, un caso speciale dell'algoritmo Metropolis-Hastings, questi due schemi computazionali hanno caratteristiche differenti. Prendendo in considerazione, infatti, la definizione di algoritmo Metropolis-within-Gibbs, trattato precedentemente, se si considerano

come *proposal distributions* le *full conditionals* si dimostra che l'*acceptance rate* α è sempre pari a 1 e pertanto ogni valore simulato è accettato. Questo risultato significa che la convergenza per questo algoritmo è da trattare in modo differente rispetto a metodologie *Metropolis-Hastings*, per le quali vi sono molte discussioni sul tasso di accettazione ottimale. Il Gibbs sampler, inoltre è per costruzione multidimensionale e non è possibile applicarlo in problemi in cui il numero dei parametri varia.

Comparato all'algoritmo Metropolis-Hastings, il numero di sue possibili implementazioni è quindi ridotto. Una comparazione più generale dei due metodi è poco significativa; in base al problema in questione e nel modo in cui lo si approccia un algoritmo potrebbe convergere più velocemente di un altro. Il Gibbs sampler non è necessariamente la miglior scelta per l'implementazione di un MCMC, infatti se da una parte quest'ultimo è costruito direttamente dalla distribuzione obiettivo e quindi non comporta alcun tipo di contributo soggettivo, è anche vero che aggiornando la catena una componente alla volta può portare ad una convergenza di scarsa qualità. Al contrario, l'algoritmo Metropolis-Hastings utilizzando *random walks* come *proposal distribution* può essere inefficiente, ma permette salti che sono capaci di raggiungere modalità di π più lontane.

Capitolo 3

Modello di Regressione Probit

In questo capitolo si introduce il modello di regressione probit per risposte binarie, analizzandolo nel contesto bayesiano. Si approfondiscono Albert & Chib (1993), documento molto influente dal punto di vista storico per l'introduzione di strategie di *data augmentation* per modellare risposte binarie con metodi bayesiani, e Durante (2019), articolo recente che dimostra che la distribuzione a posteriori per i coefficienti probit, con densità Gaussiane a priori, è una *unified skew normal*. Questo risultato permette un'inferenza bayesiana efficiente per un'ampia gamma di applicazioni, in particolare in situazioni in cui p è grande e n è piccolo o moderato.

3.1 Regressione Probit

In molti campi è d'interesse capire come la funzione di probabilità di una risposta binaria y_i assuma valori diversi in base ad un insieme di covariate x osservate. Approcci comuni a questo problema assumono che y_i sia una variabile casuale distribuita come una $\mathtt{Bernoulli}$, con θ suo parametro di probabilità che varia in base ad una combinazione lineare dei predittori associati, sotto una mappatura logit o probit. Nel caso di studio si prende in considerazione il link probit. L'approccio frequentista a questo problema è ben radicato e funzionale, ma quello bayesiano con scoperte come quelle fatte da Albert & Chib (1993) e Durante (2019) è saldo e ricco di ulteriori spunti. Rispetto al metodo frequentista, ora i coefficenti di regressione β sono delle variabili casuali con una propria distribuzione a priori. L'obiettivo ultimo nel paradigma bayesiano è quindi quello di fare inferenza a posteriori sui β.

Siano $y = (y_1, ..., y_n)^T$ un vettore di risposte binarie osservate, X la corrispondente matrice del disegno, con l'i-esima riga pari a $x_i = (1, x_{i2}, ..., x_{ip})^T$, per $i = 1, ..., n$. Si consideri quindi il seguente *GLM*:

$$(y_i \mid \theta_i) \overset{int}{\sim} Bern(\theta_i), \;\; \pi_i = \Phi(\eta_i), \;\; \eta_i = x_i^T \beta$$

Allora la funzione di verosimiglianza di un modello di regressione probit è la seguente:

$$\pi(y \mid \beta) = \prod_{i=1}^{n} \Phi(x_i^T \beta)^{y_i} \{1 - \Phi(x_i^T \beta)\}^{1-y_i}$$

Assumendo che $\beta \sim N(b, B)$ si ottiene così la seguente distribuzione a posteriori:

$$\pi(\beta \mid y) = \frac{\pi(\beta) \prod_{i=1}^{n} \Phi(x_i^T \beta)^{y_i} \{1 - \Phi(x_i^T \beta)\}^{1-y_i}}{\int_{\mathbb{R}^d} \pi(\beta) \prod_{i=1}^{n} \Phi(x_i^T \beta)^{y_i} \{1 - \Phi(x_i^T \beta)\}^{1-y_i} d\beta}$$

L'apparente assenza di una distribuzione a posteriori trattabile ha portato allo sviluppo di diverse soluzioni computazioni come algoritmi Metropolis-Hastings o strategie di *data augmentation* che permette l'implementazione di un Gibbs Sampler. In realtà la distribuzione sopra presentata è nota; è stato dimostrato recentemente in Durante (2019) che essa appartiene alla classe delle distribuzioni *unified skew normal*.

3.2 Albert & Chib (1993)

Nella letteratura che si interessa della regressione probit in ambito bayesiano, di certo è stato di peso l'articolo pubblicato nel 1993 dagli studiosi Albert e Chib. In Albert & Chib (1993) vengono sviluppati metodi bayesiani per modelli con risposta binaria poggiandosi ad una strategia di *data augmentation*. Viene mostrato che la regressione probit per risposte binarie ha una connessione con il modello lineare per risposte continue latenti. I valori dei dati latenti possono essere simulati da una normale troncata, e se quest'ultimi sono noti, allora i coefficienti β possono essere ottenuti come estrazioni da una normale. I campionamenti da questa distribuzione a posteriori sono poi utilizzati per simulare nuovi dati latenti, si compie così un'iterazione del processo determinando la costruzione di un Gibbs Sampler. Questo approccio comporta la possibilità di compiere inferenza a posteriori per i modelli di regressione probit; ad oggi sappiamo che in realtà, quest'ultimo è uno schema sub-ottimale.

3.2.1 Data Augmentation e Gibbs Sampler per dati binari

Si introduce un vettore di variabili latenti, cioè variabili non osservate, $Z = (Z_1, ..., Z_n)^T$, con le Z_i che sono distribuite come delle Normali indipendenti $N(x_i^T\beta, 1)$. Inoltre si definisce $y_i = 1$ se $z_i > 0$ e $y_i = 0$ viceversa. Ricordando che le Y_i sono delle Bernoulli indipendenti con $\theta = P(Y_i = 1) = \Phi(x_i^T\beta)$ la verosimiglianza completa o *augmented likelihood* è definita come:

$$\pi(y, z|\beta) = \prod_{i=1}^{n} \phi(z_i \mid x_i^T\beta, 1)\{\mathbb{1}(z_i > 0)\mathbb{1}(y_i = 1) + \mathbb{1}(z_i \leqslant 0)\mathbb{1}(y_i = 0)\}$$

Si dimostra che la funzione di verosimiglianza $\pi(y|\beta)$ è la marginalizzazione rispetto alle z della verosimiglianza completa. Assumendo una distribuzione a priori Gaussiana sui β del tipo $\beta \sim N_k(b, B)$, allora la distribuzione a posteriori congiunta dei β e delle Z, dati le $y = (y_1, ..., y_n)$ è pari a:

$$\begin{aligned} \pi(\beta, z \mid Y) &\propto \pi(y, z)\pi(\beta) = \\ &= \pi(\beta) \prod_{i=1}^{n} \phi(z_i; x_i^T\beta, 1)\{\mathbb{1}(z_i > 0)\mathbb{1}(y_i = 1) + \mathbb{1}(z_i \leqslant 0)\mathbb{1}(y_i = 0)\} \end{aligned}$$

Si noti che questa distribuzione congiunta è complicata poichè difficile da normalizzare e da simularci direttamente. Ma la derivazione della distribuzione marginale a posteriori di β, compiuta con il Gibbs sampler, richiede la conoscenza solo della distribuzione a posteriori dei β condizionatamente alle Z e la distribuzione a posteriori delle Z condizionatamente ai β. Richiede, quindi, la conoscenza delle distribuzioni *full conditionals*; è facile dimostrare che quest'ultime sono disponibili in forma chiusa e che è semplice campionarci. Si dimostra che:

- La distribuzione *full conditional* dei β è coniugata nel caso di una distribuzione a priori Gaussiana ed è data da:

$$(\beta \mid y, z) \sim N_p\left(\Sigma\left(X^T z + B^{-1}b\right),\ \left(X^T X + B^{-1}\right)^{-1}\right). \quad (1)$$

  Questa distribuzione a posteriori *full conditional* è la densità a posteriori usuale per i coefficienti di regressione in un modello lineare normale $Z = X\beta + \epsilon$, con $X = (x_i^T, ..., x_N^T)^T$, $\epsilon \sim N_N(0, I)$ e I matrice identità.

