Università degli studi di Padova
Dipartimento di Scienze Statistiche

Corso di Laurea Magistrale in Scienze Statistiche

Selezione bayesiana delle variabili nel modello di regressione logistica ad alta dimensionalità

Relatore Prof. Mauro Bernardi
Dipartimento di Scienze Statistiche

Correlatore Dott. Andrea Sottosanti
Dipartimento di Scienze Statistiche

Laureando Claudio Busatto
Matricola N 1155623

Anno Accademico 2018/2019

Indice

List of Algorithms

Elenco dei codici

Introduzione

Un campo statistico in piena evoluzione è senza dubbio quello biostatistico, con una vastità di nuove metodologie e risorse computazionali in grado di aprire nuovi orizzonti. In particolare, la statistica genetica è diventata di fondamentale importanza per l'analisi e la ricerca in campo genetico.

Dopo l'introduzione del metodo Sanger nel 1977 (Sanger, Nicklen e Coulson, 1977), il sequenziamento del DNA fu l'oggetto di ricerca di un elevato numero di ricercatori. Il *Progetto Genoma Umano* (HGP), il progetto internazionale di ricerca scientifica più ampio della storia, venne lanciato nel 1990 con l'obiettivo di determinare la sequenza di nucleotidi che compongono il codice genetico umano. Tale sfida fu portata a termine nel 2003 dopo miliardi di dollari investiti. L'evoluzione tecnologica portò, pochi anni dopo, alla realizzazione di nuove macchine per l'analisi del DNA, introducendo il sequenziamento *next-generation* e facendo crollare i tempi e i costi per leggere e codificare un intero genoma: oggigiorno è possibile sequenziare un intero DNA umano nel giro di 24 ore e con poche migliaia di dollari.

Una caratteristica dei dataset genetici è quella di disporre di un numero di variabili molto elevato, se messo a rapporto con il numero di osservazioni a disposizione. Questa proprietà esclude l'uso dei più classici metodi per l'analisi dei dati e giustifica l'applicazione di tecniche bayesiane, con le quali si ha la possibilità di svolgere l'inferenza in maniera semplice una volta ottenuta la distribuzione a-posteriori. L'esplorazione di questa non è sempre immediata ed è necessario ricorrere ad ipotesi specifiche, quali la coniugazione tra la distribuzione del parametro e la verosimiglianza del modello, o a tecniche di *data-augmentation* per poter implementare l'algoritmo *Gibbs Sampling*. Un altro metodo per poter esplorare la distribuzione a-posteriori del parametro

di interesse è dato dall'algoritmo *Metropolis-Hastings*, metodo ricorrente in questo lavoro.

Questa tesi trova le sue fondamenta nella nuova distribuzione *Polya-Gamma*, creata su misura da Polson, Scott e Windle (2012) per la modellizzazione di dati dicotomici. La nuova tecnica di *data-augmentation* prevede l'inserimento di tale variabile latente nel modello di regressione logistica e permette di ottenere una distribuzione *full-conditional* in forma chiusa per i parametri associati ai regressori del modello, cosa non possibile in precedenza. La distribuzione Normale Multivariata per i parametri, infatti, è coniugata rispetto alla verosimiglianza del modello nell'ambito della regressione lineare, tuttavia perde questa caratteristica quando si lavora con una verosimiglianza binomiale. In questa chiave vengono rivisitati tecniche e metodi quali l'approccio *Spike and Slab* e *Reversible-Jump Markov Chain*: i due metodi vengono analizzati sotto determinate ipotesi, grazie alle quali si può dimostrare che, aumentando il dataset con la variabile latente Polya-Gamma, si è in possesso di un nuovo efficiente schema per la selezione delle variabili nel caso di dati binari quando il numero di regressori è particolarmente elevato. Inoltre, viene analizzata anche l'estensione *Multiple-Try* dell'algoritmo Reversible-Jump, aumentando la flessibilità del metodo.

L'obiettivo di questo lavoro è la formalizzazione di nuove tecniche bayesiane per l'esplorazione della distribuzione a-posteriori e la selezione dei regressori nel delicato caso in cui il numero di modelli possibili non sia numerabile. In quest'ultimo caso, infatti, l'analisi di tutto lo spazio parametrico non è fattibile ed è necessario uno schema che riesca a restringere il problema allo studio di una sotto-classe di modelli maggiormente supportata dai dati.

Durante tutte le analisi è stato sempre preso in considerazione anche il carico computazionale: per gli algoritmi *Metropolis-Hastings* presentati viene scelta una funzione proposta in grado di facilitare leggermente i calcoli. Tuttavia, proprio tale proposta è il limite principale degli schemi presentati, in quanto le catene costruite si muovono sempre localmente, non permettendo di spaziare liberamente tra i modelli, proprietà desiderabile nel caso di distribuzioni obiettivo di dimensioni elevate. Inoltre, non gode di proprietà ottimali in ottica computazionale.

Il lavoro è suddiviso in 6 Capitoli: il Capitolo 1 propone una breve introduzione sulla Statistica bayesiana e sul metodo Monte Carlo Markov Chain per generare da una distribuzione obiettivo; nel Capitolo 2 viene formalizzato l'approccio Spike and Slab per la selezione delle variabili nel caso della regressione lineare e, successivamente, viene presentato uno schema per la sua generalizzazione a dati binari tramite l'inserimento della variabile latente Polya-Gamma; il Capitolo 3 formalizza l'approccio Reversible-Jump per la selezione del modello e presenta una nuova estensione valida per dati dicotomici tramite l'inserimento della variabile latente Polya-Gamma; il Capitolo 4 presenta l'approccio Multiple-Try per generare da una distribuzione obiettivo e formalizza la sua applicazione al metodo Reversible-Jump sia per dati continui sia per dati binari, sempre tramite l'inserimento della variabile latente Polya-Gamma; il Capitolo 5 mostra l'applicazione degli algoritmi presentati a dataset simulati e a dati reali; infine, nel Capitolo 6 vengono discussi possibili miglioramenti ed estensioni, relativamente alla diagnostica dei modelli ed alla possibilità di usare nuove proposte per i nuovi valori della catena. Le analisi vengono svolte con il software R.

Capitolo 1

Statistica bayesiana

La statistica bayesiana deve il proprio nome al teorema di Bayes e permette di aggiungere conoscenze a-priori all'evidenza dei dati. In questo campo tutto ciò che è ignoto è considerato casuale, ed amette, dunque, una distribuzione: il parametro $\boldsymbol{\theta}$ che governa il modello è esso stesso realizzazione di una variabile casuale. La scelta della distribuzione a-priori per il parametro deve rispecchiare le informazioni a disposizione, assegnando maggior probabilità ai valori che si ritengono più plausibili per $\boldsymbol{\theta}$. Dato il vettore di dati $\mathbf{y} = (y_1, \ldots, y_n)$, indichiamo con $f(\mathbf{y} \mid \boldsymbol{\theta})$ la sua funzione di densità. La funzione di verosimiglianza è definita come

$$L(\boldsymbol{\theta}) = f(y_1, \ldots, y_n \mid \boldsymbol{\theta}) = \prod_{i=1}^{n} f(y_i \mid \boldsymbol{\theta}), \tag{1.1}$$

che vale nel caso le osservazioni siano indipendenti. Assumendo $\boldsymbol{\theta} \sim \pi(\boldsymbol{\theta})$, dal teorema di Bayes si ha

$$\pi(\boldsymbol{\theta} \mid \mathbf{y}) = \frac{f(\mathbf{y} \mid \boldsymbol{\theta})\pi(\boldsymbol{\theta})}{f(\mathbf{y})} = \frac{f(\mathbf{y} \mid \boldsymbol{\theta})\pi(\boldsymbol{\theta})}{\int_\Theta f(\mathbf{y} \mid \boldsymbol{\theta})\pi(\boldsymbol{\theta}) d\boldsymbol{\theta}}. \tag{1.2}$$

Il termine al denominatore non dipende dal parametro d'interesse, ed è dunque trattabile come una costante. In altre parole, la funzione che esprime la distribuzione del parametro condizionatamente ai dati osservati è

$$\pi(\boldsymbol{\theta} \mid \mathbf{y}) \propto f(\mathbf{y} \mid \boldsymbol{\theta})\pi(\boldsymbol{\theta}). \tag{1.3}$$

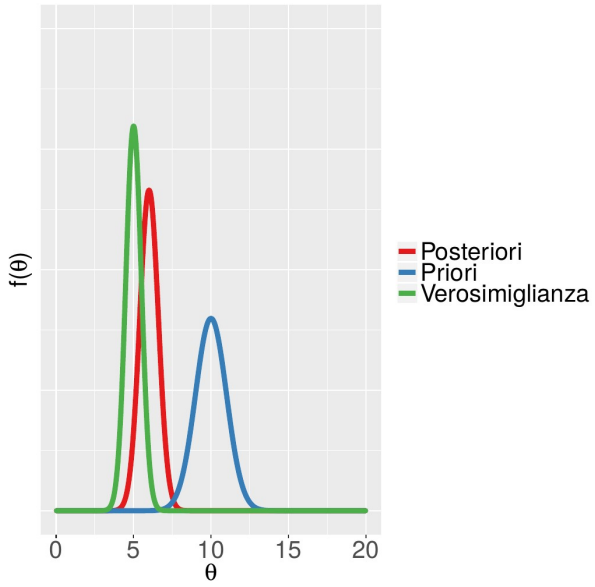


Figura 1.1: Applicazione del teorema di Bayes.

La figura 1.1 mostra un esempio di come la funzione di verosimiglianza e la distribuzione a-priori influenzano la distribuzione a-posteriori: questa si avvicina molto alla funzione di verosimiglianza ma risente dell'informazione inserita a-priori, che suggerisce un valore per il parametro più elevato rispetto a quanto indicato dai dati.

La statistica bayesiana presenta vantaggi e svantaggi. Il problema maggiore è dal punto di vista computazionale: se il vettore di parametri è di grandi dimensioni, l'integrale al denominatore di (1.2) è difficilmente calcolabile. Tale problema è sormontabile se si è disposti ad accettare distribuzioni a-priori coniugate per i parametri: una distribuzione si dice coniugata se $\pi(\boldsymbol{\theta} \mid \mathbf{y})$ segue la stessa distribuzione di $\pi(\boldsymbol{\theta})$. Si evita così il calcolo esplicito dell'integrale.

Nel caso in cui non vi sia abbastanza informazione a-priori sul parametro, la scelta di $\pi(\boldsymbol{\theta})$ inserisce soggettività non giustificata nel modello. Di fronte a tale problema si possono assumere distribuzioni dette non informative, che condizionano in maniera minima la distribuzione a-posteriori: una funzione di densità piatta sullo spazio di $\boldsymbol{\theta}$ assegna probabilità simili ai vari valori del parametro, forzando la distribuzione a-posteriori a concentrarsi maggiormente sull'informazione proveniente dai dati osservati. Al contrario,

in alcune circostanze si ha a disposizione una buona conoscenza del fenomeno in questione: l'arbitrarietà della funzione $\pi(\boldsymbol{\theta})$ diventa, in questo caso, un vantaggio, in quanto è possibile influenzare la distribuzione a-posteriori aggiungendo informazioni derivanti dall'esterno.

Un vantaggio notevole è dato dal fatto che tutta l'informazione per l'inferenza è racchiusa in $\pi(\boldsymbol{\theta} \mid \mathbf{y})$: a partire da questa si possono ricavare le stime degli indici di posizione e di variabilità e permette, inoltre, di costruire intervalli di credibilità a livello di significatività α, tali per cui

$$Pr(\boldsymbol{\theta} \in [a, b] \mid \mathbf{y}) = 1 - \alpha. \tag{1.4}$$

1.1 Markov Chain Monte Carlo (MCMC)

Lo studio della distribuzione a-posteriori (1.3) diventa di difficile applicazione qualora la dimensione del vettore $\boldsymbol{\theta}$ sia elevata. In questo caso l'approssimazione *Monte Carlo* entra in aiuto. Si supponga di voler stimare una quantità I_θ del tipo

$$I_\theta = E(f(\boldsymbol{\theta})) = \int_\Theta f(\boldsymbol{\theta})\pi(\boldsymbol{\theta} \mid \mathbf{y})d\boldsymbol{\theta} : \tag{1.5}$$

tale stima diventa meno efficiente all'aumetare dello spazio parametrico. E' di uso comune approssimare I_θ tramite Monte Carlo: potendo disporre di un campione i.i.d. di numerosità B per la distribuzione a-posteriori (1.3), allora la stima di I_θ è data da

$$\hat{I}_\theta = \frac{1}{B}\sum_{b=1}^{B} f(\boldsymbol{\theta}_b), \tag{1.6}$$

e converge in probabilità a I_θ per la legge debole dei grandi numeri.

Tuttavia l'ipotesi di indipendenza e identica distribuzione per il campione proveniente da $\pi(\boldsymbol{\theta} \mid \mathbf{y})$ non è sempre rispettata. Gli algoritmi MCMC permettono di generare una catena di Markov ergodica e invariante per una distribuzione obiettivo del tipo (1.3). Tuttavia, gli stati della catena $\boldsymbol{\theta}_b$ sono correlati tra loro: la stima Monte Carlo perde in efficienza ma rimane consistente. Per ovviare a tale problema è necessario disporre di un campione più numeroso e ridurre la correlazione del campione tramite l'applicazione di particolari test, come il test di auto-correlazione (ACF).

1.2 Metropolis-Hastings (MH)

Uno degli algoritmi più famosi per la costruzione di catene di Markov è l'algoritmo *Metropolis-Hastings* proposto da Hastings (1970): si supponga di essere nello stato $\boldsymbol{\theta}$ della catena e di voler esplorare un nuovo stato $\boldsymbol{\theta}'$, proposto a partire da una densità $q(\boldsymbol{\theta}' \mid \boldsymbol{\theta})$. Ad ogni iterazione il valore $\boldsymbol{\theta}'$ viene accettato con probabilità $\alpha(\boldsymbol{\theta}, \boldsymbol{\theta}')$. Una proprietà fondamentale per la convergenza dell'algoritmo è quella di reversibilità, tale per cui

$$\pi(\boldsymbol{\theta} \mid \mathbf{y}) q(\boldsymbol{\theta}' \mid \boldsymbol{\theta}) \alpha(\boldsymbol{\theta}, \boldsymbol{\theta}') = \pi(\boldsymbol{\theta}' \mid \mathbf{y}) q(\boldsymbol{\theta} \mid \boldsymbol{\theta}') \alpha(\boldsymbol{\theta}', \boldsymbol{\theta}). \tag{1.7}$$

L'algoritmo MH pone $\alpha(\boldsymbol{\theta}, \boldsymbol{\theta}') = min\left(1, \frac{w(\boldsymbol{\theta}' \mid \boldsymbol{\theta})}{w(\boldsymbol{\theta} \mid \boldsymbol{\theta}')}\right)$, dove $w(\boldsymbol{\theta}' \mid \boldsymbol{\theta}) = \frac{\pi(\boldsymbol{\theta}' \mid \mathbf{y})}{q(\boldsymbol{\theta}' \mid \boldsymbol{\theta})}$, soddisfacendo la condizione (1.7) e garantendo una catena ergodica e con distribuzione invariante $\pi(\boldsymbol{\theta} \mid \mathbf{y})$. I pesi w così definiti sono detti *importance weigths*, ma non sono gli unici che portano a catene di Markov ergodiche e con distribuzione invariante (1.3): è possibile dimostrare che tali proprietà sono raggiungibili se i pesi sono definiti come

$$w(\boldsymbol{\theta}' \mid \boldsymbol{\theta}) = \pi(\boldsymbol{\theta}' \mid \mathbf{y}) q(\boldsymbol{\theta} \mid \boldsymbol{\theta}') \xi(\boldsymbol{\theta}, \boldsymbol{\theta}'), \tag{1.8}$$

dove $\xi(\boldsymbol{\theta}, \boldsymbol{\theta}') = \xi(\boldsymbol{\theta}', \boldsymbol{\theta})$, Martino (2018). Se si pone $\xi(\boldsymbol{\theta}, \boldsymbol{\theta}') = [q(\boldsymbol{\theta} \mid \boldsymbol{\theta}') q(\boldsymbol{\theta}' \mid \boldsymbol{\theta})]^{-1}$ si ottengono gli *importance weights*. Scegliendo la funzione costante $\xi(\boldsymbol{\theta}, \boldsymbol{\theta}') = 1$, invece, si ottengono i pesi $w(\boldsymbol{\theta}' \mid \boldsymbol{\theta}) = \pi(\boldsymbol{\theta}' \mid \mathbf{y}) q(\boldsymbol{\theta} \mid \boldsymbol{\theta}')$. Infine, un caso interessante è quello in cui la densità per la proposta $q(\boldsymbol{\theta}' \mid \boldsymbol{\theta})$ è simmetrica in $\boldsymbol{\theta}$ e $\boldsymbol{\theta}'$: imponendo $\xi(\boldsymbol{\theta}', \boldsymbol{\theta}) = q(\boldsymbol{\theta} \mid \boldsymbol{\theta}')^{-1}$ si ottengono dei pesi pari alla densità a-posteriori calcolata in $\boldsymbol{\theta}'$, $w(\boldsymbol{\theta}' \mid \boldsymbol{\theta}) = \pi(\boldsymbol{\theta}' \mid \mathbf{y})$.

Capitolo 2

Spike and Slab

2.1 Spike and Slab prior

Viene introdotto ora l'approccio *Spike and Slab* per la selezione dei regressori di un modello in ambito bayesiano. Questa tecnica venne implementata da George e McCulloch (1997) per la regressione lineare: sia $\mathbf{y} = (y_1, \ldots, y_n)$ il vettore risposta e $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{ip})$, $i = 1, \ldots, n$, un vettore p-dimensionale relativo alle covariate associate all'i-mo individuo. Si assume

$$\mathbf{y} = \mathbf{X}\boldsymbol{\theta} + \boldsymbol{\varepsilon}, \tag{2.1}$$

con $\boldsymbol{\varepsilon} \sim \mathsf{N}(0, \sigma^2 \mathbf{I}_n)$. L'idea è quella di sfruttare una distribuzione a-priori per il vettore di parametri $\boldsymbol{\theta}$ composta da una mistura di due componenti, per poter inferire a posteriori la probabilità di inclusione della j-ma variabile, $j = 1, \ldots, p$, nel modello. La scelta delle due componenti deve comprendere una distribuzione concentrata in 0 ed una più uniforme sullo spazio parametrico: per quanto riguarda la prima, scelte ragionevoli sono la Delta di Dirac e la distribuzione Normale con varianza prossima allo 0, mentre per la seconda è di uso comune la distribuzione Normale con varianza elevata.

Viene definito il vettore $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_p)$, in cui $\gamma_j = 0$ se il j-mo parametro è 0, 1 altrimenti. Si ha, dunque, $\pi(\gamma_j \mid \pi_j, \theta_j) \sim \mathsf{Be}(\pi_j)$. A questo punto è, dunque, possibile scrivere la distribuzione a-priori nel seguente modo

$$\pi(\theta_j \mid \sigma^2, \gamma_j) = (1 - \gamma_j)\mathsf{N}(0, \sigma^2 v_0) + \gamma_j \mathsf{N}(0, \sigma^2 v_1), \tag{2.2}$$

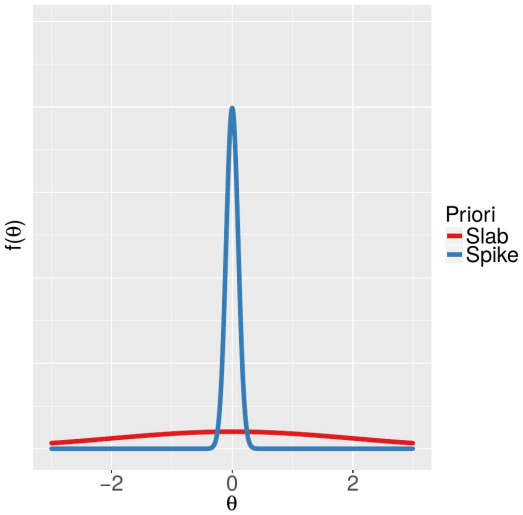


Figura 2.1: Densità delle componenti *spike* $N(0, \sigma^2 v_0)$ e *slab* $\mathsf{N}(0, \sigma^2 v_1)$.

con $v_0 \ll v_1$, rappresentata in figura 2.1.

Le variabili θ_j, $j = 1, \ldots, p$ dipendono da un parametro σ^2 associato alla loro variabilità. Questo si assume provenire da una distribuzione $\mathsf{IG}(\lambda, \nu)$. Sotto tali ipotesi è possibile costruire un modello bayesiano gerarchico: le densità a-priori dei parametri $\boldsymbol{\theta}$ e σ^2 sono coniugate rispetto alla verosimiglianza del modello e le loro distribuzioni *full conditionals* ammettono una soluzione analitica, consentendo l'uso del Gibbs Sampling. Tuttavia, è la distribuzione marginale a-posteriori $\pi(\boldsymbol{\gamma} \mid \mathbf{y})$ a racchiudere tutta l'informazione per la selezione delle variabili e per l'individuare dei modelli maggiormente supportati dai dati. Questa si ricava integrando $\boldsymbol{\theta}$ e σ^2 dalla distribuzione congiunta a-posteriori

$$\pi(\boldsymbol{\gamma}, \boldsymbol{\theta}, \sigma^2 \mid \mathbf{y}) = f(\mathbf{y} \mid \boldsymbol{\theta}, \boldsymbol{\gamma}) \pi(\boldsymbol{\theta} \mid \boldsymbol{\gamma}) \pi(\boldsymbol{\gamma}) \pi(\sigma^2), \tag{2.3}$$

dove $\pi(\boldsymbol{\gamma}) \sim \mathsf{Bin}(p, \phi)$, con ϕ probabilità a-priori di inserimento delle variabili nel modello, e si può dimostrare (George e McCulloch (1997)) che dal calcolo dell'integrale risulta

$$\pi(\boldsymbol{\gamma} \mid \mathbf{y}) \propto g(\boldsymbol{\gamma}) = \left| \tilde{\mathbf{X}}' \tilde{\mathbf{X}} \right|^{-1/2} |\boldsymbol{\Sigma}_\theta|^{-1/2} \left(\nu\lambda + S_\gamma^2 \right)^{-(n+\nu)/2} \pi(\boldsymbol{\gamma}), \tag{2.4}$$

con

$$\tilde{\mathbf{X}} = \begin{bmatrix} \mathbf{X} \\ \boldsymbol{\Sigma}_\theta^{-1/2} \end{bmatrix} \qquad \text{e} \qquad S_\gamma^2 = \mathbf{y}'\mathbf{y} - \mathbf{y}'\mathbf{X}(\mathbf{X}'\mathbf{X} + \boldsymbol{\Sigma}_\theta^{-1})^{-1}\mathbf{X}'\mathbf{y}, \tag{2.5}$$

dove la matrice $\boldsymbol{\Sigma}_\theta$ è la matrice di varianza-covarianza della distribuzione a-piori di $\boldsymbol{\theta}$. Quest'ultima è descritta in (2.2) ed è alternativamente rappresentabile come $\pi(\boldsymbol{\theta} \mid \boldsymbol{\gamma}, \sigma^2) \sim \mathsf{N}(0, \mathbf{D}_\gamma^{-1/2}\mathbf{R}\mathbf{D}_\gamma^{-1/2})$, con $\mathbf{R}$ matrice di correlazione e $\mathbf{D}_\gamma$ matrice diagonale, che dipende dai parametri σ^2 e $\boldsymbol{\gamma}$; il generico elemento della diagonale è $\mathbf{D}_\gamma^{[j,j]} = \sigma^2\left((1-\gamma_j)v_0 + \gamma_j v_1\right)$.

Un semplice metodo per la costruzione di una catena

$$\boldsymbol{\gamma}^{(1)}, \ldots, \boldsymbol{\gamma}^{(B)} \tag{2.6}$$

da (2.4) è quello di considerare la distribuzione condizionata

$$\gamma_j \mid \boldsymbol{\gamma}_{(j)}, \mathbf{y}, j = 1, \ldots, p, \tag{2.7}$$

dove $\boldsymbol{\gamma}_{(j)} = (\gamma_1, \ldots, \gamma_{j-1}, \gamma_{j+1}, \ldots, \gamma_p)$. Le componenti di (2.6) sono generate a partire da una distribuzione Bernoulliana, il cui parametro π_j, $j = 1, \ldots, p$, è governato dal rapporto

$$\frac{\pi(\gamma_j = 1, \boldsymbol{\gamma}_{(j)} \mid \mathbf{y})}{\pi(\gamma_j = 0, \boldsymbol{\gamma}_{(j)} \mid \mathbf{y})} = \frac{g(\gamma_j = 1, \boldsymbol{\gamma}_{(j)})}{g(\gamma_j = 0, \boldsymbol{\gamma}_{(j)})}. \tag{2.8}$$

In particolare, la probabilità di inserimento della variabile j-ma nel modello viene aggiornata secondo la seguente relazione:

$$p(\pi_j \mid \mathbf{y}, \mathbf{X}, \boldsymbol{\theta}, \sigma^2) = \frac{1}{1 + \dfrac{(1-\pi_j)}{\pi_j}\dfrac{g(\gamma_j = 0)}{g(\gamma_j = 1)}}. \tag{2.9}$$

Per il calcolo del denominatore in (2.8) è sufficiente togliere la colonna j-ma di $\hat{\mathbf{X}}$ da (2.4), mentre questa viene inserita al numeratore. Ad ogni iterazione dell'algoritmo, uno dei due valori in (2.8) è riciclabile dal passo precedente, mentre è possibile aggiornare $g(\gamma_j)$ in maniera efficiente, in seguito all'inserimento/eliminazione di una riga in $\hat{\mathbf{X}}$, seguendo l'algoritmo di Chambers (1971). L'integrazione dei parametri $\boldsymbol{\theta}$ e σ^2 dalla distribuzione a-posteriori porta il Gibbs Sampling a convergere più velocemente rispetto allo schema generale in cui anche le distribuzioni *full-conditional* dei parametri integrati vengono esplorate (Liu, Wong e Kong, 1994).

2.2 Spike and Slab per dati binari

Viene presentato ora uno schema per la selezione di variabili in caso di dati binari sfruttando l'approccio Spike and Slab trattato nella sezione precedente. A differenza del modello lineare, la distribuzione Normale non è coniugata rispetto alla funzione di verosimiglianza binomiale e, dunque, le distribuzioni *full-conditional* non hanno una soluzione analitica, impedendo l'uso del Gibbs Sampling.

La distribuzione *Polya-Gamma* è stata introdotta da Polson, Scott e Windle (2012) per lo studio bayesiano di dati dicotomici. Si dimostra che, sotto la normalità a-priori per il vettore di parametri $\boldsymbol{\theta}$, aumentando il dataset con una variabile latente Polya-Gamma, $\pi(\omega_i) \sim \mathsf{PG}(0,1)$, $i = 1, \ldots, n$, la densità a-posteriori dei parametri associati alle covariate del modello segue a sua volta una distribuzione Normale.

Le ipotesi del modello sono

$$Y_i \sim \mathsf{Be}(\psi_i) \tag{2.10}$$

$$\psi_i = \frac{e^{\eta_i}}{1 + e^{\eta_i}} \tag{2.11}$$

$$\eta_i = \alpha + \mathbf{x}_i \boldsymbol{\beta}, \tag{2.12}$$

dove α è l'intercetta e $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)'$ è il vettore p-dimensionale di parametri da stimare.

Una proposta non informativa per $\boldsymbol{\theta} = (\alpha, \boldsymbol{\beta})$ è data da $\pi(\boldsymbol{\theta}) \propto 1$. Tuttavia, si assume $\alpha \sim \mathsf{N}(\mu_\alpha, \sigma^2_\alpha)$ e $\boldsymbol{\beta} \sim \mathsf{N}(\boldsymbol{\mu}_\beta, \boldsymbol{\Sigma}_\beta)$, e dunque $\pi(\boldsymbol{\theta}) \sim \mathsf{N}(\boldsymbol{\mu}_\theta, \boldsymbol{\Sigma}_\theta)$. Con riferimento a Polson, Scott e Windle (2012), aumentando il dataset con una variabile latente Polya-Gamma, $\pi(\omega_i) \sim \mathsf{PG}(1,0)$, $i = 1, \ldots, n$, la distribuzione condizionata del vettore $\boldsymbol{\theta}$ è una Normale Multivariata. Questa nuova tecnica di *data-augmentation* parte dall'uguaglianza

$$\frac{\left(e^{\psi}\right)^a}{\left(1 + e^{\psi}\right)^b} = 2^{-b} e^{k\psi} \int_0^\infty e^{-\omega\psi^2/2} p(\omega) d\omega \tag{2.13}$$

con $k = a - \dfrac{b}{2}$. La verosimiglianza è, quindi, esprimibile come:

$$
\begin{aligned}
\pi\left(\mathbf{y}, \boldsymbol{\omega} \mid \mathbf{X}, \boldsymbol{\theta}\right) &= \prod_{i=1}^{n} \mathbb{P}\left(Y_i = y_i \mid \mathbf{x}_i, \omega_i, \boldsymbol{\theta}\right) \pi\left(\omega_i\right) \\
&= \prod_{i=1}^{n} \left(\frac{e^{\alpha+\mathbf{x}_i\boldsymbol{\beta}}}{1+e^{\alpha+\mathbf{x}_i\boldsymbol{\beta}}}\right)^{y_i} \left(1 - \frac{e^{\alpha+\mathbf{x}_i\boldsymbol{\beta}}}{1+e^{\alpha+\mathbf{x}_i\boldsymbol{\beta}}}\right)^{1-y_i} \pi\left(\omega_i\right) \\
&= \prod_{i=1}^{n} \frac{\left(e^{\alpha+\mathbf{x}_i\boldsymbol{\beta}}\right)^{y_i}}{1+e^{\alpha+\mathbf{x}_i\boldsymbol{\beta}}} \pi\left(\omega_i\right) \\
&= \prod_{i=1}^{n} \frac{1}{2} \exp\left\{y_i\left(\alpha + \mathbf{x}_i\boldsymbol{\beta}\right) - \frac{\alpha + \mathbf{x}_i\boldsymbol{\beta}}{2}\right\} \\
&\qquad \times \exp\left\{-\frac{\omega_i\left(\alpha + \mathbf{x}_i\boldsymbol{\beta}\right)^2}{2}\right\} \pi\left(\omega_i\right) \\
&= \frac{1}{2^n} \exp\left\{\sum_{i=1}^{n} \tilde{y}_i\left(\alpha + \mathbf{x}_i\boldsymbol{\beta}\right)\right\} \\
&\qquad \times \exp\left\{-\sum_{i=1}^{n} \frac{\omega_i\left(\alpha + \mathbf{x}_i\boldsymbol{\beta}\right)^2}{2}\right\} \pi\left(\omega_i\right), \qquad (2.14)
\end{aligned}
$$

dove $\tilde{y}_i = y_i - \frac{1}{2}$, $\mathbf{X} = \left(\mathbf{x}_1', \ldots, \mathbf{x}_p'\right)'$ e $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_n)$. Dunque, la distribuzione condizionata di $\boldsymbol{\theta}$ è

$$
\begin{aligned}
\pi(\boldsymbol{\theta} \mid \mathbf{y}, \mathbf{X}, \boldsymbol{\omega}) &\propto \exp\left\{\sum_{i=1}^{n} \tilde{y}_i\left(\alpha + \mathbf{x}_i\boldsymbol{\beta}\right)\right\} \exp\left\{-\sum_{i=1}^{n} \frac{\omega_i\left(\alpha + \mathbf{x}_i\boldsymbol{\beta}\right)^2}{2}\right\} \pi(\boldsymbol{\theta}) \\
&\propto \exp\left\{-\frac{1}{2}\sum_{i=1}^{n} \omega_i\left(\alpha + \mathbf{x}_i\boldsymbol{\beta} - \frac{\tilde{y}_i}{\omega_i}\right)^2\right\} \pi(\boldsymbol{\theta}) \\
&\propto \exp\left\{-\frac{1}{2}(\tilde{\tilde{\mathbf{y}}} - \mathbf{X}_*\theta)'\mathbf{W}(\tilde{\tilde{\mathbf{y}}} - \mathbf{X}_*\theta)\right\} \pi(\boldsymbol{\theta}) \qquad (2.15)
\end{aligned}
$$

dove $\mathbf{X}_* = \begin{bmatrix}\mathbf{1}_n & \mathbf{X}\end{bmatrix}$, $\tilde{\tilde{\mathbf{y}}} = \left(\frac{\tilde{y}_1}{\omega_1}, \ldots, \frac{\tilde{y}_n}{\omega_n}\right)$, $\mathbf{W} = \mathrm{diag}(\omega_1, \ldots, \omega_n)$. L'equazione (2.15) è proporzionale al nucleo di una Normale Multivariata, di parametri $\boldsymbol{\mu}_\theta^*$ e $\boldsymbol{\Sigma}_\theta^*$, con

$$
\boldsymbol{\mu}_\theta^* = \boldsymbol{\Sigma}_\theta^*\left(\mathbf{X}_*'\mathbf{W}\tilde{\tilde{\mathbf{y}}} + \boldsymbol{\Sigma}_\theta^{-1}\boldsymbol{\mu}_\theta\right), \qquad (2.16)
$$

$$
\boldsymbol{\Sigma}_\theta^* = \left(\mathbf{X}_*'\mathbf{W}\mathbf{X}_* + \boldsymbol{\Sigma}_\theta^{-1}\right)^{-1}, \qquad (2.17)
$$

dove $\boldsymbol{\mu}_\theta$ e $\boldsymbol{\Sigma}_\theta$ sono rispettivamente i momenti primo e secondo della distribuzione a-priori di $\boldsymbol{\theta}$. Inoltre, la distribuzione condizionata di $\boldsymbol{\omega}$ segue a sua volta una densità Polya-Gamma, $\pi(\boldsymbol{\omega} \mid \mathbf{y}, \mathbf{X}_*, \boldsymbol{\theta}) \sim \mathsf{PG}(1, \mathbf{x}_*\boldsymbol{\theta})$ (Polson, Scott e Windle, 2012). Dunque, con l'introduzione della variabile latente $\boldsymbol{\omega}$ si è in possesso di una nuova tecnica bayesiana per lo studio di dati dicotomici che permette l'uso del Gibbs Sampling, in quanto le distribuzioni condizionate delle variabili $\boldsymbol{\theta}$ e $\boldsymbol{\omega}$ sono in forma chiusa.

Seguendo lo schema presentato in George e McCulloch (1997), viene formalizzato l'approccio Spike and Slab per dati dicotomici.