- La distribuzione a posteriori di Z, condizionata ai β ha anch'essa una forma semplice. Le variabili casuali $Z_1, .., Z_N$ sono indipendenti e distribuite come delle normali troncate:

$$(Z_i \mid y, \beta) \sim N(x_i^T\beta, 1) \text{ troncata a sinistra da } 0 \text{ se } y_i = 1$$
$$(Z_i \mid y, \beta) \sim N(x_i^T\beta, 1) \text{ troncata a destra da } 0 \text{ se } y_i = 0. \quad (2)$$

Si osservi come sia computazionalmente semplice simulare da entrambe le *full conditionals* sopra considerate. Al passo (r+1)-esimo della catena, dato il vettore $(\beta^{(r)}, z^{(r)})$, la successiva iterazione del Gibbs sampler produrrà Z e β dalle distribuzioni (2) e (1) rispettivamente. Per il valore d'inizio di $\beta = \beta^{(0)}$ può essere considerata la stima di massima verosimiglianza oppure, alternativamente, la stima dei minimi quadrati, pari a $\hat{\beta} = (X^TX)^{-1}X^Ty$.

Di questa strategia è importante comprendere e rimarcare la facilità della sua implementazione, con riguardo, però, alla sua capacità di convergenza, la quale deve essere sempre verificata anche come si è visto negli esempi del capitolo 2.

3.3 Durante (2019)

La convinzione dell'indisponibilità di una distribuzione a posteriori trattabile per i β in un modello di regressione per dati binari, con a priori Gaussiane ha portato allo sviluppo di certi metodi computazionali. Come illustrato nella sezione precedente uno schema ordinario coinvolge l'utilizzo di strategie di *data augmentation*, che, però, nella pratica producono scarsa convergenza e *mixing* carente, specialmente per datasets sbilanciati. Un'altra soluzione è quella di costruire un algoritmo Metropolis-Hastings, ma anch'esso mostra delle difficoltà, specialmente quando affronta una situazione in cui p è grande. Le strategie di cui si è discusso si dimostrano sub-ottimali comparate al caso in cui la distribuzione a posteriori in questione appartenga ad una classe di variabili casuali nota e trattabile. Infatti se così fosse vero, verrebbero facilitati i calcoli di certe quantità rilevanti per l'inferenza a posteriori, senza l'appoggio a metodi MCMC. É dimostrato che nel caso di modelli di regressione probit che hanno distribuzioni Gaussiane per i coefficienti, la distribuzione a posteriori appartiene alla classe delle distribuzioni *unified skew normal*. Questo è un risultato recente, presentato in Durante (2019), che è il riferimento principale di questa sezione.

3.3.1 Unified Skew Normal

In questo paragrafo si introduce la distribuzione *unified skew normal*, per approfondimenti e maggior contesto si consiglia la consultazione di Durante (2019) e

Azzalini & Capitanio (2014). Questa variabile casuale unifica diverse generalizzazioni della distribuzione *skew-normal* multivariata $z \sim SN_p(\xi, \Omega, \alpha)$, la quale funzione di densità di probabilità è ottenuta modificando quello di una Gaussiana p-variata $N_p(\xi, \Omega)$ con la funzione di ripartizione di una Normale standard valutata in $\alpha^T \omega^{-1}(z - \xi)$, con ω una matrice diagonale $p \times p$ che contiene le radici quadrate degli elementi sulla diagonale di Ω. Questa tecnica permette di introdurre asimmetria in $N_p(\xi, \Omega)$, controllata da $\alpha = (\alpha_1, ..., \alpha_p)^T \in \mathbb{R}^p$. Quando $\alpha = 0_p$ la distribuzione *skew-normal* multivariata coincide con una Normale multivariata $N_p(\xi, \Omega)$. Nella letteratura sono poi state proposte altre estensioni per catturare ulteriori proprietà. Due importanti generalizzazioni sono state quella di aggiungere un nuovo parametro γ, che porta alla distribuzione *extended skew-normal* multivariata, e quella di rendere il meccanismo che governa l'asimmetria multivariato. Queste estensioni conferiscono non solo flessibilità, ma anche proprietà di chiusura per distribuzioni marginali, condizionate e congiunte; nei fatti consentendo la creazione di una classe generale.

Queste considerazioni sono state unificate in una singola rappresentazione, detta *unified skew normal*; pertanto se $z \sim SUN_{p,n}(\xi, \Omega, \Delta, \gamma, \Gamma)$, la sua funzione di densità è

$$\pi(z) = \phi_p(z - \xi; \Omega) \frac{\Phi_n \left\{ \gamma + \Delta^T \bar{\Omega}^{-1} \omega^{-1}(z - \xi); \Gamma - \Delta^T \bar{\Omega}^{-1} \Delta \right\}}{\Phi_n(\gamma; \Gamma)}.$$

- $\phi_p(z - \xi; \Omega)$ rappresenta la densità di una Gaussiana p-variata con media $\xi = (\xi_1, ..., \xi_p)^T$ e matrice di var-cov $p \times p$ $\Omega = \omega \bar{\Omega} \omega$, combinazione di una matrice di correlazione $\bar{\Omega}$ e di una matrice diagonale ω contenente le radici quadrate degli elementi sulla diagonale di Ω;

- Al numeratore della frazione si ha $\Phi_n \left\{ \gamma + \Delta^T \bar{\Omega}^{-1} \omega^{-1}(z - \xi); \Gamma - \Delta^T \bar{\Omega}^{-1} \Delta \right\}$, che è la funzione di ripartizione di una Gaussiana Multivariata $N_n(0_n, \Gamma - \Delta^T \bar{\Omega}^{-1} \Delta\})$ calcolata in $\gamma + \Delta^T \bar{\Omega}^{-1} \omega^{-1}(z - \xi)$;

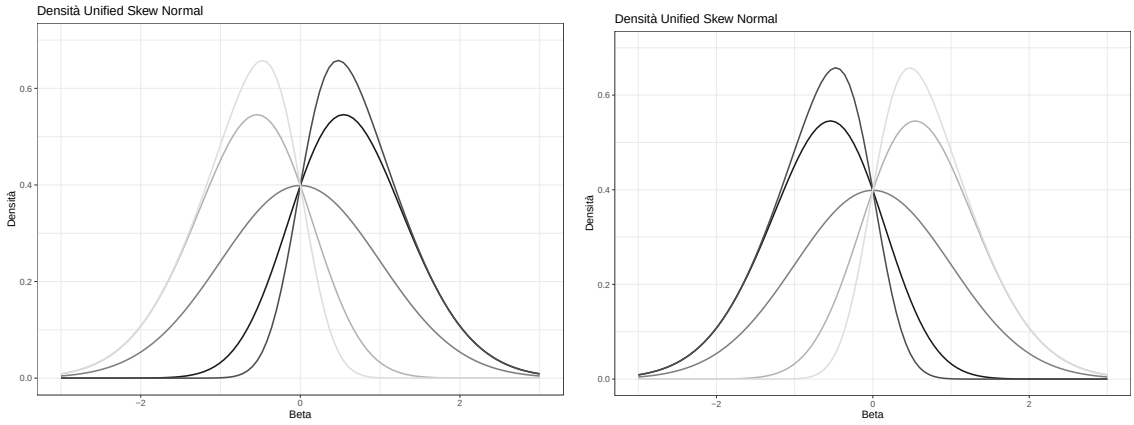
- Al denominatore, invece, si ha $\Phi_n(\gamma; \Gamma)$, cioè la funzione di ripartizione di una Gaussiana multivariata $N_n(0_n, \Gamma)$ calcolata in γ;

- La matrice $p \times n$ Δ ha l'effetto principale sull'asimmetria. Quando Δ è pari a zero la distribuzione della unified-skew normal coincide con una normale di media ξ e varianza Ω $N_p(\xi, \Omega)$;

- Il vettore γ introduce flessibilità aggiuntiva nello scostamento dalla normalità;

- In Arrellano-Valle & Azzalini (2006) viene introdotta una ulteriore condizione che costringe la matrice $(n+p)\times(n+p)$ Ω^*, con blocchi $\Omega^*_{[11]}=\Gamma, \Omega^*_{[22]}=\bar{\Omega}$ e $\Omega^*_{[21]}=\Omega^*_{[12]}=\Delta$, ad essere una matrice a rango pieno. Questo vincolo di identificabilità è, però, non necessario in uno scenario bayesiano, poichè i parametri a posteriori dell' *unified skew normal* per i coefficienti β sono in funzione dei dati osservati e pre-specificati dagli *hyperparameters* della distribuzione a priori Gaussiana;

Per ulteriori approfondimenti sulla natura di queste considerazioni è d'obbligo la consultazione di Arrellano-Valle & Azzalini (2006) e Durante (2019).



(a) **(b)**

Figura 3.1: Funzione di densità di una $SUN_{1,1}\left\{0, 1, (2y-1)x(x^2+1)^{-1/2}, 0, 1\right\}$, distribuzione a posteriori per β, per diverse combinazioni di x e y: (a) $y=1$; (b) $y=0$. Il tono del colore della curva varia da grigio chiaro a grigio scuro secondo $x\in\{-3;-1.5,0;1.5;3\}$ che va da -3 a 3.

Teorema 3.1 Se $y=(y_1,...,y_n)^T$ comprende risposte binarie indipendenti da un modello probit $(y_i \mid x_i,\beta)\sim Ber\{\Phi(x_i^T\beta)\}$ $(i=1,...,n)$ e $\beta\sim N_p(\xi,\Omega)$, allora

$$(\beta \mid y, X)\sim SUN_{p,n}(\xi_{post},\Omega_{post},\Delta_{post},\gamma_{post},\Gamma_{post})$$

con parametri a posteriori

$$\xi_{post}=\xi,\ \ \Omega_{post}=\Omega,\ \ \Delta_{post}=\bar{\Omega}\omega D^T s^{-1},\ \ \gamma_{post}=s^{-1}D\xi,$$
$$\Gamma_{post}=s^{-1}(D\Omega D^T+I_n)s^{-1}$$

per ogni matrice $n\times p$ $D=diag(2y_1-1,...,2y_n-1)X$ e ogni matrice diagonale positiva $n\times n$ $s=diag\left\{(d_1^T\Omega d_1+1)^{1/2},...,(d_n^T\Omega d_n+1)^{1/2}\right\}$; il vettore generico d_i^T è l'i-esima riga di D, X è la matrice del disegno e I_n denota la matrice identita $n\times n$.

Adattando l'espressione generale della funzione di densità di una SUN ai risultati del Teorema 3.1, si ottiene:

$$\pi(\beta|y, X) = \phi_p(\beta - \xi; \Omega)\frac{\Phi_n\left(s^{-1}D\beta; s^{-1}s^{-1}\right)}{\Phi_n\{s^{-1}D\xi; s^{-1}(D\Omega D^T + I_n)s^{-1}\}}$$

Essenziali in Durante (2019) sono i risultati del Corollario 1. Quest'ultimi sono fondamentali sia per la costruzione dell'algoritmo per campionare indipendentemente da una *Unified Skew Normal*, sia per approfondire e comprendere il ruolo di parametri a priori ed effetto dei dati y e X.

Corollario 3.1 Se $(\beta|y, X)$ si distribuisce come l'*unified skew normal* del Teorema 1, allora

$$(\beta|y, X) \stackrel{d}{=} \xi + \omega\{V_0 + \bar{\Omega}\omega D^T(D\Omega D^T + I_n)^{-1}sV_1\} \qquad (V_0 \perp V_1),$$

con $V_0 \sim N_p\{0_p, \bar{\Omega} - \bar{\Omega}\omega D^T(D\Omega D^T + I_n)^{-1}D\omega\bar{\Omega}\}$ e V_1 proveniente da una normale n-variata troncata con media 0_n, matrice di covarianze $s^{-1}(D\Omega D^T + I_n)s^{-1}$ e troncamento a $-s^{-1}D\xi$.

Da questa espressione si possono compiere varie considerazioni. ξ ha un effetto primario sulla *location*, ma gioca anche un ruolo nel controllare la deviazione dall normalità, poichè appare nel troncamento $s^{-1}D\xi$. La matrice di covarianza a priori Ω influisce principalmente sulla dispersione e sulla dipendenza a posteriori dei coefficienti β, ma contribuisce anche al peso assegnato alla Gaussiana multivariata troncata V_1. I dati D si prestano a controllare anch'essi la deviazione dalla normalità, infatti se D ha elementi vicini a 0, allora V_1 ha un'importanza trascurabile comparata a quella della Gaussiana multivariata V_0. Alcune di queste considerazioni trovano corrispondenza nella figura 3.1, che è un esempio illustrativo, in cui si considerano una singola covariata x e una singola risposta y, con $\beta \sim N(0, 1)$. Nei grafici si rappresenta la distribuzione a posteriori di β per valori differenti per y e x. Come ci si aspetta $\Delta_{post} = (2y - 1)x(x^2 + 1)^{-1/2}$ controlla l'asimmetria; maggiore è $|x|$, maggiore sarà l'asimmetria osservata nella distribuzione a posteriori. Quest'ultima sarà positiva o negativa in base al segno di $(2y - 1)x$.

3.3.2 Inferenza e Previsione

Lo studio della regressione nel contesto bayesiano si focalizza principalmente su distribuzioni a posteriori marginali $(\beta_j \mid y, X)$ $(j = 1, ..., p)$ e sul calcolo dei loro momenti associati. Una proprietà fondamentale della distribuzione

unified skew normal, che facilita questo tipo d'inferenza, è che questa classe di variabili casuali è chiusa rispetto alla marginalizzazione, combinazione lineare e condizionamento. Ciò significa che, ad esempio, $(\beta_j \mid y, X) \sim SUN_{1,n}(\xi_{postj}, \Omega_{postjj}, \Delta_{postj}, \gamma_{post}, \Gamma_{post})$, con Δ_{postj} che rappresenta la j-esima riga di $\bar{\Omega}\omega D^T s^{-1}$, ξ_{postj} l'i-esimo elemento del vettore delle media a priori ξ e Ω_{postjj} l'elemento (j, j) della matrice Ω, mentre γ_{post} e Γ_{post} coincidono con la definizione data nel Teorema 1. Tali considerazioni valgono anche per sotto-insiemi di coefficienti, combinazioni lineari di questi e distribuzioni a posteriori condizionali. Questo risultato facilita la rappresentazione grafica delle distribuzioni a posteriori marginali e congiunte, ma semplifica anche il calcolo di momenti a posteriori per i coefficienti della regressioni probit attraverso integrali ad una dimensione di densità a posteriori marginali. Questo può essere compiuto attraverso l'integrazione numerica ogni qual volta è possibile valutare $\Phi_n()$ con efficienza e accuratezza.