Sotto le assunzioni

$$\begin{aligned}
&\pi(\boldsymbol{\omega}) \sim \mathsf{PG}(1, 0),\\
&\pi(\boldsymbol{\theta} \mid \boldsymbol{\gamma}, \sigma^2) \sim \mathsf{N}(\boldsymbol{\mu}_\theta, \sigma^2 \mathbf{D}_\gamma),\\
&\pi(\boldsymbol{\gamma}) \sim \mathsf{Bin}(p, \phi),\\
&\pi(\sigma^2) \sim \mathsf{IG}(\lambda, v),
\end{aligned} \tag{2.18}$$

dove ϕ è la probabilità a-priori di inclusione delle variabili nel modello e $\mathbf{D}_\gamma = \sigma^2 \operatorname{diag}(b_1, \ldots, b_p)$, $b_j = (1-\gamma_j)v_0 + \gamma_j v_1$, indica indipendenza a-priori tra i regressori, la densità congiunta a-posteriori è

$$\begin{aligned}
\pi\left(\boldsymbol{\theta}, \boldsymbol{\gamma}, \sigma^2 \mid \mathbf{y}, \mathbf{X}_*, \boldsymbol{\omega}\right) &\propto \pi(\boldsymbol{\gamma})\pi(\boldsymbol{\theta})\pi(\sigma^2) \prod_{i=1}^{n} \mathbb{P}\left(Y_i = y_i \mid \mathbf{x}_i, \omega_i, \boldsymbol{\theta}\right) \pi\left(\omega_i\right)\\
&\propto \exp\left\{-\frac{1}{2}(\tilde{\tilde{\mathbf{y}}} - \mathbf{X}_*\boldsymbol{\theta})'\mathbf{W}(\tilde{\tilde{\mathbf{y}}} - \mathbf{X}_*\boldsymbol{\theta})\right\} \pi(\boldsymbol{\theta})\pi(\boldsymbol{\gamma})\pi(\sigma^2).
\end{aligned} \tag{2.19}$$

A questo punto è possibile scrivere le distribuzioni *full-conditional* di tutti i parametri considerati. Queste sono

$$\begin{aligned}
&\pi(\boldsymbol{\omega} \mid \mathbf{X}_*, \boldsymbol{\theta}) \sim \mathsf{PG}(1, \mathbf{X}_*\boldsymbol{\theta}),\\
&\pi(\boldsymbol{\theta} \mid \mathbf{y}, \mathbf{X}_*, \boldsymbol{\omega}, \boldsymbol{\gamma}, \sigma^2) \sim \mathsf{N}(\boldsymbol{\Sigma}^*_\theta(\mathbf{X}'_*\mathbf{W}\tilde{\tilde{\mathbf{y}}}), \boldsymbol{\Sigma}^*_\theta),\\
&\pi(\gamma_j \mid \pi_j, \boldsymbol{\theta}, \sigma^2) \sim \mathsf{Be}(\pi_j),\\
&\pi(\boldsymbol{\sigma^2} \mid \boldsymbol{\theta}, \boldsymbol{\gamma}) \sim \mathsf{IG}\left(\lambda + \frac{p}{2}, \nu + \frac{1}{2}\sum_{j=1}^{p} \frac{\theta_j^2}{b_j}\right).
\end{aligned} \tag{2.20}$$

L'algoritmo è descritto in 1

Algorithm 1 Spike and Slab Gibbs Sampling

Inizializzazione di $\boldsymbol{\pi}_0, \boldsymbol{\theta}_0$ e σ_0^2, simulando dalla distribuzione a-priori dei parametri;

for $b = 2$ to B **do**:
  for $j = 1$ to p **do**:
    $\gamma_j^{(b)} \mid \pi_j, \boldsymbol{\theta}, \sigma^2 \sim \mathsf{Be}(\hat{\pi}_j)$;
  end for
  Simulare dalle distribuzioni condizionate:
    $\hat{\boldsymbol{\omega}}^{(b)} \mid \mathbf{X}_*, \boldsymbol{\theta} \sim \mathsf{PG}(1, \mathbf{X}_* \boldsymbol{\theta}^{(b-1)})$;
    $\hat{\sigma}^2_{(b)} \mid \boldsymbol{\theta}, \boldsymbol{\gamma} \sim \mathsf{IG}\left(\lambda + \frac{p}{2}, \nu + \frac{1}{2}\sum_{j=1}^{p} \frac{\theta_j^2}{b_j}\right)$;
    $\hat{\boldsymbol{\theta}}^{(b)} \mid \mathbf{y}, \mathbf{X}_*, \boldsymbol{\omega}, \boldsymbol{\gamma}, \sigma^2 \sim \mathsf{N}(\boldsymbol{\Sigma}^*_\theta(\mathbf{X}'_* \mathbf{W} \tilde{\tilde{\mathbf{y}}}), \boldsymbol{\Sigma}^*_\theta)$;
  for $j = 1$ to p **do**
    aggiornamento $\hat{\pi}_j$;
  end for
end for

La dipendenza di γ_j da $\boldsymbol{\theta}$ e σ^2 è incorporata nel parametro π_j, relativo alla probabilità di inclusione del j-mo regressore nel modello. Questo viene aggiornato nell'ultimo passo dell'algoritmo seguendo due possibili proposte. Assumendo la seguente equazione,

$$p(\pi_j \mid \mathbf{y}, \mathbf{X}_*, \boldsymbol{\theta}, \sigma^2) = \frac{1}{1 + \dfrac{(1-\pi_j)}{\pi_j} g(\theta_j)}, \tag{2.21}$$

con $g(\theta_j)$ opportunamente specificata, la prima fa riferimento a George e McCulloch (1997), mentre la seconda è stata presentata per dati continui da Malsiner-Walli e Wagner (2016).

La prima proposta assume

$$g(\theta_j) = \frac{\pi(\gamma_j = 0, \sigma^2 \mid \mathbf{y}, \mathbf{X}_*, \boldsymbol{\omega})}{\pi(\gamma_j = 1, \sigma^2 \mid \mathbf{y}, \mathbf{X}_*, \boldsymbol{\omega})}, \tag{2.22}$$

dove

$$\begin{aligned}\pi(\boldsymbol{\gamma}, \sigma^2 \mid \mathbf{y}, \mathbf{X}_*, \boldsymbol{\omega}) &= \int_\theta p(\mathbf{y}, \boldsymbol{\omega}, \mathbf{X}_*, \boldsymbol{\theta}) \pi(\boldsymbol{\theta} \mid \boldsymbol{\gamma}, \sigma^2) \pi(\boldsymbol{\gamma}) \pi(\sigma^2) d\boldsymbol{\theta} \\ &\propto \pi(\boldsymbol{\gamma}) \pi(\sigma^2) \left| \mathbf{X}_*' \mathbf{W} \mathbf{X}_* + \mathbf{D}_\gamma^{-1} \right|^{-1/2} |\mathbf{D}_\gamma|^{-1/2} \\ &\quad \times \exp\left\{ \frac{1}{2} \tilde{\tilde{\mathbf{y}}}' \mathbf{W} \mathbf{X}_* \left(\mathbf{X}_*' \mathbf{W} \mathbf{X}_* + \mathbf{D}_\gamma^{-1} \right)^{-1} \mathbf{X}_*' \mathbf{W} \tilde{\tilde{\mathbf{y}}} \right\}. \end{aligned} \tag{2.23}$$

Ad ogni iterazione dell'algoritmo, per aggiornare il parametro π_j è necessario invertire la matrice $\boldsymbol{\Sigma}_\theta^*$, cosa che può essere fatta in maniera efficiente usando la decomposizione di Cholesky (Thisted, 1988). Per il calcolo del numeratore in $g(\theta_j)$ viene tolto il j-mo regressore dalla matrice $\mathbf{X}_*$; al contrario, questo viene inserito al denominatore. Un metodo efficiente per la computazione dell'inversione di una matrice in seguito all'inserimento/eliminazione di una riga è offerto da Chambers (1971).

La secondo proposta è pari a

$$g(\theta_j) = \frac{\pi_{spike}(\theta_j)}{\pi_{slab}(\theta_j)}, \tag{2.24}$$

alternativamente rappresentabile come

$$\frac{\pi(\theta_j | 0, \sigma^2 v_0)}{\pi(\theta_j | 0, \sigma^2 v_1)} = \sqrt{\frac{v_1}{v_0}} \exp\left\{ \frac{\theta_j^2}{2\sigma^2} \left(\frac{1}{v_1} - \frac{1}{v_0} \right) \right\}. \tag{2.25}$$

Questa è computazionalmente più efficiente della precedente, in quanto non sono necessari calcoli complessi per l'inversione della matrice di varianza-covarianza.

Capitolo 3

Reversible-Jump MCMC

Nel caso in cui il numero p di variabili sia molto elevato, il Gibbs Sampling presentato nel capitolo precedente non è computazionalmente efficiente, in quanto valuta l'importanza di ogni singola variabile ad ogni iterazione. Un metodo creato appositamente per la selezione di variabili in un dataset con $p \gg n$ è l'algoritmo *Reversible-Jump MCMC* (RJ), che permette salti trans-dimensionali per il vettore $\boldsymbol{\theta}$.

Sia $\mathbf{y} = (y_1, \ldots, y_n)$ il vettore risposta e $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{ip})$, $i = 1, \ldots, n$, un vettore p-dimensionale relativo alle covariate associate all'i-mo individuo. Il numero di modelli possibili è 2^p, i quali possono essere indicizzati con il vettore $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_p)$, con $\gamma_j = 0$ o 1 a seconda del fatto che il j-esimo regressore sia rispettivamente assente o presente nel modello. Il numero di variabili presenti nel generico modello è dato da $p_\gamma = \sum_{j=1}^{p} \gamma_j$. Le ipotesi sulle distribuzioni a-priori di $\boldsymbol{\theta}$ e $\boldsymbol{\gamma}$ sono della forma $\pi(\boldsymbol{\theta}_\gamma, \boldsymbol{\gamma}) = \pi(\boldsymbol{\theta} \mid \boldsymbol{\gamma})\pi(\boldsymbol{\gamma})$ e, generalmente, si assume

$$\pi(\boldsymbol{\theta} \mid \boldsymbol{\gamma}) \sim \mathsf{N}_{p_\gamma}(0, \boldsymbol{\Omega}_\gamma) \quad \text{e} \quad \pi(\boldsymbol{\gamma}) \sim \mathsf{Bin}(p, \phi). \tag{3.1}$$

La dipendenza di $\boldsymbol{\Omega}$ da γ deriva dal fatto che tale matrice è una matrice quadrata e simmetrica con p_γ righe e colonne; due scelte possono essere fatte per $\boldsymbol{\Omega}$: la prima ipotizza indipendenza a-priori ed è esprimibile come $\boldsymbol{\Omega}_\gamma = \mathbf{I}_{p_\gamma}$, mentre la seconda è data dalla *g-prior*, con $\boldsymbol{\Omega}_\gamma = c(\mathbf{X}_\gamma' \mathbf{X}_\gamma)^{-1}, c > 0$. Per quanto riguarda la distribuzione a-priori di $\boldsymbol{\gamma}$, questa segue una distribu-

zione Binomiale, dove ϕ è la probabilità d'inclusione di ogni variabile nel modello; ipotizzando $\phi = 0.5$ ogni modello ha la stessa probabilità di 2^{-p}. La distribuzione congiunta a-posteriori è, quindi, calcolabile nel seguente modo,

$$\pi(\boldsymbol{\gamma}, \boldsymbol{\theta} \mid \mathbf{y}, \mathbf{X}) \propto f(\mathbf{y} \mid \mathbf{X}, \boldsymbol{\theta}) \pi(\boldsymbol{\theta} \mid \boldsymbol{\gamma}) \pi(\boldsymbol{\gamma}) \tag{3.2}$$

ed è la distribuzione target della catena di Markov.

La costruzione della catena si basa sull'algoritmo Metropolis-Hastings. Ponendo $\boldsymbol{\delta} = (\boldsymbol{\gamma}, \boldsymbol{\theta}_\gamma)$, una scelta comune per $w(\boldsymbol{\delta}' \mid \boldsymbol{\delta})$ che assicura la condizione di reversibilità è data dagli *importance weights*. La transizione da $\boldsymbol{\delta}$ a $\boldsymbol{\delta}'$ prevede il passaggio dal modello $\boldsymbol{\gamma}$ con p_γ variabili al modello $\boldsymbol{\gamma}'$ con $p_{\gamma'}$ variabili, $p_{\gamma'} \neq p_\gamma$: siamo, dunque, in presenza di un salto transdimensionale per il parametro $\boldsymbol{\theta}$.

3.1 *Between Models* moves

Il salto trans-dimensionale si basa sul concetto di "*dimension matching*". Si ipotizza di voler passare dallo stato $(\boldsymbol{\gamma}, \boldsymbol{\theta})$ allo stato $(\boldsymbol{\gamma}', \boldsymbol{\theta}')$, in cui $p_{\gamma'} > p_\gamma$: viene introdotto un vettore casuale $\mathbf{u}$ con densità $h_{\gamma \to \gamma'}(\mathbf{u})$ e di lunghezza $(p_{\gamma'} - p_\gamma)$, che permette di pareggiare la dimensionalità di $\boldsymbol{\gamma}'$; lo stato corrente del vettore $\boldsymbol{\theta}_\gamma$ e il vettore $\mathbf{u}$ vengono quindi mappati in maniera biettiva con il nuovo stato $\boldsymbol{\theta}_{\gamma'} = g_{\gamma \to \gamma'}(\boldsymbol{\theta}_\gamma, \mathbf{u})$. Poichè la probabilità di accettazione deve tenere in considerazione la differenza tra le due dimensioni, viene introdotta la matrice jacobiana della funzione g, e la probabilità di accettazione diventa

$$\alpha(\boldsymbol{\delta}, \boldsymbol{\delta}') = \min \left\{ 1, \frac{\pi(\boldsymbol{\delta}' \mid \mathbf{y}, \mathbf{X}) q(\boldsymbol{\gamma} \mid \boldsymbol{\gamma}')}{\pi(\boldsymbol{\delta} \mid \mathbf{y}, \mathbf{X}) q(\boldsymbol{\gamma}' \mid \boldsymbol{\gamma}) h_{\gamma \to \gamma'}(\mathbf{u})} \left| \frac{\partial g_{\gamma \to \gamma'}(\boldsymbol{\theta}_\gamma, \mathbf{u})}{\partial (\boldsymbol{\theta}_\gamma, \mathbf{u})} \right| \right\} \tag{3.3}$$

dove l'ultimo termine indica, appunto, lo jacobiano. Chiaramente, l'arbitrarietà della scelta per le funzioni $h_{\gamma \to \gamma'}(\mathbf{u})$ e $g_{\gamma \to \gamma'}(\boldsymbol{\theta}_\gamma, \mathbf{u})$ può influenzare le prestazioni della catena.

Più in generale, si possono rilassare le ipotesi sulle dimensioni di $\mathbf{u}$ e assumere la lunghezza di tale vettore pari a n_γ; in questo caso la dimensione in due stati della catena viene pareggiata se $n_\gamma + p_\gamma = n_{\gamma'} + p_{\gamma'}$. Tuttavia, sotto tali ipotesi, viene meno l'esistenza di una funzione inversa calcolabile in modo deterministico, ed è necessario generare un vettore $\mathbf{u}'$ da una densità

$h_{\gamma'\to\gamma}(\mathbf{u}')$. La mossa all'indietro è, dunque, $\boldsymbol{\theta}_\gamma = g_{\gamma'\to\gamma}(\boldsymbol{\theta}_{\gamma'}, \mathbf{u}')$. La probabilità di accettazione è, dunque, pari a

$$\alpha(\boldsymbol{\delta}, \boldsymbol{\delta}') = \min\left\{1, \frac{\pi(\boldsymbol{\delta}' \mid \mathbf{y}, \mathbf{X}) q(\boldsymbol{\gamma} \mid \boldsymbol{\gamma}') h_{\gamma'\to\gamma}(\mathbf{u}')}{\pi(\boldsymbol{\delta} \mid \mathbf{y}, \mathbf{X}) q(\boldsymbol{\gamma}' \mid \boldsymbol{\gamma}) h_{\gamma\to\gamma'}(\mathbf{u})} \left|\frac{\partial g_{\gamma\to\gamma'}(\boldsymbol{\theta}_\gamma, \mathbf{u})}{\partial(\boldsymbol{\theta}_\gamma, \mathbf{u})}\right|\right\}. \quad (3.4)$$

Un esempio pratico è presente in Green (1995), Brooks (1998) e Fan e Sisson (2010): si consideri il caso in cui si voglia passare da un modello M_1, con parametro scalare $\theta_1 \in \mathbb{R}^1$, al modello M_2, con vettore di parametri $\boldsymbol{\theta}_2 \in \mathbb{R}^2$. Viene generato uno scalare u per pareggiare le dimensioni nei due modelli e una scelta possibile per la proposta $\boldsymbol{\theta}_2$ è data da $\theta_2^{(1)} = \theta_1 + u$ e $\theta_2^{(1)} = \theta_1 - u$. La funzione per il passo inverso è calcolata in maniera deterministica, e risulta $\theta_1 = \frac{1}{2}(\theta_2^{(1)} + \theta_2^{(2)})$.

Sono presenti in letteratura vari metodi per la scelta di $h_{\gamma\to\gamma'}(\mathbf{u})$, tra i più famosi si citano:

i) *Marginalizzazione e Data-augmentation*: sotto particolari assunzioni per le densità dei vari parametri, come ad esempio l'assunzione di distribuzioni a-priori coniugate con la verosimiglianza del modello o l'introduzione di variabili latenti, è possibile integrare il vettore di parametri $\boldsymbol{\theta}_\gamma$ dalla distribuzione a posteriori, riducendo l'algoritmo ad una catena di dimensione fissa (George e McCulloch (1997)). Tale metodo permette di evitare la scelta arbitraria di $h_{\gamma\to\gamma'}(\mathbf{u})$ e $g_{\gamma\to\gamma'}(\boldsymbol{\theta}_\gamma, \mathbf{u})$ ed è efficiente dal punto di vista computazionale, in quanto considera la distribuzione del signolo parametro $\boldsymbol{\gamma}$;

ii) *Centering & order methods*: questo metodo è stato proposto da Brooks, Giudici e Roberts (2003) e permette di determinare il parametro di scala che regola la densità $h_{\gamma\to\gamma'}(\mathbf{u})$ in maniera automatica. La funzione $c_{\gamma\to\gamma'}(\boldsymbol{\theta}_\gamma) = g_{\gamma\to\gamma'}(\boldsymbol{\theta}_\gamma, \mathbf{u})$ rappresenta il *"centering point"* per la nuova proposta $(\boldsymbol{\gamma}', \boldsymbol{\theta}_{\gamma'})$, ed è definita tale per cui la verosimiglianza calcolata nel nuovo stato è uguale alla verosimiglianza calcolata nello stato precedente per una qualche particolare scelta di $\mathbf{u}$, $L(\mathbf{y} \mid \boldsymbol{\gamma}, \boldsymbol{\theta}_\gamma) = L(\mathbf{y} \mid \boldsymbol{\gamma}', c_{\gamma\to\gamma'}(\boldsymbol{\theta}_\gamma))$. Se il parametro di scala è un valore scalare, lo si può scegliere tale per cui la probabilità di accettazione

$\alpha((\boldsymbol{\gamma}, \boldsymbol{\theta}_\gamma), (\boldsymbol{\gamma}', c_{\gamma\to\gamma'}(\boldsymbol{\theta}_\gamma)))$ è esattamente uno (0^{th}-*order method*), mentre nel caso in cui $h_{\gamma\to\gamma'}(\mathbf{u})$ sia un vettore k-dimensionale, il metodo più generale è detto k^{th}-*order method*, e pone le k derivate rispetto a $\mathbf{u}$ della probabilità di accettazione, calcolate nel *"centering point"*, pari a zero

$$\nabla^k \alpha((\boldsymbol{\gamma}, \boldsymbol{\theta}_\gamma), (\boldsymbol{\gamma}', c_{\gamma\to\gamma'}(\boldsymbol{\theta}_\gamma))) = 0. \tag{3.5}$$

A partire dalle k derivate è, quindi, possibile calcolare il vettore di parametri associati a $h_{\gamma\to\gamma'}(\mathbf{u})$ in forma deterministica. La probabilità di accettazione α diventa piatta attorno a $c_{\gamma\to\gamma'}(\boldsymbol{\theta}_\gamma)$ e permette di esplorare punti distanti dallo stato corrente, permettendo un'esplorazione esaustiva dello spazio di $\boldsymbol{\gamma}$. L'espressione (3.5) è sensibile alla scelta di $g_{\gamma\to\gamma'}(\boldsymbol{\theta}_\gamma, \mathbf{u})$, tuttavia una scelta non perfetta è compensata da una stima ottimale di $\mathbf{u}$.