Un approccio interessante per ricavare quantità a posteriori è quello di derivarle dalla funzione generatrice dei momenti. Si dimostra che $(\beta \mid y, X)$ è caratterizzato dalla seguente funzione generatrice dei momenti

$$M(t) = \exp(\xi^T t + 0.5 t^T \Omega t) \frac{\Phi_n \left\{ s^{-1} D\xi + s^{-1} D\Omega t; s^{-1}(D\Omega D^T + I_n)s^{-1} \right\}}{\Phi_n \{ s^{-1} D\xi; s^{-1}(D\Omega D^T + I_n)s^{-1} \}}.$$

Si deriva, utilizzando l'espressione sopra presentata, così la media a posteriori di β

$$E(\beta|y, X) = \xi + \frac{1}{\Phi_n(s^{-1} D\xi; s^{-1}(D\Omega D^T + I_n)s^{-1})} \Omega D^T s^{-1} \eta$$

con η vettore la cui i-esima componente è

$$\phi(\bar{\gamma}_i)\Phi_{n-1}(\bar{\gamma}_{-i} - \bar{\Gamma}_{-i}\bar{\gamma i}; \bar{\Gamma}_{-i,-i} - \bar{\Gamma}_{-i}\bar{\Gamma}_{-i}^T)$$

con $\bar{\gamma i}$ che è l'i-esimo elemento di $s^{-1} D\xi = \gamma_{post}$, $\bar{\gamma}_{-i}$ il vettore $(n-1) \times 1$ ottenuto rimuovendo l'i-esima riga in γ_{post}. Similarmente $\bar{\Gamma}_{-i,-i}$ definisce la sottomatrice che si ottiene rimuovendo la i-esima riga e l'esima colonna da $\Gamma_{post} = s^{-1}(D\Omega D^T + I_n)s^{-1}$, mentre $\bar{\Gamma}_{-i}$ è l'i-esima colonna di Γ_{post} con l'elemento della riga i-esima rimosso. Utilizzare questa metodo per calcolare la media a posteriori di β è più efficiente che utilizzare l'integrazione numerica, poichè richiede il calcolo di sole $n + 1$ funzioni di ripartizione, molte meno stime di $\Phi_n()$ necessarie per l'integrazione numerica delle distribuzioni a posteriori marginali. Nonostante questo, ottenere espressioni come momenti marginali di ordine alto

o momenti congiunti, dalla funzione generatrice dei momenti richiede calcoli difficoltosi e pesanti, motivando quindi l'utilizzo di metodi Monte Carlo basati su campioni della distribuzione a posteriori.

Per quanto riguarda i modelli di regressione probit, non è solo importante riuscire a fare inferenza sui coefficenti β, ma è anche d'interesse la previsione di una futura risposta $y_{new} \in \{0; 1\}$ date le covariate associate $x_{new} \in \mathbb{R}^p$ e i dati y, X già noti. Nel paradigma bayesiano, per adempiere a questo compito è necessaria la derivazione della *posterior predictive distribution*, la quale nel caso in questione è definita come

$$P(y_{new} = 1 \mid y, X, x_{new}) = \int \Phi(x_{new}^T \beta)\pi(\beta \mid y, X)d\beta.$$

Corollario 3.2 Se $(y_i \mid x_i, \beta) \sim Ber\{\Phi((x_i^T\beta)\}$ $(i = 1, ..., n)$ e $\beta \sim N_p(\xi, \Omega)$, allora

$$P(y_{new} = 1 \mid y, X, x_{new}) = \frac{\Phi_{n+1}(s_{new}^{-1} D_{new}\xi; s_{new}^{-1}(D_{new}\Omega D_{new}^T + I_n)s_{new}^{-1})}{\Phi_n(s^{-1}D\xi; s^{-1}(D\Omega D^T + I_n)s^{-1})},$$

con D_{new} matrice $(n+1) \times p$ ottenuta aggiungendo come ultima riga a D $d_{new}^T = x_{new}^T$ e $s_{new} = diag\{(d_1^T\Omega d_1 + 1)^{1/2}, ..., (d_n^T\Omega d_n + 1)^{1/2}, (d_{new}^T\Omega d_{new} + 1)^{1/2}\}$.

Un vantaggio di quest'espressione rispetto all'utilizzo di tecniche MCMC è che la previsione non richiede l'integrazione Monte Carlo per $\int \Phi(x_{new}^T\beta)\pi(\beta \mid y, X)d\beta$ tramite campionamento di β dalla distribuzione a posteriori, facendo sì che p non gravi nel peso del fardello computazionale. Questo risultato, quindi, si rivela molto efficace in situazioni in cui p è grande e n è piccolo o moderato.

3.3.3 Procedura di Campionamento

Per ottenere un campionamento indipendente dalla distribuzione a posteriori del Teorema 1, in Durante (2019) si propone l'algoritmo seguente.

Algoritmo 3: Schema per estrarre campioni indipendenti da una SUN.

Per r da 1 a R

1. Si campioni $V_0^{(r)}$ da $N_p\{0_p, \bar{\Omega} - \bar{\Omega}\omega D^T(D\Omega D^T + I_n)^{-1}D\omega\bar{\Omega}\}$;
2. Si campioni $V_1^{(r)}$ da una normale n-variata troncata con media 0_n, matrice di covarianze $s^{-1}(D\Omega D^T + I_n)s^{-1}$ e troncamento a $-s^{-1}D\xi$, usando l'algoritmo *accept-reject* di Botev (2017);
3. Si calcoli $\beta^{(r)}$ come $\beta = \xi + \omega\{V_0^{(r)} + \bar{\Omega}\omega D^T(D\Omega D^T + I_n)^{-1}sV_1^{(r)}\}$;

Output: $\beta^{(1)}, ..., \beta^{(R)}$

Si combina la rappresentazione stocastica di $(\beta \mid y, X)$ risultato del Corollario 1, con lo schema proposto da Botev (2017) per ottenere un campionamento indipendente da una Gaussiana multivariata troncata. Per come quest'ultimo è costruito è molto efficiente in situazioni in cui p è grande ed n è piccolo o moderato. Quando n cresce e p descresce campionare da una Gaussiana multivariata troncata diventa computazionalmente sempre più pesante, favorendo altre strategie come quelle MCMC.

Capitolo 4

Studio Empirico

Quest'ultimo capitolo applica ad un dataset preso da R gli algoritmi che si sono trattati in maniera teorica nel corso di questo studio.

4.1 Pima.tr dataset

Il pacchetto di *R*, *MASS*, contiene il dataset Pima.tr; maggiori informazioni possono essere consultate a questo link RDocumentation (2023). *Pima.tr* contiene 200 soggetti selezionati casualmente dal dataset originario, i cui dati sono stati raccolti, secondo le linee guida dell'Organizzazione Mondiale della Sanità, dal *US National Institute of Diabetes and Digestive and Kidney Diseases*. Sono state fatte diverse misurazioni di tipo medico ad una popolazione di donne di almeno 21 anni con retaggio indiano appartenente alla tribù Pima per la ricerca del diabete. Il dataset consiste in sette covariate ed una varibile risposta, binaria, che indica la presenza, o assenza, di diabete nel soggetto. Le variabili esplicative sono:

- *Pregnancies*: Indica il numero di gravidanze di un soggetto;
- *Glucose*: Indica la concentrazione di glucosio nel plasma del soggetto;
- *Blood Pressure*: Indica la pressione arteriosa diastolica (mm Hg);
- *Skin Thickness*: Indica lo spessore della piega cutanea del tricipite (mm);
- *Insulin*: Indica il livello dell'insulina;
- *BMI*: Indice di massa corporea; è il rapporto tra peso e altezza del soggetto $(\mathrm{kg}/\mathrm{m})^2$;
- *Diabetes Pedigree Function*: Indica la predisposizione del soggetto al diabete tenendo conto della sua età e del trascorso della malattia nella sua famiglia;

- *Age*: Indica l'età del soggetto;

Chiaramente, quello utilizzato in questo caso, è un dataset di benchmark, poichè ampiamente conosciuto nella letteratura e facile da utilizzare. Premesso ciò, un obiettivo in applicazioni simili su dataset reali può essere quello di quantificare gli effetti dei vari predittori di origine medica nella probabilità di possedere una certa malattia, come nel nostro esempio il diabete, oppure quello di predirne la comparsa su un soggetto al momento non malato. Per lo scopo di valutare la performance predittiva, si è diviso il dataset in training set e test set, corrispondenti rispettivamente a 160 osservazioni il primo, pari all'80% del dataset completo, e 40 osservazioni il secondo.

4.2 Metodi Utilizzati

In questa sezione si approfondiscono certi aspetti metodi computazionali utilizzati per campionare dalla distribuzione a posteriori presentata nel paragrafo 3.1.

Le condizioni a priori sono le stesse per tutti e tre gli algoritmi. Si è scelta una distribuzione a priori poco informativa per i coefficienti della regressione, $\beta \sim N(0, 16 \times I_p)$ in linea con Durante (2019), il quale a sua volta cita Gelman et al. (2008), Botev (2017) e Chopin & Ridgway (2017). Il primo articolo illustra le linee guida per compiere decisioni di questo tipo, ed è consigliato per una visione più completa dell'argomento, mentre gli altri due articoli contengono implementazioni simili. Sempre in accordo con questi lavori le covariate del dataset sono state standardizzate affichè abbiano media zero e deviazione standard pari a 0.5. Gli algoritmi MCMC sono stati fatti iterare per 20000 volte, dopo un periodo di burn-in di 5000 iterazioni. Il metodo di campionamento proposto (si guardi la sezione 3.3.3) fornisce campioni indipendenti ed identicamente distribuiti dalla distribuzione a posteriori, pertanto non richiede alcun tipo di controllo per la convergenza o periodo di burn-in.

Il primo algoritmo che si considera è il Metropolis-Hasings. La sua presentazione teorica è già stata trattata nella sezione 2.2. Per eseguire l'algoritmo sono necessarie due scelte. La prima è quella della *proposal distribution*; in questo caso si è optato per una Normale multivariata. La seconda decisione da compiere è la scelta della matrice di covarianze della *proposal distribution*. Fare una scelta sbagliata, sebbene secondo la teoria la convergenza sia assicurata, può portare alla creazione di una catena altamente correlata con poca informazione disponibile,

producendo un *mixing* scarso con problemi nell'inferenza e nella previsione a posteriori. Nel caso preso in esame non è oggetto di studio questa particolare scelta e per ulteriori approfondimenti si consiglia la consultazione di Chopin & Ridgway (2017). La matrice di covarianze asintoticamente ottimale è pari a $2.38^2\Sigma/p$. Con Σ ci si riferisce alla matrice di covarianze della distribuzione a posteriori. Nel caso della regressione probit può essere ricavata facilmente tramite la funzione `R`, `vcov()`. Una precisazione da fare per quanto riguarda il secondo algoritmo considerato, il Gibbs sampler, è da fare sulla scelta del valore d'inizio per quanto riguarda i β. É stato scelto come *starting value* la stima dei minimi quadrati $\beta^{(0)} = (X^TX)^{-1}X^Ty$, in linea con Albert & Chib (1993). Bisogna, però, anche far notare che la teoria del Gibbs sampler implica che quest'ultimo sia automatico, nel senso che non necessita di alcun preciso valore d'inizio per arrivare alla convergenza.

β	Unified Skew-Normal sampler	Gibbs Sampler	MH sampler	Approccio frequentista	Inferenza esatta
const	−0.7351	−0.7372	−0.7292	−0.7165	−0.2155
npreg	0.5714	0.5695	0.5515	0.5542	0.4129
glu	1.1992	1.1985	1.2035	1.1465	1.2548
bp	−0.3558	−0.3614	−0.3743	−0.3264	−0.2330
skin	0.3739	0.3866	0.3605	0.3436	0.5807
bmi	0.4853	0.4783	0.4913	0.4716	0.4455
ped	0.6538	0.6551	0.6499	0.6233	0.8560
age	0.6518	0.6564	0.6554	0.6269	0.70535

Tabella 4.1: Stime dei coefficenti β secondo i tre schemi di campionamento analizzati, secondo l'approccio frequentista al problema, e le stime di questi utilizzando l'espressione della media a posteriori derivata dalla funzione generatrice dei momenti.

La tabella 4.1 rappresenta la stima dei coefficienti β secondo i vari metodi trattati. Si può notare che vi è affinità tra i risultati ottenuti da tutti e tre i metodi di campionamento approfonditi. Da questo risultato possiamo quindi inferire che il *mixing* ottenuto dai tre algoritmi di simulazione sia conforme e buono. Di certo è interessante paragonare queste stime con quelle ottenute con l'approccio frequentista al problema e con quelle ottenute con l'inferenza esatta. Le stime frequentiste sono in linea con quelle prodotte dai tre algoritmi di campionamento. Invece, per quanto riguarda le stime ottenute con l'espressione del valore atteso, calcolato dalla funzione generatrice dei momenti, si nota un certo scarto. Ciò potrebbe essere dovuto al modo con cui si è calcolato il valore atteso dei coefficienti utilizzando l'espressione descritta in 3.3.2. Si è utilizzato, infatti, una funzione `R`, `mvNcdf()` per la computazione della funzione

di ripartizione di una normale multivariata, la quale può aver prodotto una stima non ottimale di essa.

4.3 Risultati e Futuri spunti

É necessario, prima di argomentare le conclusioni conseguite, contestualizzare i concetti di *mixing* ed *effective sample size*. L'ESS rappresenta il numero di campioni indipendenti ed efficacemente informativi contenuti in un campione di dimensione finita. Poichè le simulazioni generate da algoritmi MCMC sono dipendenti e quindi correlate fra loro, quest'ultimo produce una stima del numero di campioni indipendenti che forniscono informazioni significative sulle proprietà della distribuzione della variabile target, che nel caso di studio è la distribuzione a posteriori dei coefficienti β. L'autocorrelazione precedentemente citata tra valori della catena può far sì che quest'ultima sia troppo lenta e che finisca per trascorrere troppo tempo in una determinata regione dello spazio dei parametri, producendo un'immagine distorta della distribuzione a posteriori. Nei casi più gravi alcune parti della distribuzione target potrebbero essere addirittura mancate. Ci si riferisce con *mixing* scarso ad una situazione di quel tipo. Proprio l'ESS è un modo per misurare la qualità del *mixing* di una catena.

	Campioni di β per secondo	Minimo	Primo quartile	Mediana
Unified Skew-Normal sampler	6.2425	20000.00	20000.00	20000.00
Gibbs Sampler	330.1880	4047.5233	4424.8640	4541.8863
MH sampler	3530.0090	932.3071	968.7088	996.7356

Tabella 4.2: Efficienza computazionale: numero di campioni al secondo da $(\beta|y, x)$ e statistiche che riassumono gli ESS delle catene prodotte per i coefficienti β per ogni metodo di campionamento analizzato.