Un esempio di 0^{th}-*order method* è mostrato in Brooks, Giudici e Roberts (2003), i quali considerano un modello AR con k componenti, del tipo

$$\mathbf{X}_t = \sum_{j=1}^{k} a_j \mathbf{X}_{t-j} + \boldsymbol{\varepsilon}_t, \tag{3.6}$$

dove $\varepsilon_t \sim \mathsf{N}(0, \sigma_\varepsilon^2)$, $a_j \sim \mathsf{N}(0, \sigma_a^2)$, $k \sim U[1, k_{max}]$. Viene assunta, inoltre, una distribuzione Gamma-Inversa per σ_ε^2. Si consideri il passaggio dal modello $\boldsymbol{\gamma}$ con k componenti al modello $\boldsymbol{\gamma}'$ con $k' = k + 1$ componenti: definendo la funzione $g_{\gamma\to\gamma'}(\boldsymbol{\theta}_\gamma, u) = (\boldsymbol{\theta}_\gamma, \sigma u)$, con $h_{\gamma\to\gamma'}(\mathbf{u}) \sim \mathsf{N}(0, 1)$, il "*centering point*" tale per cui $L(\mathbf{y} \mid \boldsymbol{\gamma}, \boldsymbol{\theta}_\gamma) = L(\mathbf{y} \mid \boldsymbol{\gamma}', c_{\gamma\to\gamma'}(\boldsymbol{\theta}_\gamma))$ è dato da $c_{\gamma\to\gamma'}(\boldsymbol{\theta}_\gamma) = (\boldsymbol{\theta}_\gamma, 0)$. Lo jacobiano della funzione $g_{\gamma\to\gamma'}$ è σ, dunque la probabilità di accettazione è

$$\begin{aligned} \alpha((\boldsymbol{\gamma}, \boldsymbol{\theta}_\gamma), (\boldsymbol{\gamma}', (\boldsymbol{\theta}_\gamma, 0))) &= \frac{\pi(\boldsymbol{\gamma}', (\boldsymbol{\theta}_\gamma, 0) \mid \mathbf{X}) q(\boldsymbol{\gamma} \mid \boldsymbol{\gamma}') \sigma}{\pi(\boldsymbol{\gamma}, \boldsymbol{\theta}_\gamma \mid \mathbf{X}) q(\boldsymbol{\gamma}' \mid \boldsymbol{\gamma}) h_{\gamma\to\gamma'}(0)} \\ &= \frac{(\sigma_a^2)^{-1/2} q(\boldsymbol{\gamma} \mid \boldsymbol{\gamma}') \sigma}{q(\boldsymbol{\gamma}' \mid \boldsymbol{\gamma})}. \end{aligned} \tag{3.7}$$

Risolvendo l'equazione $\alpha((\boldsymbol{\gamma}, \boldsymbol{\theta}_\gamma), (\boldsymbol{\gamma}', (\boldsymbol{\theta}_\gamma, 0))) = 1$ si arriva alla stima per il parametro di scala,

$$\sigma^2 = \sigma_a^2 \left(\frac{q(\boldsymbol{\gamma}' \mid \boldsymbol{\gamma})}{q(\boldsymbol{\gamma} \mid \boldsymbol{\gamma}')} \right)^2, \tag{3.8}$$

che dipende solo da σ_a, $\boldsymbol{\gamma}$ e $\boldsymbol{\gamma}'$, e non da $\boldsymbol{\theta}_\gamma$ o $\mathbf{X}$;

iii) *Generic Sampler*: un metodo per minimizzare l'arbitrarietà della scelta di $g_{\gamma\to\gamma'}(\boldsymbol{\theta}_\gamma, \mathbf{u})$ è presente in letteratura in Green (2003). Dato il modello $\boldsymbol{\gamma}$, si supponga di conoscere i primi due momenti, $\boldsymbol{\mu}_\gamma$ e $\boldsymbol{\Sigma}_\gamma = \mathbf{B}_\gamma \mathbf{B}_\gamma^T$, del vettore di parametri $\boldsymbol{\theta}_\gamma$, dove $\mathbf{B}_\gamma$ è una matrice $p_\gamma \times p_\gamma$ derivante dalla decomposizione di Cholesky. La proposta per il set di parametri $\boldsymbol{\theta}_{\gamma'}$ associato al nuovo stato $\boldsymbol{\gamma}'$ della catena è

$$\boldsymbol{\theta}_{\gamma'} = \begin{cases} \boldsymbol{\mu}_{\gamma'} + \mathbf{B}_{\gamma'}[\mathbf{R}\mathbf{B}_{\gamma'}^{-1}(\boldsymbol{\theta}_\gamma - \boldsymbol{\mu}_\gamma)]_1^{p_{\gamma'}}, & \text{se } p_{\gamma'} < p_\gamma, \\ \boldsymbol{\mu}_{\gamma'} + \mathbf{B}_{\gamma'}\mathbf{R}\mathbf{B}_{\gamma'}^{-1}(\boldsymbol{\theta}_\gamma - \boldsymbol{\mu}_\gamma), & \text{se } p_{\gamma'} = p_\gamma, \\ \boldsymbol{\mu}_{\gamma'} + \mathbf{B}_{\gamma'}\mathbf{R}\begin{pmatrix} \mathbf{B}_{\gamma'}^{-1}(\boldsymbol{\theta}_\gamma - \boldsymbol{\mu}_\gamma) \\ \mathbf{u} \end{pmatrix}, & \text{se } p_{\gamma'} > p_\gamma, \end{cases} \tag{3.9}$$

dove $[\cdot]_1^m$ rappresenta i primi m elementi del vettore, $\mathbf{R}$ è una matrice ortogonale di ordine $max(p_\gamma, p_{\gamma'})$ e $\mathbf{u} \sim h_{\gamma\to\gamma'}(\mathbf{u})$ è un vettore di dimensione $(p_{\gamma'} - p_\gamma)$. Il vettore $\mathbf{u}$ entra in gioco solo quando si passa ad un modello con un numero maggiore di variabili o nel calcolo della probabilità di accettazione per la mossa inversa dal modello $\boldsymbol{\gamma}'$ al modello $\boldsymbol{\gamma}$ quando $p_\gamma > p_{\gamma'}$. La proposta per $\boldsymbol{\theta}_{\gamma'}$ è deterministica quando $p_{\gamma'} < p_\gamma$. Ricordando che $\boldsymbol{\delta} = (\boldsymbol{\gamma}, \boldsymbol{\theta})$, la probabilità di accettazione diventa

$$\alpha(\boldsymbol{\delta}, \boldsymbol{\delta'}) = \frac{\pi(\boldsymbol{\delta'} \mid \mathbf{y}, \mathbf{X}) q(\boldsymbol{\gamma} \mid \boldsymbol{\gamma}')}{\pi(\boldsymbol{\delta} \mid \mathbf{y}, \mathbf{X}) q(\boldsymbol{\gamma} \mid \boldsymbol{\gamma}'))} \frac{|\mathbf{B}_{\gamma'}|}{|\mathbf{B}_\gamma|} \begin{cases} h_{\gamma\to\gamma'}(\mathbf{u}), & \text{se } p_{\gamma'} < p_\gamma, \\ 1, & \text{se } p_{\gamma'} = p_\gamma, \\ 1/h_{\gamma\to\gamma'}(\mathbf{u}), & \text{se } p_{\gamma'} > p_\gamma. \end{cases} \tag{3.10}$$

Un esempio di *Generic Sampler* è rappresentato in Papathomas, Dellaportas e Vasdekis (2009), nel quale gli autori studiano il caso in cui il vettore $\boldsymbol{\theta}_{\gamma'}$ si distribuisce secondo una Normale Multivariata, come nel modello lineare, $\boldsymbol{\theta}_{\gamma'} \sim \mathsf{N}(\boldsymbol{\mu}_{\gamma'|\theta_\gamma}, \boldsymbol{\Sigma}_{\gamma'|\theta_\gamma})$. Partendo dalle ipotesi

$$\mathbf{y} = \mathbf{X}_\gamma \boldsymbol{\theta}_\gamma + \boldsymbol{\varepsilon}, \qquad \boldsymbol{\varepsilon} \sim \mathsf{N}(0, \mathbf{V}), \tag{3.11}$$

le stime dei parametri per la proposta $\boldsymbol{\theta}_{\gamma'}$ sono

$$\begin{aligned}\boldsymbol{\mu}_{\gamma'|\theta_\gamma} =& (\mathbf{X}_{\gamma'}^T\mathbf{V}^{-1}\mathbf{X}_{\gamma'})^{-1}\mathbf{X}_{\gamma'}^T\mathbf{V}^{-1}\left[\mathbf{y} + \mathbf{B}^{-1}\mathbf{V}^{-1/2}\left(\mathbf{X}_\gamma\boldsymbol{\theta}_\gamma - \boldsymbol{P}_\gamma\mathbf{y}\right)\right] \\ \boldsymbol{\Sigma}_{\gamma'|\theta_\gamma} =& \boldsymbol{Q}_{\gamma',\gamma'} - \boldsymbol{Q}_{\gamma',\gamma'}\boldsymbol{Q}_{\gamma',\gamma}^{-1}\boldsymbol{Q}_{\gamma,\gamma}\boldsymbol{Q}_{\gamma,\gamma'}^{-1}\boldsymbol{Q}_{\gamma',\gamma'} + c\mathbf{I}_{p_{\gamma'}}\end{aligned} \tag{3.12}$$

dove $\mathbf{B} = \left(\mathbf{V} + \mathbf{X}_{\gamma'}\boldsymbol{\Sigma}_{\gamma'|\theta_\gamma}\mathbf{X}_{\gamma'}^T\right)^{-1/2}$, $\boldsymbol{P}_\gamma = \mathbf{X}_\gamma(\mathbf{X}_\gamma^T\mathbf{V}^{-1}\mathbf{X}_\gamma)^{-1}\mathbf{X}_\gamma^T\mathbf{V}^{-1}$, $\boldsymbol{Q}_{\gamma,\gamma'} = (\mathbf{X}_\gamma^T\mathbf{V}^{-1}\mathbf{X}_{\gamma'})^{-1}$ e $c > 0$. Le stime dei momenti derivate sono quelle di massima verosimiglianza per il modello $\boldsymbol{\gamma}'$ più una correzione basata sulla discrepanza tra $\boldsymbol{\theta}_\gamma$ e la moda a-posteriori del modello $\boldsymbol{\gamma}$. La verosimiglianza calcolata nei due stati della catena, $\boldsymbol{\theta}_\gamma$ e $\boldsymbol{\theta}_{\gamma'}$, di conseguenza, assume valori non distanti tra loro. La funzione che mappa i due stati è

$$\boldsymbol{\theta}_{\gamma'} = \boldsymbol{\mu}_{\gamma'|\theta_\gamma} + \boldsymbol{\Sigma}_{\gamma'|\theta_\gamma}^{1/2}\mathbf{u}, \tag{3.13}$$

dove $\mathbf{u} \sim \mathsf{N}(0, \mathbf{I}_{p_{\gamma'}})$. Lo jacobiano della funzione $g_{\gamma\to\gamma'}(\boldsymbol{\theta}_\gamma, \mathbf{u})$ è dato da $\left|\boldsymbol{\Sigma}_{\gamma'|\theta_\gamma}^{1/2}\right|\left|\boldsymbol{\Sigma}_{\gamma|\theta_{\gamma'}}^{1/2}\right|$: in questo modo la costante c diventa un parametro di regolazione per la probabilità di accettazione. Lo svantaggio principale di tale metodo, tuttavia, è l'inversione di matrici che in alcuni casi possono essere di grandi dimensioni, perdendo il confronto dal punto di vista computazionale se paragonato ad altri metodi.

3.2 Reversible-Jump MCMC Marginalizzato

Lo scopo di questo lavoro è quello di proporre un nuovo algoritmo per la selezione di variabili in presenza di dati binari nel caso in cui $p \gg n$. L'idea è quella di sfruttare la distribuzione marginale di $\boldsymbol{\gamma}$ e σ^2 a-posteriori, definita in (2.23), come funzione target della catena di Markov.

L'inserimento della variabile latente $\pi(\omega_i) \sim \mathsf{PG}(0, 1)$ permette l'integrazione del vettore dei parametri $\boldsymbol{\theta}$ dalla distribuzione a-posteriori $\pi(\boldsymbol{\theta}, \boldsymbol{\gamma}, \sigma^2 \mid \mathbf{y}, \mathbf{X}, \boldsymbol{\omega})$, risolvendo il problema del "*dimension matching*". Poichè tutti gli stati della catena hanno la stessa dimensione $\mathbb{R}_\gamma^p \times \mathbb{R}_{\sigma^2}$, viene evitato l'uso arbitrario delle funzioni $h_{\gamma\to\gamma'}(\mathbf{u})$ e $g_{\gamma\to\gamma'}(\boldsymbol{\theta}_\gamma, \boldsymbol{\gamma})$.

Sia $\mathbf{y} = (y_1, \ldots, y_n)$ il vettore risposta e $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{ip})$, $i = 1, \ldots, n$, il vettore di variabili associate all'i-mo individuo. Sotto le ipotesi (2.18), è possibile fare riferimento all'algoritmo MH per la selezione di

variabili: la probabilità di accettazione diventa

$$\alpha\left(\boldsymbol{\gamma}, \boldsymbol{\gamma}'\right) = \min\left\{1, \frac{q(\boldsymbol{\gamma} \mid \boldsymbol{\gamma}')\pi(\boldsymbol{\gamma}', \sigma^2 \mid \mathbf{y}, \mathbf{X}_{\gamma'}, \boldsymbol{\omega})}{q(\boldsymbol{\gamma}' \mid \boldsymbol{\gamma})\pi(\boldsymbol{\gamma}, \sigma^2 \mid \mathbf{y}, \mathbf{X}_{\gamma}, \boldsymbol{\omega})}\right\}, \tag{3.14}$$

dove $\mathbf{X}_\gamma$ indica la matrice dei dati contenente solo le p_γ variabili inserite nel modello $\boldsymbol{\gamma}$. Il valore del nuovo stato della catena di $\boldsymbol{\gamma}$ dipende dalla scelta di $q(\boldsymbol{\gamma}' \mid \boldsymbol{\gamma})$, per la quale si fa riferimento a Lamnisos, Griffin e Steel (2009): nell'articolo gli autori prendono in considerazione la funzione simmetrica in $\boldsymbol{\gamma}$ e $\boldsymbol{\gamma}'$

$$q(\boldsymbol{\gamma}' \mid \boldsymbol{\gamma}) = \frac{1}{p}, \qquad \text{se } \sum_{j=1}^{p} \left|\gamma_j' - \gamma_j\right| = 1. \tag{3.15}$$

In questa maniera vengono presi in considerazione tutti i modelli con $(p_\gamma + 1)$ e $(p_\gamma - 1)$ variabili e, condizionatamente al valore corrente della catena, la densità $q(\boldsymbol{\gamma}' \mid \boldsymbol{\gamma})$ è uguale per tutti i $\boldsymbol{\gamma}'$ ammissibili. La probabilità di accettazione diventa pari al rapporto della distribuzione target calcolata in $\boldsymbol{\gamma}$ e $\boldsymbol{\gamma}'$, ed è pari a

$$\alpha\left(\boldsymbol{\gamma}, \boldsymbol{\gamma}'\right) = \min\left\{1, \frac{\pi(\boldsymbol{\gamma}', \sigma^2 \mid \mathbf{y}, \mathbf{X}_{\gamma'}, \boldsymbol{\omega})}{\pi(\boldsymbol{\gamma}, \sigma^2 \mid \mathbf{y}, \mathbf{X}_{\gamma}, \boldsymbol{\omega})}\right\}, \tag{3.16}$$

che dipende solo dall'informazione a-priori sul parametro ϕ, relativo alla probabilità di inclusione delle variabili, e dalla funzione target (2.23).

La cardinalità dell'insieme dei modelli con un numero generico k di variabili è dato da $\binom{p}{k}$: il numero di modelli con k variabili raggiunge il massimo quando $k = p/2$, portando la catena ad esplorare un insieme maggiore di stati e migliorando le capacità dell'algoritmo di saltare da un modello all'altro. L'iperparametro ϕ che regola la probabilità di inclusione a-priori delle variabili è, dunque, trattabile come un parametro di regolazione per la probabilità di accettazione.

Una volta ottenuta la probabilità di accettazione α e, avendo a disposizione tutte le distribuzione *full-conditional* dei parametri $\boldsymbol{\theta}_\gamma$, σ^2 e $\boldsymbol{\omega}$, è possibile definire l'algoritmo *RJ Marginalizzato* presente in 2, dove $\hat{\boldsymbol{\Sigma}}_\theta = (\mathbf{X}_\gamma' \mathbf{W} \mathbf{X}_\gamma + \frac{1}{\sigma^2}\boldsymbol{\Omega}^{-1})^{-1}$, $\boldsymbol{\Omega}$ è la matrice a-priori di varianza e covarianza per $\boldsymbol{\theta}$ e $\mathbf{W} = \text{diag}(\omega_1, \ldots, \omega_n)$. In 2 i regressori sono assunti indipendenti a-priori.