La tabella 4.2 confronta i tre approcci in tempo computazionale e qualità del mixing prodotto. Si può notare come i due algoritmi MCMC siano considerevolmente più veloci nel campionare rispetto al metodo proposto da Durante (2019). Questo è un risultato preventivato, essendo specificato, come base del lavoro di Durante, quello di provare a creare un algoritmo che funzioni bene in situazioni in cui gli algoritmi MCMC funzionano male, cioè in contesti di p-grande ed n-piccolo o moderato. Se il Metropolis-Hastings, però, si dimostra l'algoritmo più rapido è anche quello con *mixing* peggiore. Questa considerazione può essere fatta sia in luce ai valori delle statistiche degli ESS in tabella 4.2, sia in luce al grafico 4.1, nel quale si nota come il boxplot riferito a questo metodo di

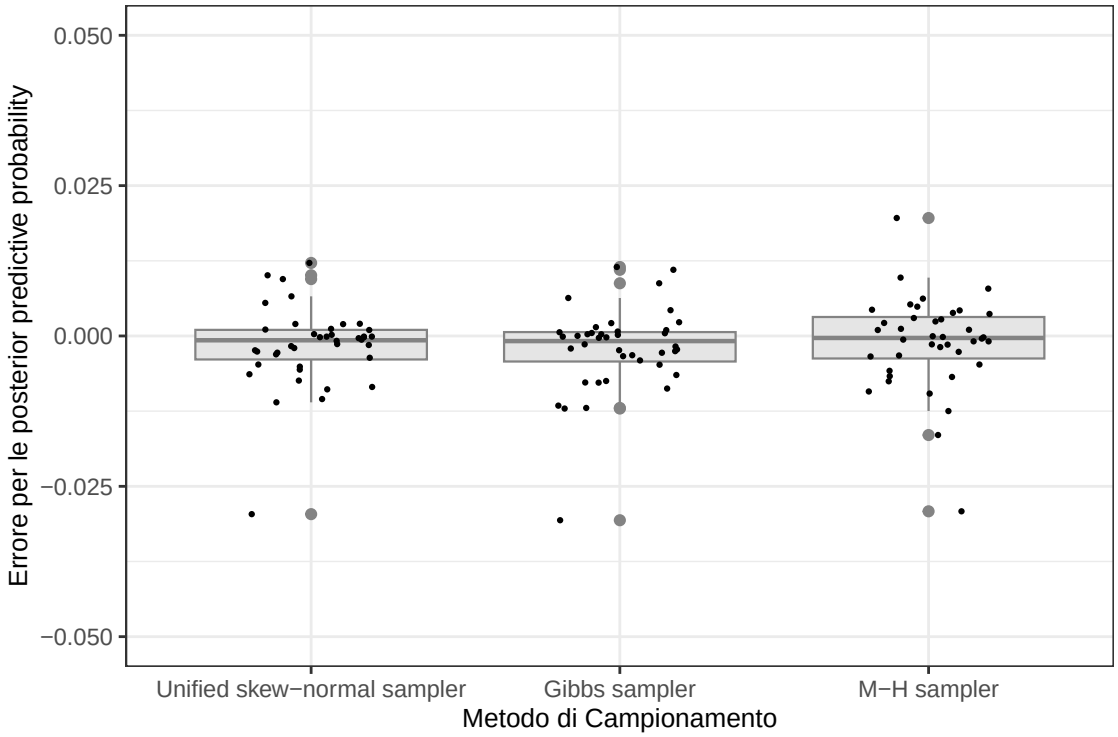


Figura 4.1: Prestazione di previsione: boxplot delle differenze tra le *posterior predictive probabilities* per le unità del test set basate sui campioni prodotti dagli algoritmi analizzati e quelle calcolate con l'espressione risultato del corollario 2. I punti rappresentano le differenze da cui poi ogni boxplot è creato.

campionamento sia quello con l'intervallo tra interquantili più esteso. Un cattivo *mixing*, infatti, comporta un effetto negativo sull'accuratezza dell'inferenza a posteriori e della previsione. Chiaramente avere a disposizione campionamenti iid comporta l'elusione di questi problemi: una qualità da sottolineare dell'algoritimo approfondito in 3.3.3.

Spunti futuri per questo studio potrebbero essere l'approfondimento di altre metodologie di campionamento come l'Adaptive Metropolis-Hastings oppure l'Hamiltonial Monte Carlo. Di certo sarebbe intrigante confrontare questi algoritmi in situazioni differenti: in Durante (2019) la parte applicativa era sviluppata con dataset con p-grande ed n-piccolo, mentre in questo studio la situazione era opposta. Un approfondimento del comportamento di questi algoritmi con dataset dalla forma diversa potrebbe portare anche a una maggior comprensione di cosa significhi il concetto di p-grande, n-grande e p-piccolo, n-piccolo. Ai fini della ricerca scientifica, sarebbe interessante avere anche una maggiore comprensione della funzione generatrice dei momenti di una *unified*

skew-normal, che potrebbe facilitare il calcolo di quantità d'interesse senza la necessità di campionare da $(\beta \mid y, X)$. Inoltre sarebbe utile migliorare il metodo di valutazione di $\Phi_n()$ in applicazioni con n-grande.

Appendice A

Codice R

```
##################################
########## Capitolo 3 ##########
##################################

library(mvtnorm)
library(TruncatedNormal)
library(tidyverse)

# Densità di una Unified Skew Normal univariata
SUN_dens <- function(x, y, z){
  xi <- 0
  Omega <- 1
  Delta <- (2*y -1)*z*(z^2 +1)^(-0.5)
  gamma <- 0
  Gamma <- 1
  omega <- 1
  bar_Omega <- 1

  valori <- dnorm(x, mean = xi, sd = sqrt(Omega))*
  pnorm(gamma+Delta*bar_Omega^(-1)*omega^(-1)*(x-xi),
  mean=0, sd = sqrt(Gamma - Delta*bar_Omega^(-1)*Delta))/
  pnorm(gamma, mean = 0, sd = srt(Gamma))}

# Grafico della densità
graph1 <- ggplot(data = data.frame(x = 0), mapping = aes(x = x))

(graph2 <- graph1+stat_function(fun = SUN_dens, args = list(y = 1,z = 0),
   color = "grey50", lwd=0.75)+ xlim(-3,3)+ylim(0, 0.7)+
   labs(title="Densità Unified Skew Normal")+xlab("Beta")+ylab("Densità")+
```

```
   stat_function(fun = SUN_dens, args = list(y = 1,z = 1.5),
   color="grey10",lwd=0.75)+stat_function(fun = SUN_dens,
   args = list(y = 1,z = 3),color="grey30",lwd=0.75)+
   stat_function(fun = SUN_dens, args = list(y = 1,
   z = -1.5),color="grey70",lwd=0.75)+
   stat_function(fun = SUN_dens, args = list(y = 1,z-3),
   color="grey87",lwd=0.75)+theme_bw())

graph3 <- ggplot(data = data.frame(x = 0), mapping = aes(x = x))

(graph4 <- graph1+stat_function(fun = SUN_dens, args = list(y = 0,z = 0),
    color = "grey50", lwd=0.75)+xlim(-3,3)+ylim(0, 0.7)+
    labs(title="Densità Unified Skew Normal")+xlab("Beta")+
    ylab("Densità")+stat_function(fun = SUN_dens,
    args = list(y = 0,z = 1.5),color="grey10",lwd=0.75)+
    stat_function(fun = SUN_dens, args = list(y = 0,z = 3),
    color="grey30",lwd=0.75)+stat_function(fun = SUN_dens,
    args = list(y = 0,z = -1.5),color="grey70",lwd=0.75)+
    stat_function(fun = SUN_dens, args = list(y = 0,z = -3),
    color="grey87",lwd=0.75)+theme_bw())

################################
########## Capitolo 4 ##########
################################

# Confronto SUN, Gibbs Sampler e MH

rm(list=ls())
library(arm)
library(MASS)

data <- MASS::Pima.tr
y_data <- data[, 8]
y_data <- as.numeric(data$type == "Yes")
X_data <- apply(data[,-8],2,rescale)
X_data <- cbind(rep(1,dim(X_data)[1]),X_data)

set.seed(123)

j <- sample(c(1:dim(X_data)[1]), nrow(X_data)*0.8 ,replace=FALSE)
y <- y_data[j]
X <- X_data[j,]
y_new <- y_data[-j]
```

```
X_new <- X_data[-j,]

fit_probit <- glm(y ~ X - 1, family = binomial(link = "probit"))
summary(fit_probit)
probit.coef <- fit_probit$coefficients

# Salvo le quantità rilevanti nel file pima_data.RData.

save(y,X,y_new,X_new,j, probit.coef, fit_probit, file="pima_data.RData")

#########################################################################
## Campionamento diretto dalla a posteriori della unified skew normal

library(mvtnorm)
library(TruncatedNormal)

load("pima_data.RData")

n <- dim(X)[1]
p <- dim(X)[2]

# Numero di campionamenti i.i.d dalla a posteriori
N_sampl <- 20000

# Si definiscono le quantità chiave per eseguire l' Algoritmo 1

Omega <- diag(16,p,p)
omega <- sqrt(diag(Omega[cbind(1:p,1:p)],p,p))
bar_Omega <- solve(omega)%*%Omega%*%solve(omega)
xi <- matrix(0,p,1)
D <- diag(2*y-1,n,n)%*%X
s <- diag(sqrt((D%*%Omega%*%t(D)+diag(1,n,n))[cbind(1:n,1:n)]),n,n)
gamma_post <- solve(s)%*%D%*%xi
Gamma_post <- solve(s)%*%(D%*%Omega%*%t(D)+diag(1,n,n))%*%solve(s)

# Rappresentazione stocastica di una SUN, risultato del corollario 1

coef_V1 <- omega%*%bar_Omega%*%omega%*%t(D)%*%solve(D%*%Omega%*%t(D)+
           diag(1,n,n))%*%s
coef_V0 <- omega
Var_V0 <- bar_Omega-bar_Omega%*%omega%*%t(D)%*%solve(D%*%Omega%*%t(D)+
          diag(1,n,n))%*%D%*%omega%*%bar_Omega
Var_V0 <- 0.5*(Var_V0+t(Var_V0))
```

```
start_time <- Sys.time()
set.seed(123)

V_0 <- t(rmvnorm(N_sampl,mean=rep(0,p),sigma=Var_V0))
V_0_scale_plus_xi <- apply(V_0,2,function(x) xi+coef_V0%*%x)
V_1 <- mvrandn(-gamma_post,rep(Inf,n),Gamma_post,N_sampl)
V_1_scale <- apply(V_1,2,function(x) coef_V1%*%x)

beta_SUN <- V_0_scale_plus_xi+V_1_scale
end_time <- Sys.time()

time_SUN <- difftime(end_time, start_time, units=("secs"))[[1]]

# Medie a posteriori
SUN_means <- apply(beta_SUN,1,mean)
# Posterior predictive probabilities
pred_SUN <- rep(0,dim(X_new)[1])
beta_SUN <- t(beta_SUN)

for (i in 1:dim(X_new)[1]){
  pred_SUN[i] <- mean(pnorm((beta_SUN%*%X_new[i,]),0,1))
  print(i)}

save(time_SUN,beta_SUN,SUN_means,pred_SUN,file="pima_output.RData")

############################################################
## Gibbs Sampler per regressione probit

rm(list=ls())
library(truncnorm)
library(coda)
library(mixtools)

load("pima_data.RData")

n <- dim(X)[1]
p <- dim(X)[2]

# Numero di campionamenti MCMC e perido di burn-in
N_sampl <- 25000
burn_in <- 5000
```

```
gibbs_R <- function(y, X, b, B, M, Burn_in){
  beta <- (t(X)%*%X)^(-1)%*%t(X)%*%y
  out <- matrix(0, nrow = M, ncol = length(beta))
  beta <- (t(X)%*%X)^(-1)%*%t(X)%*%y
  for (i in 1:(burn_in + M)){
    N1  <- sum(y)
    N0  <- length(y) - N1
    mu_z <- X %*% beta
    z <- c()

    for (j in 1:length(mu_z)){
      if (y[j] == 0){
        z[j] <- rtruncnorm(1, -Inf, 0 ,mu_z[j], 1)
      } else {z[j] <- rtruncnorm(1, 0, Inf, mu_z[j], 1)}
    }

    sigma <- solve(solve(B)+t(X)%*%X)
    mu <- sigma%*%(solve(B)%*%b + t(X)%*%z)
    beta <- t(rmvnorm(1, mu, sigma))

    if (i > burn_in){
      out[i - burn_in, ] <- beta
    }
  }
  out
}


# Distribuzione a priori per beta
b <- matrix(0, nrow = dim(X)[2], ncol = 1)
B <- diag(16, nrow = dim(X)[2], ncol = dim(X)[2])

start_time <- Sys.time()
set.seed(123)

fit.GIBBS <- gibbs_R(y, X, b, B, N_sampl, burn_in)

end_time <- Sys.time()

time_GIBBS <- difftime(end_time, start_time, units=("secs"))[[1]]

fit.GIBBS <- as.mcmc(fit.GIBBS)
# Grafici coda
```

```
plot(fit.GIBBS[, 1:3])
plot(fit.GIBBS[, 4:6])
plot(fit.GIBBS[, 7])

# Medie a posteriori
GIBBS_means <- colMeans(fit.GIBBS)

# Posterior predictive probabilities
pred_GIBBS <- rep(0,dim(X_new)[1])

for (i in 1:dim(X_new)[1]){
  pred_GIBBS[i] <- mean(pnorm((fit.GIBBS%*%X_new[i,]),0,1))
  print(i)}

save(time_GIBBS,fit.GIBBS,GIBBS_means,pred_GIBBS,file="GIBBS_output.RData")

############################################################
## Algoritmo MH per regressione probit

rm(list=ls())
library(coda)
library(mixtools)

load("pima_data.RData")

n <- dim(X)[1]
p <- dim(X)[2]

# Numero di campionamenti MCMC e perido di burn-in
N_sampl <- 25000
burn_in <- 5000

# Loglikelihood del modello di regressione probit
loglik <- function(beta, y, X) {
  sum(dbinom(y, size = 1, prob = pnorm(X%*%beta), log = TRUE))
}
# Logposterior
logpost <- function(beta, y, X) {
  loglik(beta, y, X) + sum(dnorm(beta, 0, 4, log = T))
}

MH.probit <- function(M, x, y, S, burn_in) {
  n <- length(y)
```

```
  p <- dim(x)[2]

  BETA <- matrix(0, M, p)
  beta <- rep(0, p)
  logp <- logpost(beta, y, x)

  for (r in 1:(M+burn_in)) {
    beta.p <- t(rmvnorm(1, beta, S)) #proposal distribution
    logp.new <- logpost(beta.p, y, x)
    alpha <- min(1, exp(logp.new - logp))
    if (runif(1) < alpha) {
      beta <- beta.p
      logp <- logp.new
    }
    if (r > burn_in){
      BETA[r-burn_in, ] <- beta
    }
  }
  BETA
}

M <- 25000
burn_in <- 5000

# Approssimazione di Laplace
S.Laplace <- 2.38^2 * vcov(fit_probit)/dim(X)[2]

start_time <- Sys.time()
set.seed(123)

fit.MH <- MH.probit(M,X,y,S.Laplace, burn_in)

end_time <- Sys.time()
time_MH <- difftime(end_time, start_time, units=("secs"))[[1]]

fit.MH <- as.mcmc(fit.MH)
# Grafici coda
plot(fit.MH[, 1:3])
plot(fit.MH[, 4:6])
plot(fit.MH[, 7])

# Medie a posteriori
MH_means <- colMeans(fit.MH)
```

```
# Posterior predictive probabilities
pred_MH <- rep(0,dim(X_new)[1])

for (i in 1:dim(X_new)[1]){
  pred_MH[i] <- mean(pnorm((fit.MH%*%X_new[i,]),0,1))
  print(i)}

save(time_MH,fit.MH,MH_means,pred_MH,file="MH_output.RData")

############################################################
## Implementazione dell'inferenza esatta

rm(list=ls())
library(mvtnorm)
library(TruncatedNormal)

load("pima_data.RData")

n <- dim(X)[1]
p <- dim(X)[2]

# Parametri rilevanti per calcolare medie a posteriori
# e posterior predictive probabilities
Omega <- diag(16,p,p)
omega <- sqrt(diag(Omega[cbind(1:p,1:p)],p,p))
bar_Omega <- solve(omega)%*%Omega%*%solve(omega)
xi <- matrix(0,p,1)
D <- diag(2*y-1,n,n)%*%X
s <- diag(sqrt((D%*%Omega%*%t(D)+diag(1,n,n))[cbind(1:n,1:n)]),n,n)
gamma_post <- solve(s)%*%D%*%xi
Gamma_post <- solve(s)%*%(D%*%Omega%*%t(D)+diag(1,n,n))%*%solve(s)

set.seed(123)

# Costante di Normalizzazione
Norm_const <- mvNcdf(l=rep(-Inf,n),u=gamma_post,Sig=Gamma_post,10^(6))$prob

eta <- matrix(0,n,1)
for (i in 1:n){
  eta[i,1] <- dnorm(gamma_post[i])*mvNcdf(l=rep(-Inf,n-1),u=gamma_post[-i]-
  (Gamma_post[,i])[-i]*gamma_post[i],Sig=(Gamma_post[,-i])[-i,]-
  (Gamma_post[,i])[-i]%*%t((Gamma_post[,i])[-i]),10^(5.5))$prob
```

```
  print(i)}

# Medie a posteriori senza campionare dalla SUN
NUMERICAL_means <- xi+Omega%*%t(D)%*%solve(s)%*%(eta/Norm_const)
probit.coef

set.seed(123)

# Costante di normalizzazione per i dati di addestramento
Norm_const_obs <- mvNcdf(l=rep(-Inf,n),u=gamma_post,Sig=Gamma_post,10^6)$prob

pred_NUMERICAL <- rep(0,dim(X_new)[1])

# Calcolo delle posterior predictive probabilities per le unità del test
# set senza campionare dalla SUN
for (i in 1:dim(X_new)[1]){
  D_new <- rbind(D,X_new[i,])
  s_new <- diag(sqrt((D_new%*%Omega%*%t(D_new)+diag(1,n+1,n+1))
           [cbind(1:(n+1),1:(n+1))]),n+1,n+1)
  gamma_new <- solve(s_new)%*%D_new%*%xi
  Gamma_new <- solve(s_new)%*%(D_new%*%Omega%*%t(D_new)+diag(1,n+1,n+1))%*%
               solve(s_new)

  pred_NUMERICAL[i] <- mvNcdf(l=rep(-Inf,n+1),u=gamma_new,
                       Sig=Gamma_new,10^(5.5))$prob/Norm_const_obs
  print(i)}

save(NUMERICAL_means, pred_NUMERICAL,file="NUMERICAL_output.RData")

############################################################
## Valutazione delle prestazioni

rm(list=ls())

library(ggplot2)
library(coda)
library(reshape)
library(knitr)

load("pima_data.RData")
load("pima_output.RData")
load("GIBBS_output.RData")
load("MH_output.RData")
```

```
load("NUMERICAL_output.RData")

# Creazione Tabella numero tabella
N_sampl_SUN <- 20000
N_sampl <- 25000

Table_perf <- matrix(0,3,4)
rownames(Table_perf) <- c("Unified skew-normal sampler", "Gibbs sampler",
                          "M-H sampler")
colnames(Table_perf) <- c("Samples of beta per sec.", "Min ESS", "Q1 ESS",
                          "Median ESS")

# Unified skew-normal sampler
Table_perf[1,c(2:4)] <- N_sampl_SUN
Table_perf[1,1] <- N_sampl_SUN/time_SUN

# Gibbs sampler
Table_perf[2,c(2:4)] <- summary(apply(fit.GIBBS,2,effectiveSize))[1:3]
Table_perf[2,1] <- N_sampl/time_GIBBS

# Metropolis-Hastings sampler
Table_perf[3,c(2:4)] <- summary(apply(fit.MH,2,effectiveSize))[1:3]
Table_perf[3,1] <- N_sampl/time_MH

kable(Table_perf)

# Grafico numero da decidere
# Unified skew-normal sampler
data_matrix_SUN_plot <- c(pred_SUN-pred_NUMERICAL)
data_matrix_SUN_plot <- melt(data_matrix_SUN_plot)
data_matrix_SUN_plot$method <- c("Unified skew-normal sampler")

# Gibbs sampler
data_matrix_GIBBS_plot <- c(pred_GIBBS-pred_NUMERICAL)
data_matrix_GIBBS_plot <- melt(data_matrix_GIBBS_plot)
data_matrix_GIBBS_plot$method <- c("Gibbs sampler")

# Metropolis-Hastings sampler
data_matrix_MH_plot <- c(pred_MH-pred_NUMERICAL)
data_matrix_MH_plot <- melt(data_matrix_MH_plot)
data_matrix_MH_plot$method <- c("M-H sampler")

data_final_plot <- rbind(data_matrix_GIBBS_plot,data_matrix_SUN_plot,
```

```
                    data_matrix_MH_plot)
data_final_plot$method <- factor(data_final_plot$method,
                          levels=c("Unified skew-normal sampler","Gibbs sampler",
                          "M-H sampler"))
data_final_plot$description <- "Quality in posterior predictive probability
                                calculation via Monte Carlo methods"

set.seed(123)
ggplot(data_final_plot, aes(x=method, y=value))+
  geom_boxplot(color="#838383",fill="#e5e5e5",lwd=0.4)+
  theme_bw()+ geom_jitter(width = 0.2,shape=16,size=0.8)+
  ylab("Errore per le posterior predictive probability")+
  xlab("Metodo di Campionamento")+theme(axis.title.x = element_text(size=10),
  axis.title.y = element_text(size=10),strip.text = element_text(size=12))+
  ylim(-0.1,0.1)
```