Algorithm 2 RJ Marginalizzato

Inizializzazione di $\boldsymbol{\gamma}_0, \sigma_0^2$, e $\boldsymbol{\omega}_0$, simulando dalla distribuzione a-priori dei parametri;

for $b = 2$ to B **do**:

Generare un valore $j \in \{1, \ldots, p\}$ con probabilità $1/p$:

if $\gamma_j^{(b-1)} = 0$ **then** $\gamma_j^{'} = 1$;

else $\gamma_j^{'} = 0$;

end if

Calcolare

$$\alpha\left(\boldsymbol{\gamma}^{(b-1)}, \boldsymbol{\gamma}'\right) = \min\left\{1, \frac{\pi(\boldsymbol{\gamma}', \sigma^2_{(b-1)} \mid \mathbf{y}, \mathbf{X}_{\gamma'}, \boldsymbol{\omega}^{(b-1)})}{\pi(\boldsymbol{\gamma}^{(b-1)}, \sigma^2_{(b-1)} \mid \mathbf{y}, \mathbf{X}_{\gamma}^{(b-1)}, \boldsymbol{\omega}^{(b-1)})}\right\};$$

Con probabilità $\alpha\left(\boldsymbol{\gamma}^{(b-1)}, \boldsymbol{\gamma}'\right)$:

Porre $\hat{\boldsymbol{\gamma}}^{(b)} = \boldsymbol{\gamma}'$;

Simulare dalle distribuzioni condizionate:

$\hat{\boldsymbol{\theta}}^{(b)} \mid \mathbf{y}, \mathbf{X}_{\gamma^{(b)}}, \boldsymbol{\gamma}^{(b)}, \sigma^2_{(b-1)}, \boldsymbol{\omega}^{(b-1)} \sim \mathsf{N}(\boldsymbol{\Sigma}^*_\theta(\mathbf{X}'_{\gamma^{(b)}}\mathbf{W}\tilde{\tilde{\mathbf{y}}}), \boldsymbol{\Sigma}^*_\theta)$;

$\hat{\sigma}^2_{(b)} \mid \boldsymbol{\theta}^{(b)}, \boldsymbol{\gamma}^{(b)} \sim \mathsf{IG}\left(\lambda + \frac{p_\gamma}{2}, \nu + \frac{1}{2}\sum_{j=1}^{p_\gamma}\theta_j^2\right)$;

$\hat{\boldsymbol{\omega}}^{(b)} \mid \mathbf{X}_{\gamma^{(b)}}, \boldsymbol{\theta}^{(b)}, \boldsymbol{\gamma}^{(b)} \sim \mathsf{PG}(1, \mathbf{X}_{\gamma^{(b)}}\boldsymbol{\theta}^{(b)})$;

Definire, altrimenti:

$\hat{\boldsymbol{\gamma}}^{(b)} = \boldsymbol{\gamma}^{(b-1)}, \qquad \hat{\boldsymbol{\theta}}^{(b)} = \boldsymbol{\gamma}^{(b-1)}$;

$\hat{\sigma}^2_{(b)} = \sigma^2_{(b-1)}, \qquad \hat{\boldsymbol{\omega}}^{(b)} = \boldsymbol{\omega}^{(b-1)}$.

end for

Capitolo 4

Multiple-Try MCMC

Nel capitolo 1 si fa accenno all'algoritmo Metropolis-Hastings per generare valori da una distribuzione a-posteriori del tipo (1.3): ad ogni iterazione un nuovo valore $\boldsymbol{\theta}'$ viene accettato con probabilità $\alpha(\boldsymbol{\theta}, \boldsymbol{\theta}')$. Un'estensione di questo metodo, in grado di valutare più proposte per il parametro di interesse, è il *Multiple-Try Metropolis-Hastings* (MT): ad ogni iterazione vengono presi in considerazione k valori candidati $\boldsymbol{\theta}^{(i)} \sim q(\boldsymbol{\theta}^{(i)} \mid \boldsymbol{\theta})$, per $i = 1, \ldots, k$, e il nuovo stato della catena $\boldsymbol{\theta}'$ viene scelto tra questi e accettato con probabilità $\alpha(\boldsymbol{\theta}, \boldsymbol{\theta}')$. Assumendo la densità per le proposte $q(\boldsymbol{\theta}^{(i)} \mid \boldsymbol{\theta})$ tale per cui il nuovo stato dipende dallo stato precedente, è necessario generare $(k-1)$ valori ausiliari $\mathbf{v}^{(i)} \sim q(\boldsymbol{\theta} \mid \boldsymbol{\theta}')$, $i = 1, \ldots, k$ e $i \neq j : \boldsymbol{\theta}^{(j)} = \boldsymbol{\theta}'$, per poter raggiungere la proprietà di reversibilità della catena. Una funzione di probabilità di accettazione generalizzata che assicura l'ergodicità e l'invarianza della funzione target è

$$\alpha(\boldsymbol{\theta}, \boldsymbol{\theta}') = \min\left(1, \frac{\sum_{i=1}^{k} w(\boldsymbol{\theta}^{(i)} \mid \boldsymbol{\theta})}{\sum_{i=1}^{k} w(\mathbf{v}^{(i)} \mid \boldsymbol{\theta}')}\right), \tag{4.1}$$

dove $\mathbf{v}^{(j)} = \boldsymbol{\theta}$ (Martino, 2018). E' di uso comune stimare gli *importance weights* per il calcolo dei pesi, ma altre forme del tipo (1.8) sono ammissibili.

Ogni iterazione richiede la valutazione di $2k-1$ stati, di cui $k-1$ entrano in gioco solo per il calcolo della probabilità di accettazione. Se la densità delle proposte $\boldsymbol{\theta}^{(i)}$ è indipendente dalla stato precedente, $q(\boldsymbol{\theta}^{(i)} \mid \boldsymbol{\theta}) = q(\boldsymbol{\theta}^{(i)})$, è possibile riciclare i $(k-1)$ stati $\boldsymbol{\theta}^{(i)}$ scartati nel passo in avanti. Questi

e i valori ausiliari necessari per il calcolo di (4.1) provengono dalla stessa distribuzione; senza intaccare l'ergodicità della catena si può porre

$$\begin{aligned} &\mathbf{v}^{(k)} = \boldsymbol{\theta}, \\ &\mathbf{v}^{(i)} = \boldsymbol{\theta}^{(i)}, \quad per \quad i = 1, \dots, k-1, \end{aligned} \tag{4.2}$$

(Martino, 2018). La probabilità di accettazione si riduce a

$$\alpha(\boldsymbol{\theta}, \boldsymbol{\theta}^{(j)}) = \min\left(1, \frac{w(\boldsymbol{\theta}^{(j)}) + \sum_{i=1, i\neq j}^{k} w(\boldsymbol{\theta}^{(i)})}{w(\boldsymbol{\theta}) + \sum_{i=1}^{k-1} w(\boldsymbol{\theta}^{(i)})}\right). \tag{4.3}$$

L'efficienza di questo algoritmo dipende, oltre che da $q(\boldsymbol{\theta}^{(i)} \mid \boldsymbol{\theta})$ e di $\pi(\boldsymbol{\theta})$, anche dal numero di proposte k valutate ad ogni iterazione.

Algorithm 3 MT

Inizializzazione di $\boldsymbol{\theta}_0$, simulando dalla distribuzione a-priori del parametro;

for $b = 2$ to B **do**:

Generare $\boldsymbol{\theta}^{(1)}, \dots, \boldsymbol{\theta}^{(k)} \sim q(\boldsymbol{\theta}^{(i)} \mid \boldsymbol{\theta}^{(b-1)})$;

Calcolare $w(\boldsymbol{\theta}^{(i)} \mid \boldsymbol{\theta}^{(b-1)})$, $i = 1, \dots, k$, e scegliere $\boldsymbol{\theta}^{(j)}$ secondo la densità di probabilità

$$\bar{w}_k = \frac{w(\boldsymbol{\theta}^{(j)} \mid \boldsymbol{\theta}^{(b-1)})}{\sum_{i=1}^{k} w(\boldsymbol{\theta}^{(i)} \mid \boldsymbol{\theta}^{(b-1)})};$$

Generare $\mathbf{v}^{(1)}, \dots, \mathbf{v}^{(j-1)}, \mathbf{v}^{(j+1)}, \dots, \mathbf{v}^{(k)} \sim q(\mathbf{v}^{(i)} \mid \boldsymbol{\theta}^{(j)})$ e porre $\mathbf{v}^{(j)} = \boldsymbol{\theta}^{(b-1)}$;

Calcolare $w(\mathbf{v}^{(i)} \mid \boldsymbol{\theta}^{(j)}), i = 1, \dots, k$;

Porre $\boldsymbol{\theta}^{(b)} = \boldsymbol{\theta}^{(j)}$ con probabilità

$$\alpha(\boldsymbol{\theta}^{(b-1)}, \boldsymbol{\theta}^{(j)}) = \min\left(1, \frac{\sum_{i=1}^{k} w(\boldsymbol{\theta}^{(i)} \mid \boldsymbol{\theta}^{(b-1)})}{\sum_{i=1}^{k} w(\mathbf{v}^{(i)} \mid \boldsymbol{\theta}^{(j)})}\right),$$

$\boldsymbol{\theta}^{(b)} = \boldsymbol{\theta}^{(b-1)}$ altrimenti;

end for

4.1 Generalized Multiple-Try MH

Una generalizzazione del metodo Multiple-Try (GMT) è stata proposta da Pandolfi, Bartolucci e Friel (2010). Tale metodo permette di rilassare le ipotesi sui pesi e sfruttare una forma diversa da (1.8). Insieme ad un'approssimazione quadratica della funzione target, è possibile mostrare che, a differenza dell'algoritmo MT, per il calcolo della probabilità di accettazione $\alpha(\boldsymbol{\theta}, \boldsymbol{\theta}')$ non è necessaria la valutazione della distribuzione target in ognuno dei k valori proposti.

Data una funzione per i pesi $w^*(\boldsymbol{\theta}^{(i)} \mid \boldsymbol{\theta})$, il valore $\boldsymbol{\theta}'$ viene scelto tra i k proposti in base alla funzione di probabilità

$$p_{\boldsymbol{\theta}'} = \frac{w^*(\boldsymbol{\theta}' \mid \boldsymbol{\theta})}{\sum_{i=1}^{k} w^*(\boldsymbol{\theta}^{(i)} \mid \boldsymbol{\theta})}, \tag{4.4}$$

e la probabilità di accettazione è definita come

$$\alpha(\boldsymbol{\theta}, \boldsymbol{\theta}') = \min\left\{1, \frac{\pi(\boldsymbol{\theta}')q(\boldsymbol{\theta} \mid \boldsymbol{\theta}')p_{\boldsymbol{\theta}}}{\pi(\boldsymbol{\theta})q(\boldsymbol{\theta}' \mid \boldsymbol{\theta})p_{\boldsymbol{\theta}'}}\right\}. \tag{4.5}$$

E' possibile dimostrare che la funzione (4.5) soddisfa la condizione di reversibilità (Pandolfi, Bartolucci e Friel, 2010). L'algoritmo è descritto in 4.

Per mostrare che l'algoritmo MT è un caso speciale dell'algoritmo GMT, si può riscrivere la funzione di probabilità di accettazione in (4.5) nel seguente modo

$$\alpha = \min\left\{1, \frac{\pi(\boldsymbol{\theta}^{(j)})q(\boldsymbol{\theta}^{(b-1)} \mid \boldsymbol{\theta}^{(j)})w^*(\boldsymbol{\theta}^{(b-1)} \mid \boldsymbol{\theta}^{(j)})\sum_{i=1}^{k} w^*(\boldsymbol{\theta}^{(i)} \mid \boldsymbol{\theta}^{(b-1)})}{\pi(\boldsymbol{\theta}^{(b-1)})q(\boldsymbol{\theta}^{(j)} \mid \boldsymbol{\theta}^{(b-1)})w^*(\boldsymbol{\theta}^{(j)} \mid \boldsymbol{\theta}^{(b-1)})\sum_{i=1}^{k} w^*(\mathbf{v}^{(i)} \mid \boldsymbol{\theta}^{(j)})}\right\}. \tag{4.6}$$

Usando i pesi specificati in (1.8), se si scelgono gli *importance weights*, definiti con $\xi(\boldsymbol{\theta}, \boldsymbol{\theta}') = (q(\boldsymbol{\theta} \mid \boldsymbol{\theta}')q(\boldsymbol{\theta}' \mid \boldsymbol{\theta}))^{-1}$, o $\xi(\boldsymbol{\theta}, \boldsymbol{\theta}') = 1$, l'algoritmo Multiple-Try e la sua generalizzazione coincidono.

In Pandolfi, Bartolucci e Friel (2010), gli autori propongono un'approssimazione quadratica della funzione target, definita come

$$\begin{aligned} \pi^*(\boldsymbol{\theta}') &= \pi(\boldsymbol{\theta})\mathbf{A}(\boldsymbol{\theta}, \boldsymbol{\theta}') \\ \mathbf{A}(\boldsymbol{\theta}, \boldsymbol{\theta}') &= e^{\mathbf{s}(\boldsymbol{\theta}')'(\boldsymbol{\theta}'-\boldsymbol{\theta}) + \frac{1}{2}(\boldsymbol{\theta}'-\boldsymbol{\theta})'\mathbf{D}(\boldsymbol{\theta}')(\boldsymbol{\theta}'-\boldsymbol{\theta})} \end{aligned} \tag{4.7}$$

Algorithm 4 GMT

Inizializzazione di $\boldsymbol{\theta}_0$, simulando dalla distribuzione a-priori del parametro;

for $b = 2$ to B **do**:

Generare $\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(k)} \sim q(\boldsymbol{\theta}^{(i)} \mid \boldsymbol{\theta}^{(b-1)})$;

Calcolare $w^*(\boldsymbol{\theta}^{(i)} \mid \boldsymbol{\theta}^{(b-1)}), \quad i = 1, \ldots, k$;

Scegliere $\boldsymbol{\theta}^{(j)}$ secondo la densità di probabilità

$$p_{\boldsymbol{\theta}^{(j)}} = \frac{w^*(\boldsymbol{\theta}^{(j)} \mid \boldsymbol{\theta}^{(b-1)})}{\sum_{i=1}^{k} w^*(\boldsymbol{\theta}^{(i)} \mid \boldsymbol{\theta}^{(b-1)})};$$

Generare $\mathbf{v}^{(1)}, \ldots, \mathbf{v}^{(j-1)}, \mathbf{v}^{(j+1)}, \ldots, \mathbf{v}^{(k)} \sim q(\mathbf{v}^{(i)} \mid \boldsymbol{\theta}^{(i)})$ e porre $\mathbf{v}^{(j)} = \boldsymbol{\theta}^{(b-1)}$;

Definire

$$p_{\boldsymbol{\theta}^{(b-1)}} = \frac{w^*(\boldsymbol{\theta}^{(b-1)} \mid \boldsymbol{\theta}^{(j)})}{\sum_{i=1}^{k} w^*(\mathbf{v}^{(i)} \mid \boldsymbol{\theta}^{(j)})};$$

Porre $\boldsymbol{\theta}^{(b)} = \boldsymbol{\theta}^{(j)}$ con probabilità

$$\alpha(\boldsymbol{\theta}^{(b-1)}, \boldsymbol{\theta}^{(j)}) = \min\left\{1, \frac{\pi(\boldsymbol{\theta}^{(j)}) q(\boldsymbol{\theta}^{(b-1)} \mid \boldsymbol{\theta}^{(j)}) p_{\boldsymbol{\theta}^{(b-1)}}}{\pi(\boldsymbol{\theta}^{(b-1)}) q(\boldsymbol{\theta}^{(j)} \mid \boldsymbol{\theta}^{(b-1)}) p_{\boldsymbol{\theta}^{(j)}}}\right\},$$

$\boldsymbol{\theta}^{(b)} = \boldsymbol{\theta}^{(b-1)}$ altrimenti;

end for

dove $\mathbf{s}(\boldsymbol{\theta})$ e $\mathbf{D}(\boldsymbol{\theta})$ sono, rispettivamente, le derivate prima e seconda della funzione $log(\pi(\boldsymbol{\theta}))$ rispetto a $\boldsymbol{\theta}$. Se si usano i pesi *importance weights* costruiti a partire dalla distribuzione approssimata $\pi^*(\boldsymbol{\theta})$, la funzione di probabilità per la scelta di $\boldsymbol{\theta}^{(j)}$ diventa

$$p_{\boldsymbol{\theta}^{(j)}} = \frac{\mathbf{A}(\boldsymbol{\theta}^{(b-1)}, \boldsymbol{\theta}^{(j)})}{q(\boldsymbol{\theta}^{(j)} \mid \boldsymbol{\theta}^{(b-1)})} \Big/ \sum_{i=1}^{k} \frac{\mathbf{A}(\boldsymbol{\theta}^{(b-1)}, \boldsymbol{\theta}^{(i)})}{q(\boldsymbol{\theta}^{(i)} \mid \boldsymbol{\theta}^{(b-1)})}. \tag{4.8}$$

Sostituendo (4.8) in (4.5) si dimostra che la probabilità di accettazione non dipende da $\pi(\boldsymbol{\theta})$, evitando il calcolo della funzione target per ogni modello proposto.

4.1.1 Generalized Multiple-Try Reversible-Jump MCMC

L'estensione GMT dell'algoritmo MH è applicabile anche all'algoritmo Reversible-Jump (GMT-RJ). Ad ogni passo un insieme k di nuovi modelli, insieme ai relativi set di parametri $\boldsymbol{\theta}^{(i)}_{\gamma^{(i)}}$, $i = 1, \ldots, k$, viene proposto.

Si supponga di essere nello stato della catena $\boldsymbol{\delta} = (\boldsymbol{\gamma}, \boldsymbol{\theta}_\gamma)$ e di voler valutare il passaggio al modello $\boldsymbol{\gamma}'$ con vettore di parametri $\boldsymbol{\theta}_{\gamma'}$. Data una funzione per i pesi generalizzati w^*, la funzione $q(\boldsymbol{\gamma}' \mid \boldsymbol{\gamma})$ per le proposte e la densità $h_{\gamma \to \gamma'}(\mathbf{u})$, l'algoritmo è descritto in 5 e si rimanda a Pandolfi, Bartolucci e Friel (2010) per la dimostrazione della condizione di reversibilità.

Algorithm 5 GMT-RJ

Inizializzazione di $\boldsymbol{\theta}_0$, simulando dalla distribuzione a-priori del parametro;

for $b = 2$ to B **do**:

Proporre $\boldsymbol{\gamma}^{(1)}, \ldots, \boldsymbol{\gamma}^{(k)} \sim q(\boldsymbol{\gamma}^{(i)} \mid \boldsymbol{\gamma}^{(b-1)})$ e generare $\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(k)}$;

Calcolare $w^*(\boldsymbol{\delta}^{(i)} \mid \boldsymbol{\delta}^{(b-1)})$, $i = 1, \ldots, k$;

Scegliere $\boldsymbol{\gamma}^{(j)}$ secondo la densità di probabilità

$$p_{\boldsymbol{\gamma}^{(j)}} = \frac{w^*(\boldsymbol{\delta}^{(j)} \mid \boldsymbol{\delta}^{(b-1)})}{\sum_{i=1}^{k} w^*(\boldsymbol{\delta}^{(i)} \mid \boldsymbol{\delta}^{(b-1)})};$$

Proporre $\mathbf{v}^{(1)}, \ldots, \mathbf{v}^{(j-1)}, \mathbf{v}^{(j+1)}, \ldots, \mathbf{v}^{(k)} \sim q(\mathbf{v}^{(i)} \mid \boldsymbol{\gamma}^{(i)})$ e porre $\mathbf{v}^{(j)} = \boldsymbol{\gamma}^{(b-1)}$;

Definire

$$p_{\boldsymbol{\gamma}^{(b-1)}} = \frac{w^*(\boldsymbol{\delta}^{(b-1)} \mid \boldsymbol{\delta}^{(j)})}{\sum_{i=1}^{k} w^*(\mathbf{v}^{(i)} \mid \boldsymbol{\delta}^{(j)})};$$

Porre $\boldsymbol{\delta}^{(b)} = \boldsymbol{\delta}^{(j)}$ con probabilità

$$\alpha = \min \left\{ 1, \frac{\pi(\boldsymbol{\delta}^{(j)} \mid \mathbf{X}) q(\boldsymbol{\gamma}^{(b-1)} \mid \boldsymbol{\gamma}^{(j)}) p_{\boldsymbol{\gamma}^{(b-1)}}}{\pi(\boldsymbol{\delta}^{(b-1)} \mid \mathbf{X}) q(\boldsymbol{\gamma}^{(j)} \mid \boldsymbol{\gamma}^{(b-1)}) p_{\boldsymbol{\gamma}^{(j)}} h_{\gamma^{(b-1)} \to \gamma^{(j)}}(\mathbf{u})} \left| \mathbf{J}((\boldsymbol{\theta}^{(b-1)}, \mathbf{u}), \boldsymbol{\theta}^{(j)}) \right| \right\},$$

$\boldsymbol{\delta}^{(b)} = \boldsymbol{\delta}^{(b-1)}$ altrimenti;

end for

4.2 Multiple-Try Reversible-Jump MCMC Marginalizzato

L'ultimo metodo rivisitato in questa tesi riguarda l'algoritmo Multiple-Try Reversible-Jump MCMC (MT-RJ), adattato al caso di dati binari con variabile latente Polya-Gamma e utile per la selezione di variabili quando $p \gg n$, in quanto permette al modello di esplorare maggiormente la distribuzione a-posteriori (2.23), con un capacità migliore di uscire da mode locali.

Le ipotesi sono le stesse di (2.18): il dataset viene aumentato con una variabile latente $\pi(\omega_i) \sim \mathsf{PG}(1,0)$, mentre i parametri $\boldsymbol{\theta}$ seguono una distribuzione a-priori $\pi(\boldsymbol{\theta}) \sim \mathsf{N}(\boldsymbol{\mu}_\theta, \boldsymbol{\Sigma}_\theta)$ e sono stati integrati dalla funzione target, che dipende solo da $\boldsymbol{\gamma}$ e σ^2, i quali si distribuiscono a-priori rispettivamente come una variabile $\mathsf{Bin}(p,\phi)$ e $\mathsf{IG}(\lambda, v)$. Inoltre, la stessa densità per i modelli proposti $q(\boldsymbol{\gamma}' \mid \boldsymbol{\gamma}) = 1/p$, se $\sum_{j=1}^{p} |\gamma - \gamma'| = 1$, viene scelta. Questa associa la stessa probabilità a tutti i modelli candidati e i pesi $w(\boldsymbol{\gamma}^{(i)} \mid \boldsymbol{\gamma})$ diventano pari alla distribuzione a-posteriori calcolata in $\boldsymbol{\gamma}^{(i)}$ e σ^2.

La probabilità di accettazione è della forma (4.1), ed è

$$\alpha(\boldsymbol{\gamma}, \boldsymbol{\gamma}') = \min\left\{1, \frac{\sum_{i=1}^{k} w(\boldsymbol{\gamma}^{(i)} \mid \boldsymbol{\gamma})}{\sum_{i=1}^{k} w(\mathbf{v}^{(i)} \mid \boldsymbol{\gamma}')}\right\}, \tag{4.9}$$

dove $\mathbf{v}^{(i)}$ sono i valori ausiliari generati a partire da $q(\mathbf{v}^{(i)} \mid \boldsymbol{\gamma}')$ per poter garantire la reversibilità dell'algoritmo.

L'algoritmo MT-RJ Marginalizzato, a parità di lunghezza della catena, ha la caratteristica di esplorare un numero maggiore di modelli se confrontato con l'algoritmo 2, aumentandone la flessibilità e l'efficienza. Ad ogni iterazione aumenta la probabilità di inserire un regressore importante per la costruzione del modello, diminuendo il numero di variabili non influenti inserite. Tuttavia, la stima a posteriori dei parametri $\boldsymbol{\theta}$ e σ^2 è soggetta a maggior variabilità, a causa di un maggior numero di salti trans-dimensionali dovuti al parametro k, il quale ha un effetto sulla probabilità di accettazione di un nuovo stato della catena.

L'algoritmo è descritto in 6, dove $\boldsymbol{\Sigma}_\theta^* = (\mathbf{X}_\gamma' \mathbf{W} \mathbf{X}_\gamma + \frac{1}{\sigma^2}\boldsymbol{\Omega}^{-1})^{-1}$ e $\boldsymbol{\Omega}$ è la matrice a-priori di varianza e covarianza per $\boldsymbol{\theta}$ e $\mathbf{W} = \mathrm{diag}(\omega_1, \ldots, \omega_n)$. I regressori sono assunti indipendenti a-priori.

Algorithm 6 MT-RJ Marginalizzato

Inizializzazione di $\boldsymbol{\gamma}_0, \sigma_0^2$ e $\boldsymbol{\omega}_0$, simulando dalla distribuzione a-priori dei parametri;

for $b = 2$ to B **do**:

for $i = 1$ to k **do**:

$\boldsymbol{\gamma}^{(i)} = \boldsymbol{\gamma}^{(b-1)}$;

Generare un valore $j \in \{1, \ldots, p\}$ con probabilità $1/p$:

if $\gamma_j^{(b-1)} = 0$ **then** $\gamma_j^{(i)} = 1$;

else $\gamma_j^{(i)} = 0$;

end if

Calcolare $w(\boldsymbol{\gamma}^{(i)} \mid \boldsymbol{\gamma}^{(b-1)}) = \pi(\boldsymbol{\gamma}^{(b-1)}, \sigma^2_{(b-1)} \mid \mathbf{y}, \mathbf{X}_{\gamma^{(i)}}, \boldsymbol{\omega}^{(b-1)})$;

end for

Porre $\boldsymbol{\gamma}^{(j)} = \boldsymbol{\gamma}'$ secondo la densità di probabilità

$$\bar{w}_k = \frac{w(\boldsymbol{\gamma}^{(j)} \mid \boldsymbol{\gamma}^{(b-1)})}{\sum_{i=1}^{k} w(\boldsymbol{\gamma}^{(i)} \mid \boldsymbol{\gamma}^{(b-1)})};$$

Generare $\mathbf{v}^{(1)}, \ldots, \mathbf{v}^{(j-1)}, \mathbf{v}^{(j+1)}, \ldots, \mathbf{v}^{(k)} \sim q(\mathbf{v}^{(i)} \mid \boldsymbol{\gamma}^{(j)})$ e porre $\mathbf{v}^{(j)} = \boldsymbol{\gamma}^{(b-1)}$;

Calcolare $w(\mathbf{v}^{(i)} \mid \boldsymbol{\gamma}^{(j)}), i = 1, \ldots, k$;

Calcolare

$$\alpha\left(\boldsymbol{\gamma}^{(b-1)}, \boldsymbol{\gamma}^{(j)}\right) = \min\left\{1, \frac{\sum_{i=1}^{k} w(\boldsymbol{\gamma}^{(i)} \mid \boldsymbol{\gamma}^{(b-1)})}{\sum_{i=1}^{k} w(\mathbf{v}^{(i)} \mid \boldsymbol{\gamma}^{(j)})}\right\}; \tag{4.10}$$

Con probabilità $\alpha\left(\boldsymbol{\gamma}^{(b-1)}, \boldsymbol{\gamma}^{(j)}\right)$:

Porre $\hat{\boldsymbol{\gamma}}^{(b)} = \boldsymbol{\gamma}^{(j)}$;

Simulare dalle distribuzioni condizionate:

$\hat{\boldsymbol{\theta}}^{(b)} \mid \mathbf{y}, \mathbf{X}_{\gamma^{(b)}}, \boldsymbol{\gamma}^{(b)}, \sigma^2_{(b-1)}, \boldsymbol{\omega}^{(b-1)} \sim \mathsf{N}(\boldsymbol{\Sigma}^*_\theta(\mathbf{X}'_{\gamma^{(b)}} \mathbf{W} \tilde{\tilde{\mathbf{y}}}), \boldsymbol{\Sigma}^*_\theta)$;

$\hat{\sigma}^2_{(b)} \mid \boldsymbol{\theta}^{(b)}, \boldsymbol{\gamma}^{(b)} \sim \mathsf{IG}\left(\lambda + \frac{p_\gamma}{2}, \nu + \frac{1}{2}\sum_{j=1}^{p_\gamma} \theta_j^2\right)$;

$\hat{\boldsymbol{\omega}}^{(b)} \mid \mathbf{X}_{\gamma^{(b)}}, \boldsymbol{\theta}^{(b)}, \boldsymbol{\gamma}^{(b)} \sim \mathsf{PG}(1, \mathbf{X}_{\gamma^{(b)}} \boldsymbol{\theta}^{(b)})$;

Definire, altrimenti:

$\hat{\boldsymbol{\gamma}}^{(b)} = \boldsymbol{\gamma}^{(b-1)}, \qquad \hat{\boldsymbol{\theta}}^{(b)} = \boldsymbol{\gamma}^{(b-1)}$;

$\hat{\sigma}^2_{(b)} = \sigma^2_{(b-1)}, \qquad \hat{\boldsymbol{\omega}}^{(b)} = \boldsymbol{\omega}^{(b-1)}$.

end for

Nel capitolo 6 verranno analizzate e formalizzate alcune estensioni possibili dell'algoritmo MT-RJ Marginalizzato.

Capitolo 5

Esempi numerici

Per la validazione degli algoritmi presentati nei capitoli precedenti, questi vengono testati su dati reali e dati simulati. Sono analizzati due casi, $p < n$ e $p > n$.

5.1 Dataset simulati

Sia $\boldsymbol{\gamma}^* = (\gamma_1^*, \ldots, \gamma_p^*)$ il modello generatore dei dati che include la j-esima variabile se $\gamma_j^* = 1$, $j = 1, \ldots, p$. La variabile risposta $\mathbf{Y}$ è generata nel seguente modo:

$$\begin{aligned} \mathbf{Z} &= \sum_{i=1}^{n} \mathbf{x}_i \boldsymbol{\beta} + \boldsymbol{\varepsilon} \\ \boldsymbol{\pi} &= 1/(1 + e^{-\mathbf{Z}}) \\ \mathbb{P}(\mathbf{Y} = 1) &= \boldsymbol{\pi}, \quad \mathbb{P}(\mathbf{Y} = 0) = 1 - \boldsymbol{\pi}, \end{aligned} \tag{5.1}$$

dove i β_j, $j = 1, \ldots, p$, sono scelti arbitrariamente simulando da una distribuzione uniforme $\mathsf{U}[2, 5]$, mentre $x_{ij} \sim \mathsf{N}(0, 1)$, $\forall\, ij$.

L'output dell'algoritmo è una matrice Γ, di dimensioni $B \times p$, dove B è il numero di iterazioni compiute. La riga b-esima della matrice rappresenta il modello visitato all'iterazione b, $\boldsymbol{\gamma}^{(b)}$.

La dimensionalità di $\boldsymbol{\gamma}$ non permette una semplice rappresentazione grafica dei risultati della catena. Per la visualizzazione della distribuzione a-posteriori di tale parametro, viene introdotto il vettore per l'enumerazione dei modelli $\mathbf{R}$, che mappa biunivocamente il modello $\boldsymbol{\gamma}^{(b)} \in \mathbb{R}^p$ e lo scalare $r^{(b)}$ secondo la formula

$$r^{(b)} = 1 + \boldsymbol{\gamma}^{(b)}\mathbf{d}, \tag{5.2}$$

dove $\mathbf{d}$ è il vettore $[2^0\ 2^1\ \ldots\ 2^{p-1}]'$ (Brooks, Giudici e Philippe, 2003).

Indicando con $\boldsymbol{\gamma}^i$ uno dei 2^p possibili modelli, la probabilità $\mathbb{P}(\boldsymbol{\gamma} = \boldsymbol{\gamma}^i)$ viene stimata all'interno della catena, valutando quante volte l'algoritmo ha visitato il modello i,

$$\mathbb{P}(\boldsymbol{\gamma} = \boldsymbol{\gamma}^i) = \frac{1}{B}\sum_{b=1}^{B}\mathbf{I}(\boldsymbol{\gamma}^{(b)} = \boldsymbol{\gamma}^i), \tag{5.3}$$

per $i = 1, \ldots, 2^p$.

Le probabilità a-posteriori di inserimento delle variabili nel modello sono

$$\hat{\pi}_j = \sum_{b=1}^{B}\mathbf{I}(\gamma_j^{(b)} = 1)/B, \quad j = 1, \ldots, p; \tag{5.4}$$

a partire da queste e dal vero modello $\boldsymbol{\gamma}^*$ è possibile costruire la curva ROC (Receiver Operating Characteristic) per valutare la bontà della selezione dei regressori e per il confronto tra gli algoritmi. Questi vengono inseriti nel modello nel caso in cui la loro probabilità di inclusione a-posteriori sia maggiore di una determinata soglia. La curva ROC riporta sull'asse delle ascisse la quantità *"1-Specificità"* al variare di tale soglia, mentre l'asse delle ordinate rappresenta la relativa *"Sensibilità"*.

A partire dalla tabella di confusione sottostante,

Tabella 5.1: Tabella di confusione per la selezione delle variabili.

	$\hat{\boldsymbol{\gamma}} = 0$	$\hat{\boldsymbol{\gamma}} = 1$
$\boldsymbol{\gamma}^* = 0$	a	b
$\boldsymbol{\gamma}^* = 1$	c	d

dove $\hat{\boldsymbol{\gamma}}$ indica il modello stimato, è possibile valutare la Specificità e la Sensibilità della selezione delle variabili degli algoritmi, definite come

$$\text{Specificità} = \frac{a}{a+b}, \qquad \text{Sensibilità} = \frac{c}{c+d}. \tag{5.5}$$

La prima è relativa alla corretta esclusione dei regressori non significativi dal modello, mentre la seconda è associata alla loro corretta inclusione quando questi appartengono a $\boldsymbol{\gamma}^*$.

Per la scelta del cut-off ottimale viene preso in considerazione l'indice di Youden (Baker e Kramer, 2007), che massimizza la distanza tra la bisettrice e la curva.

Un metodo per valutare la bontà della curva ROC è dato dall'*"Area Sotto la Curva"* (*AUC*): un valore pari a 1 di tale statistica è associato ad una perfetta previsione delle variabili da inserire nel modello, al contrario, un valore pari a 0.5 indica l'inserimento casuale di ogni variabile.

5.1.1 Caso 1: $p < n$

Il primo dataset è composto da 200 osservazioni e 60 variabili esplicative, alle quali va aggiunta l'intercetta, inserita nel modello con probabilità 1. Il modello generatore dei dati, $\boldsymbol{\gamma}^*$, è composto da 25 variabili. La frequenza della classe $\mathbf{Y} = 1$ è del 14%. Il Gibbs Sampling è costruito su 10000 iterazioni con un periodo di burn-in pari a 3000, mentre sono state generate catene da 20000 iterazioni per gli algoritmi RJ e MT-RJ, con burn-in pari a 5000.