Bibliografia

Albert, J. H. & Chib, S. (1993). Bayesian analysis of binary and polychotomous response data. *Journal of the American Statistical Association* **88**, 669–679.

Arrellano-Valle, R. & Azzalini, A. (2006). On the unification of families of skew-normal distributions. *Scand. J. Statist.* **33**, 561–74.

Azzalini, A. & Capitanio, A. (2014). *The Skew-Normal and Related Families*. Cambridge University Press.

Botev, Z. (2017). The normal law under linear restrictions: Simulation and estimation via minimax tilting. *The Journal of the Royal Statistical Society, Series B* **79**, 125–148.

Chopin, N. & Ridgway, J. (2017). Leave Pima Indians Alone: Binary Regression as a Benchmark for Bayesian Computation. *Statistical Science* **32**, 64 – 87.

Durante, D. (2019). Conjugate bayes for probit regression via unified skew-normal distributions. *Biometrika* **106**, 765–779.

Gelman, A. & Hennig, C. (2015). Beyond subjective and objective in statistics .

Gelman, A., Jakulin, A., Pittau, M. G. & Su, Y.-S. (2008). A weakly informative default prior distribution for logistic and other regression models. *The Annals of Applied Statistics* **2**.

Hoff, P. D. (2009). *A First Course in Bayesian Statistical Methods*. Springer.

RDocumentation (2023). Pima.tr: Diabetes in pima indian women.

Robert, C. & Casella, G. (1999). *Monte Carlo Statistical Methods*. Springer.

Robert, C. & Casella, G. (2010). *Introducing Monte Carlo Methods with R*. Springer.

Robert, C. P. (1991). *The Bayesian Choice A Decision-Theoretic Motivation*. Springer.