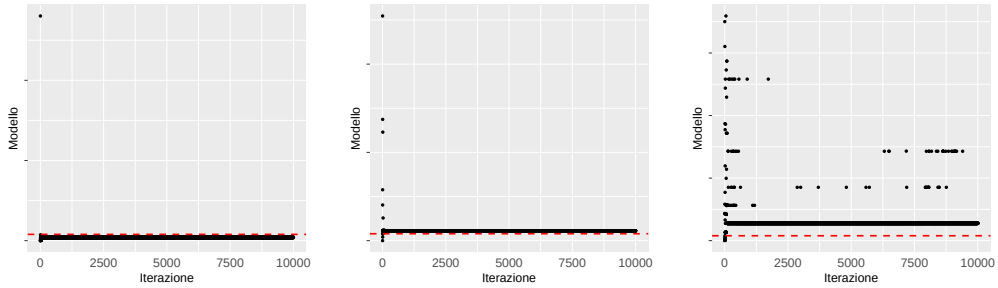
Per valutare il comportamento degli algoritmi al variare del parametro ϕ, questo è posto pari a 0.1, 0.3 e 0.5. Inoltre, per quanto riguarda il numero di proposte candidate nell'algoritmo MT-RJ, vengono analizzati i casi $k = 10$ e $k = 30$. Le componenti del vettore $\boldsymbol{\theta}$ sono assunte indipendenti tra di loro e le ipotesi sulla distribuzione a-priori del parametro sono

$$\pi(\boldsymbol{\theta}) \sim \mathsf{N}(0, \sigma^2 \mathbf{I}_p). \tag{5.6}$$

In modo tale da evitare una variabilità elevata dei valori assunti dai θ_j, $j = 1, \ldots, 60$, una distribuzione a-priori $\mathsf{IG}(2, 1)$ è scelta per σ^2. Infine, i parametri v_0 e v_1 relativi alle varianze delle priori Spike and Slab sono posti rispettivamente pari a 0.1 e 100.

I risultati del Gibbs Sampling sono incoraggianti per la proposta (2.23), mentre la seconda proposta (2.25) non ha portato risultati degni di nota. Verrà presentato, dunque, solo il primo dei due metodi.

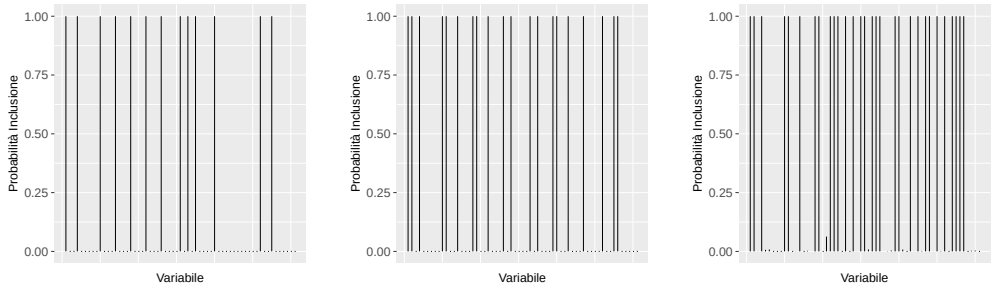
Usando la codifica (5.2), i modelli visitati dall'algoritmo sono presentati, ai vari livelli di ϕ, in Figura 5.1, dove la linea rossa indica il vero modello, $\boldsymbol{\gamma}^*$. Dopo un periodo di burn-in, l'algoritmo converge verso un modello quando $\phi = 0.1$ e $\phi = 0.3$. Tale convergenza non è mostrata quando $\phi = 0.5$, con ulteriori salti tra modelli anche dopo una fase di assestamento.



(a) $GS : \phi = 0.1$ **(b)** $GS : \phi = 0.3$ **(c)** $GS : \phi = 0.5$

Figura 5.1: Modelli visitati dall'algortimo GS, al variare di ϕ.

Le probabilità a-posteriori di inserimento nel modello delle variabili sono mostrate in Figura 5.2. L'algoritmo seleziona un numero sempre maggiore di variabili all'aumentare del parametro ϕ, con probabilità vicine a 0 o a 1.



(a) $GS : \phi = 0.1$ **(b)** $GS : \phi = 0.3$ **(c)** $GS : \phi = 0.5$

Figura 5.2: Probabilità post burn-in di inclusione delle variabili per l'algoritmo GS, al variare di ϕ.

La tabella 5.2 mostra la tabella di confusione per la selezione delle variabili tra il modello stimato, $\hat{\boldsymbol{\gamma}}$, e il modello vero, $\boldsymbol{\gamma}^*$: vengono inserite le variabili tali

per cui la probabilità a-posteriori di inserimento nel modello π_j, $j = 1, \ldots, p$, sia maggiore della soglia scelta secondo il criterio di Youden. Quando $\phi = 0.1$, l'algoritmo include le variabili nel modello stimato con una Sensibilità pari al 46%. Il tasso di errata classificazione è minimo per $\phi = 0.3$ (16%) e si alza quando $\phi = 0.5$ (23%): in quest'ultimo caso, tuttavia, al contrario di quanto succede quando $\phi = 0.1$, la Sensibilità del modello è dell'88%, mentre la Specificità si abbassa dal 97% al 71%.

Sono stati analizzati i vari modelli stimati: tutte le variabili inserite nel modello con $\phi = 0.1$ vengono inserite anche nel caso in cui $\phi = 0.3$, e, a loro volta, le covariate inserite nel modello con $\phi = 0.3$ fanno parte anche del modello stimato con $\phi = 0.5$.

		$\phi = 0.1$		$\phi = 0.3$		$\phi = 0.5$	
		$\hat{\boldsymbol{\gamma}} = 0$	$\hat{\boldsymbol{\gamma}} = 1$	$\hat{\boldsymbol{\gamma}} = 0$	$\hat{\boldsymbol{\gamma}} = 1$	$\hat{\boldsymbol{\gamma}} = 0$	$\hat{\boldsymbol{\gamma}} = 1$
GS	$\boldsymbol{\gamma}^* = 0$	34	1	33	2	25	10
	$\boldsymbol{\gamma}^* = 1$	14	12	8	18	3	23

Tabella 5.2: Risultati della selezione dei regressori per l'algoritmo GS, al variare di ϕ.

Per valutare la bontà dell'algoritmo viene presa in considerazione la curva ROC, la quale è presentata per i tre casi in Figura 5.3. La statistica AUC assume un valore sistematicamente superiore a 0.70. Il caso $\phi = 0.3$ riporta un valore pari a 0.82, 10 punti percentuali in più se confrontato con il caso $\phi = 0.1$, che presenta un'AUC del 72%. Il caso $\phi = 0.5$ restituisce un valore dell'AUC pari a 0.80.

La Figura 5.4 evidenzia il comportamento della media dei parametri all'aumentare delle iterazioni. La rappresentazione grafica di tutti i parametri θ_j, $j = 1, \ldots, 60$, richiede troppo spazio; ci si limita, dunque, alla visualizzazione di solo tre di questi. Gli algoritmi riportano, ai diversi livelli di ϕ, stime diverse per i parametri. La convergenza di questi è soddisfatta quando $\phi = 0.5$, mentre la media di θ_j, $j = 4, 18, 19$, è soggetta a maggiore variabilità quando $\phi = 0.1$ e $\phi = 0.3$. Un aumento della probabilità a-priori ϕ causa un

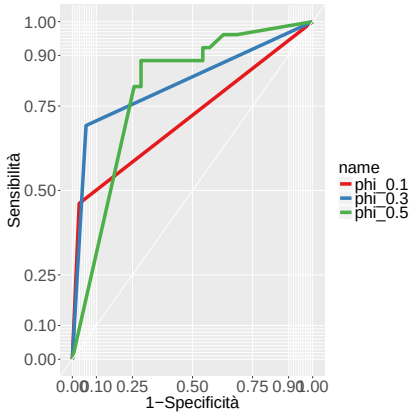
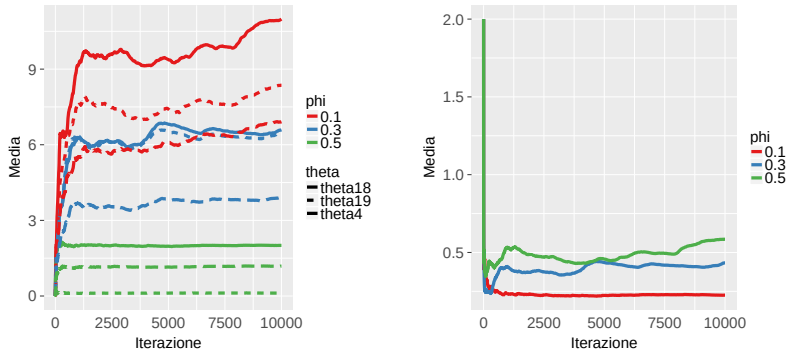


Figura 5.3: Curva ROC per le prestazioni dell'algoritmo GS, al variare di ϕ.

incremento nella stima dell'iperparametro σ^2, poichè si verifica un maggior numero di salti trans-dimensionali.



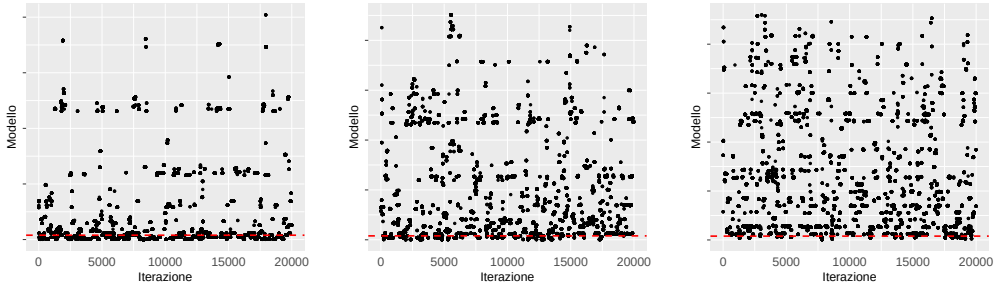
(a) $\theta_j, j = 4, 18, 19$ **(b)** σ^2

Figura 5.4: Comportamento della media dei parametri $\boldsymbol{\theta}$ e σ^2 nell'algoritmo GS, al variare di ϕ.

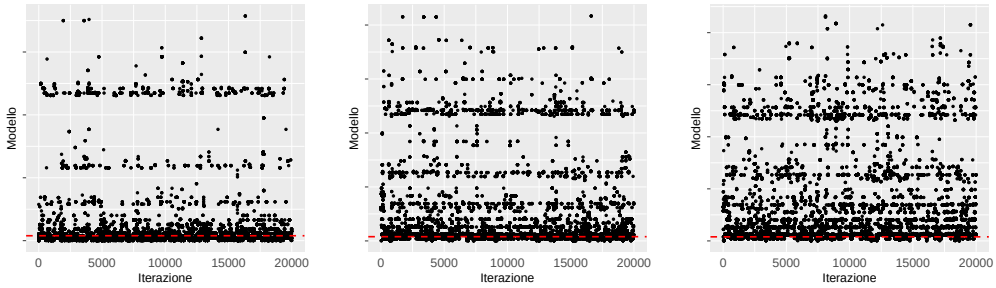
Vengono ora presentati i risultati degli algoritmi RJ e MT-RJ. Poichè $p < n$, ci si aspettano risultati simili dai due algoritmi.

La figura 5.5 mostra i modelli visitati dagli algoritmi, codificati secondo (5.2). Questi esplorano la stessa fascia di modelli, tra i quali è presente $\boldsymbol{\gamma}^*$ (linea rossa). All'aumentare di ϕ aumenta anche il *mixing* della catena, raggiungendo il massimo quando $\phi = 0.5$. Inoltre, la percentuale di nuovi valori accettati post burn-in è influenzata anche dal numero di proposte valutate ad ogni ite-

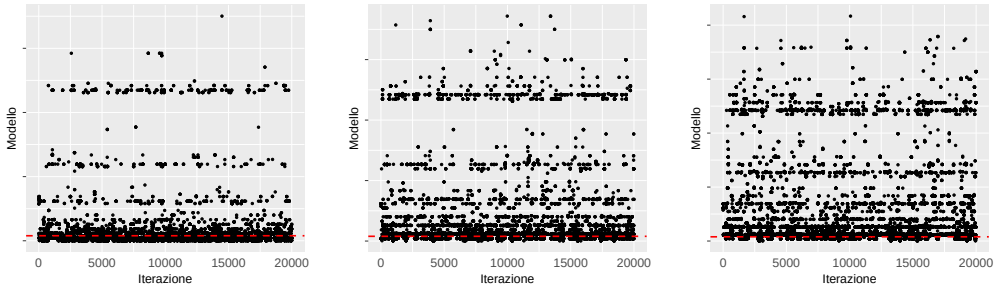
razione: questa è massima quando $k = 10$, con valori che vanno dal 58% al 64%, e scende quando $k = 30$. Il parametro k, infatti, insieme al parametro ϕ, è un parametro di regolazione per la probabilità di accettazione.



(a) $RJ : \phi = 0.1$ **(b)** $RJ : \phi = 0.3$ **(c)** $RJ : \phi = 0.5$



(d) $MT : k = 10,\ \phi = 0.1$ **(e)** $MT : k = 10,\ \phi = 0.3$ **(f)** $MT : k = 10,\ \phi = 0.5$

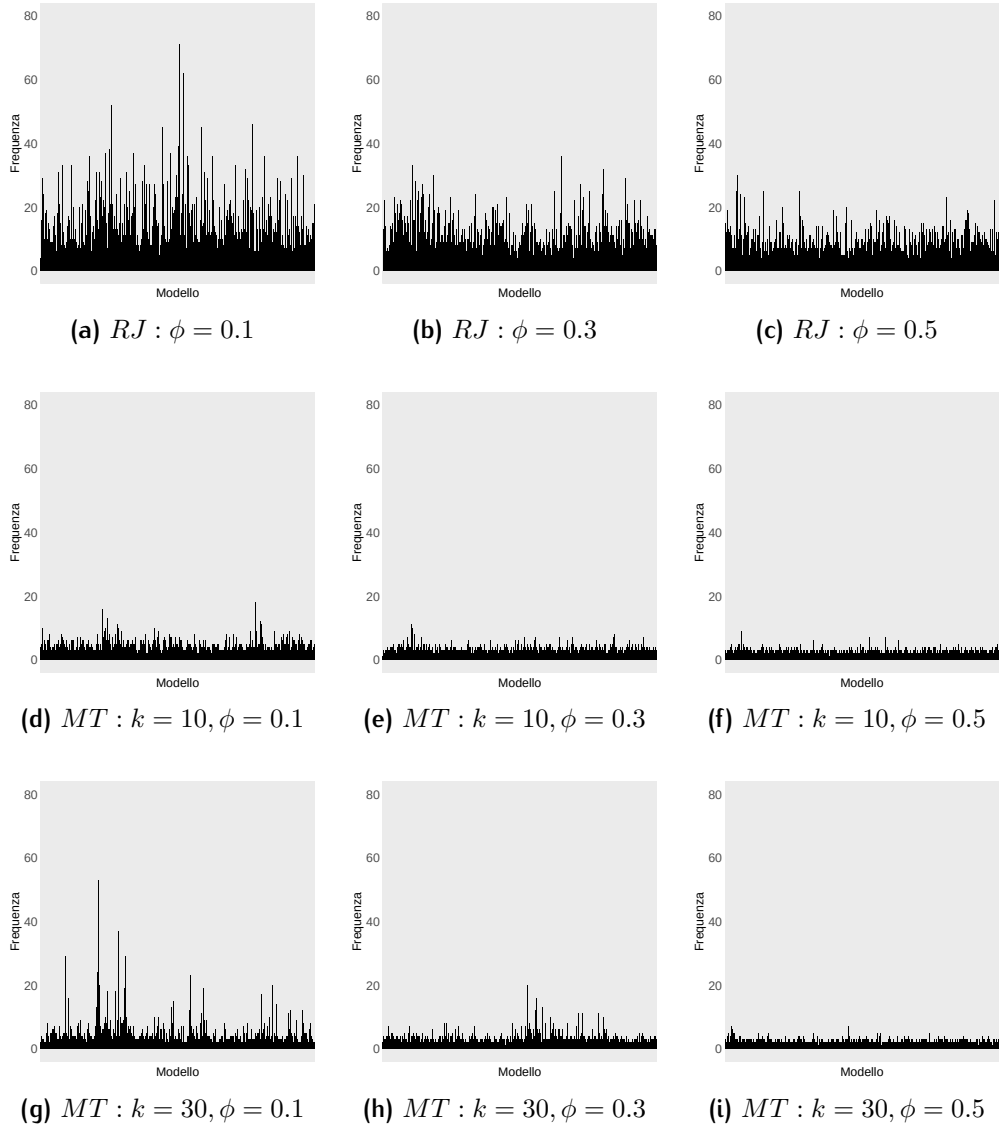


(g) $MT : k = 30,\ \phi = 0.1$ **(h)** $MT : k = 30,\ \phi = 0.3$ **(i)** $MT : k = 30,\ \phi = 0.5$

Figura 5.5: Modelli visitati dagli algoritmi RJ e MT-RJ, al variare di k e ϕ.

La distribuzione a-posteriori di $\boldsymbol{\gamma}$ è riportata in figura 5.6, dove vengono mostrate le frequenze assolute con cui i modelli vengono visitati. L'asse delle ordinate riporta i valori fino a 80, per poter visualizzare meglio il comporta-

mento di $\boldsymbol{\gamma}$ al variare di k e ϕ. Le frequenze più basse nell'algoritmo MT-RJ sono causate dal maggior numero di transizioni tra modelli e dal numero di proposte k, evidenziando una maggior flessibilità rispetto all'algoritmo RJ. Inoltre, le mode nella distribuzione di $\boldsymbol{\gamma}$ evidenziano la capacità degli algoritmi di individuare zone della distribuzione obiettivo ad alta densità.



(a) $RJ : \phi = 0.1$ **(b)** $RJ : \phi = 0.3$ **(c)** $RJ : \phi = 0.5$

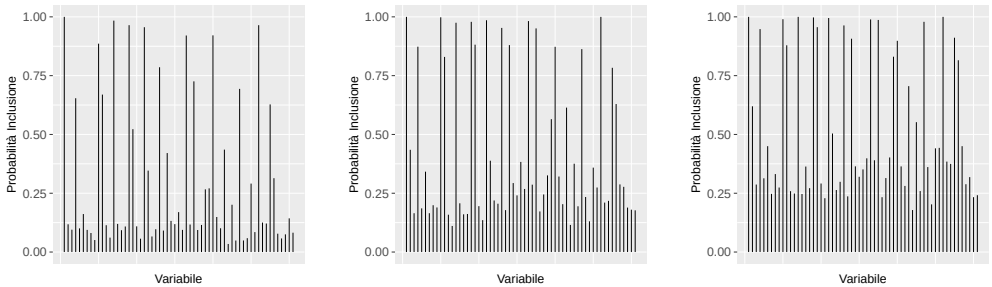
(d) $MT : k = 10, \phi = 0.1$ **(e)** $MT : k = 10, \phi = 0.3$ **(f)** $MT : k = 10, \phi = 0.5$

(g) $MT : k = 30, \phi = 0.1$ **(h)** $MT : k = 30, \phi = 0.3$ **(i)** $MT : k = 30, \phi = 0.5$

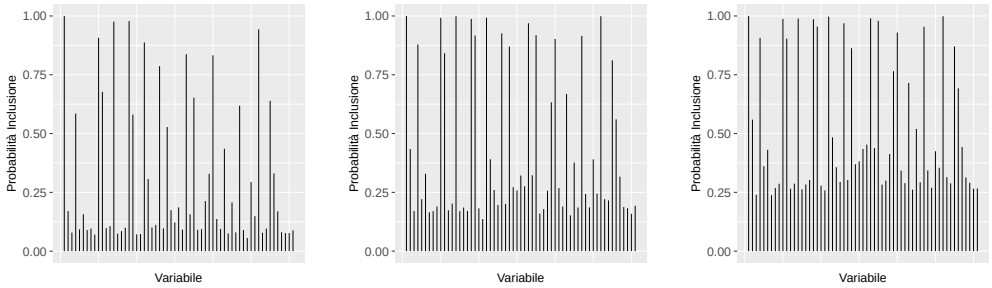
Figura 5.6: Distribuzione a-posteriori di $\boldsymbol{\gamma}$, al variare di k e ϕ.

Le probabilità a-posteriori di inclusione delle variabili π_j, $j = 1, \ldots, p$, sono

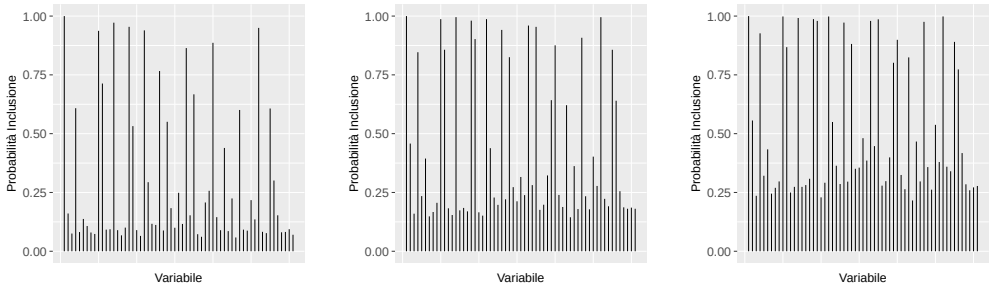
rappresentate in Figura 5.7: non vi sono grandi differenze tra gli algoritmi, ma ciò è dovuto al fatto che il numero di variabili non è alto e l'algoritmo RJ è in grado di esplorare in maniera esaustiva la densità a-posteriori di $\boldsymbol{\gamma}$.



(a) $RJ : \phi = 0.1$ **(b)** $RJ : \phi = 0.3$ **(c)** $RJ : \phi = 0.5$



(d) $MT : k = 10, \phi = 0.1$ **(e)** $MT : k = 10, \phi = 0.3$ **(f)** $MT : k = 10, \phi = 0.5$



(g) $MT : k = 30, \phi = 0.1$ **(h)** $MT : k = 30, \phi = 0.3$ **(i)** $MT : k = 30, \phi = 0.5$

Figura 5.7: Probabilità post burn-in di inclusione delle variabili per gli algoritmi RJ e MT-RJ, al variare di k e ϕ.

La selezione delle variabili avviene in base alla soglia scelta con il metodo di Youden e il j-esimo regressore viene incluso nel modello se la probabilità di

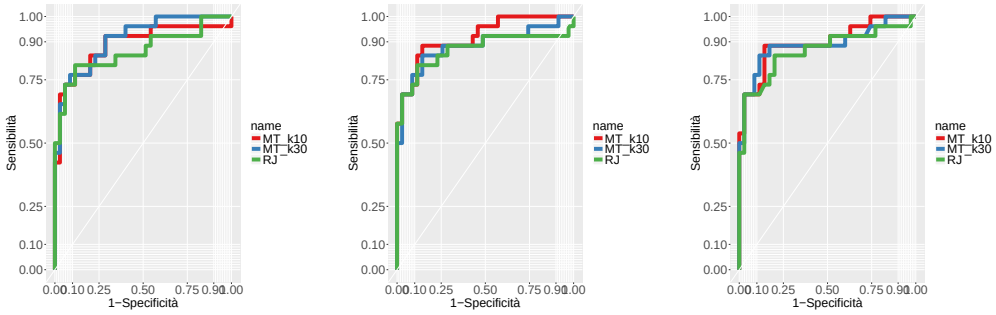
inserimento π_j, $j = 1, \ldots, p$, è maggiore di tale soglia. La tabella 5.3 riporta i risultati dei due algoritmi. Se confrontato con le prestazioni di GS (Tabella 5.2), si nota un miglioramento in termini di tasso di errata classificazione, che assume un valore compreso tra 13% e 15% per tutti i casi considerati. Inoltre, gli algoritmi soppesano maggiormente le variabili classificate in maniera errata nelle due classi, inducendo un maggior bilanciamento tra Sensibilità e Specificità. Poichè il numero di variabili esplicative è ridotto, l'algoritmo RJ esplora in maniera esaustiva la distribuzione a-posteriori di $\boldsymbol{\gamma}$ e i due metodi offrono prestazioni simili. La Sensibilità raggiunge il massimo quando la catena raggiunge un buon *mixing*: quando $\phi = 0.3$ e $\phi = 0.5$ questa raggiunge l'80% con l'algoritmo MT-RJ di parametro $k = 30$. La Specificità di MT-RJ aumenta al diminuire del parametro ϕ, con valori rispettivamente pari a 0.94 e 0.92 quando $k = 10$ e $k = 30$, se $\phi = 0.1$, contrariamente all'algoritmo RJ che, al contrario, presenta massima Specificità, pari al 97%, quando $\phi = 0.5$. Tuttavia, in questo caso la Sensibilità scende al 69%.

		$\phi = 0.1$		$\phi = 0.3$		$\phi = 0.5$	
		$\hat{\boldsymbol{\gamma}} = 0$	$\hat{\boldsymbol{\gamma}} = 1$	$\hat{\boldsymbol{\gamma}} = 0$	$\hat{\boldsymbol{\gamma}} = 1$	$\hat{\boldsymbol{\gamma}} = 0$	$\hat{\boldsymbol{\gamma}} = 1$
RJ	$\boldsymbol{\gamma}^* = 0$	31	4	31	4	34	1
	$\boldsymbol{\gamma}^* = 1$	5	21	5	21	8	18
MT-RJ k=10	$\boldsymbol{\gamma}^* = 0$	33	2	30	5	30	5
	$\boldsymbol{\gamma}^* = 1$	7	19	3	23	3	23
MT-RJ k=30	$\boldsymbol{\gamma}^* = 0$	32	3	30	5	31	4
	$\boldsymbol{\gamma}^* = 1$	6	20	4	22	4	22

Tabella 5.3: Risultati della selezione dei regressori per gli algoritmi RJ e MT-RJ, al variare di k e ϕ.

Le curve ROC per la valutazione della selezione delle variabili degli algoritmi sono raffigurate in Figura 5.8. La statistica AUC assume valore costantemente sopra 0.80, indice di un'ottima prestazione degli algoritmi. L'algoritmo MT-RJ raggiunge valori più elevati dell'algoritmo RJ in tutti i casi considerati. Le statistiche AUC relative all'algoritmo RJ assumono valore simile al variare di ϕ, pari a 0.86 quando $\phi = 0.1$ e 0.87 quando $\phi = 0.3$ e $\phi = 0.5$.

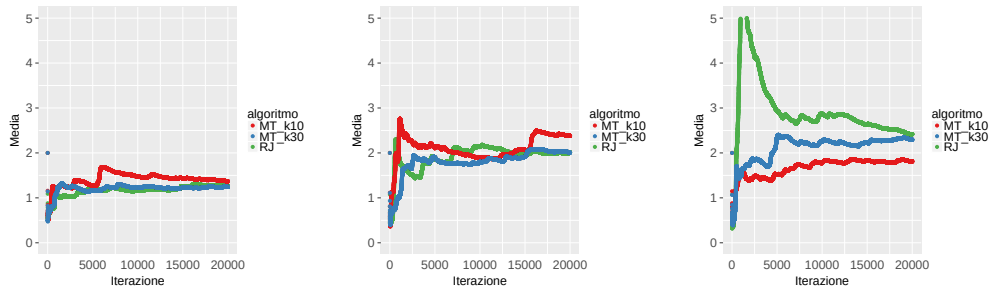
I valori dell'*AUC* per l'algoritmo MT-RJ, infine, si aggirano intorno a 0.90, con un picco pari a 0.92 quando $\phi = 0.3$ e $k = 10$.



(a) $\phi = 0.1$ **(b)** $\phi = 0.3$ **(c)** $\phi = 0.5$

Figura 5.8: Curva ROC per le prestazioni degli algoritmi RJ e MT-RJ, al variare di k e ϕ.

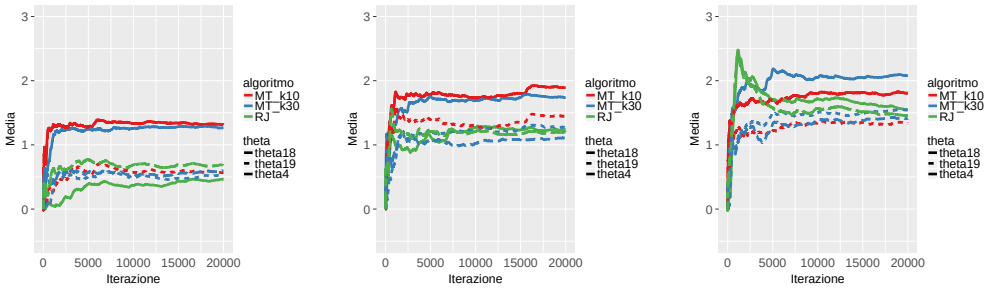
Il comportamento della media a-posteriori dell'iperparametro σ^2 è mostrato in Figura 5.9. Dalla figura si evince che all'aumentare di ϕ aumenta sia la variabilità che il valore della media: l'aumento della variabilità è dovuto ad un maggior numero di salti trans-dimensionali, mentre l'aumento della stima ad una complessità maggiore dei modelli visitati. Escludendo il periodo di burn-in, questa passa da 1.3 quando $\phi = 0.1$ a un valore superiore a 2 nei casi $\phi = 0.3$ e $\phi = 0.5$.



(a) $\phi = 0.1$ **(b)** $\phi = 0.3$ **(c)** $\phi = 0.5$

Figura 5.9: Comportamento della media di σ^2 negli algoritmi RJ e MT-RJ, al variare di k e ϕ.

In Figura 5.10 viene mostrata l'evoluzione della media dei θ_j, $j = 4, 18, 19$, già presi in considerazione dal Gibbs Sampling in Figura 5.4, all'aumentare delle iterazioni. Poichè dopo il periodo di assestamento l'algoritmo visita un numero maggiore di modelli diversi tra loro e con una complessità maggiore, il valore assunto dalla media a-posteriori post burn-in dei parametri aumenta all'aumentare di ϕ, ed è soggetta a maggiore variabilità. In particolare, la massima variabilità è presente nell'algoritmo MT-RJ con $k = 10$. Le diverse stime dei parametri tra l'algoritmo GS e gli algoritmi RJ e MT-RJ influenzano le probabilità a-posteriori di inserimento delle variabili: le stime più elevate del Gibbs Sampling forzano i regressori a entrare nel modello con probabilità 0 o 1, mentre queste assumono valori intermedi nel caso di RJ e MT-RJ.



(a) $\phi = 0.1$ **(b)** $\phi = 0.3$ **(c)** $\phi = 0.5$

Figura 5.10: Comportamento della media di $\theta_j, j = 4, 18, 19$ negli algoritmi RJ e MT-RJ, al variare di k e ϕ.

5.1.2 Caso 2: $p > n$

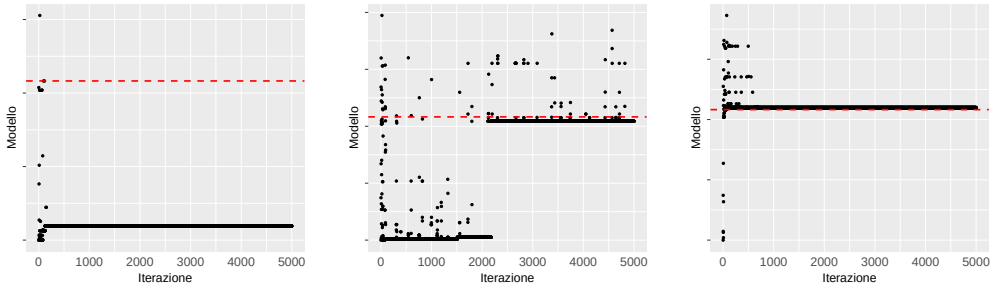
Il caso più interessante per l'applicazione dei metodi presentati è quello in cui il numero di esplicative sia elevato e più alto del numero di osservazioni. E' stato simulato un dataset composto da 200 osservazioni e 300 variabili, 25 delle quali formano il modello generatore dei dati $\boldsymbol{\gamma}^*$. Le classi della variabile risposta $\mathbf{Y}$ presentano sbilanciamento: la frequenza della categoria $\mathbf{Y} = 1$ è del 14%. Il Gibbs Sampling è costruito su 5000 iterazioni, mentre sono state generate catene da 30000 iterazioni per gli algoritmi 2 e 6, con un periodo

di burn-in pari a 5000; questo è pari a 1000 iterazioni nel Gibbs Sampling. Vengono trattati i casi $\phi = 0.1$, $\phi = 0.3$ e $\phi = 0.5$. Inoltre, per valutare il comportamento dell'algoritmo MT-RJ al variare del numero di proposte k, questo è posto pari a 10 e 30. Le ipotesi sulle distribuzioni a-priori di $\boldsymbol{\theta}$ e σ^2 sono

$$\pi(\boldsymbol{\theta}) \sim \mathsf{N}(0, \sigma^2 \mathbf{I}_p), \quad \sigma^2 \sim \mathsf{IG}(2, 1). \tag{5.7}$$

La prima condizione assume indipendenza a-priori per le componenti del parametro $\boldsymbol{\theta}$, mentre la seconda vuole evitare variabilità eccessiva tra i θ_j, $j = 1, \ldots, p$. Infine, i parametri v_0 e v_1 relativi alle varianze delle priori Spike and Slab sono posti rispettivamente pari a 0.1 e 100.

La Figura 5.11 presenta i modelli codificati secondo (5.2), visitati ad ogni iterazione dall'algoritmo GS. La probabilità a-priori di inclusione delle variabili nel modello influenza le prestazioni dell'algoritmo. Questo converge verso due diversi modelli quando $\phi = 0.1$ e $\phi = 0.5$, mentre un salto diretto tra i due avviene dopo 2000 iterazioni quando $\phi = 0.3$, dovuto all'inserimento dell'ultimo regressore θ_{301} nel modello. Tuttavia, in questo caso la probabilità di accettazione è più alta e la catena è soggetta a maggior variabilità.

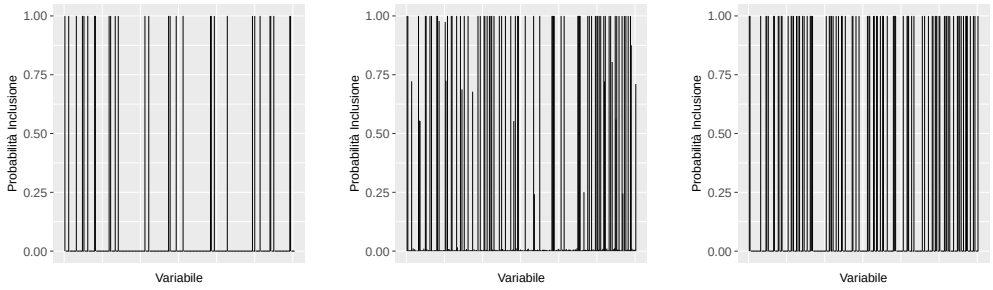


(a) $GS : \phi = 0.1$ **(b)** $GS : \phi = 0.3$ **(c)** $GS : \phi = 0.5$

Figura 5.11: Modelli visitati dall'algoritmo GS, al variare di ϕ.

L'inserimento dei regressori nel modello avviene con probabilità 0 o 1 quando $\phi = 0.1$ e $\phi = 0.5$, come evidenziato in Figura 5.12. La probabilità a-priori di inserimento delle variabili nel modello influenza le prestazioni dell'algoritmo,

aumentando la complessità del modello stimato. Come nel caso $p < n$, un aumento associato al parametro ϕ è collegato all'inserimento di nuove variabili nel modello, in quanto i regressori inseriti nel caso $\phi = 0.1$ fanno parte del modello stimato anche quando $\phi = 0.3$, così come le variabili selezionate da questo sono comprese anche nel modello se $\phi = 0.5$. Infine, il caso $\phi = 0.3$ è l'unico che restituisce probabilità di inserimento intermedie.



(a) $GS : \phi = 0.1$ **(b)** $GS : \phi = 0.3$ **(c)** $GS : \phi = 0.5$

Figura 5.12: Probabilità post burn-in di inclusione delle variabili per l'algoritmo GS, al variare di ϕ.

		$\phi = 0.1$		$\phi = 0.3$		$\phi = 0.5$	
		$\hat{\gamma} = 0$	$\hat{\gamma} = 1$	$\hat{\gamma} = 0$	$\hat{\gamma} = 1$	$\hat{\gamma} = 0$	$\hat{\gamma} = 1$
GS	$\gamma^* = 0$	258	17	204	71	199	76
	$\gamma^* = 1$	13	13	15	11	13	13

Tabella 5.4: Risultati della selezione dei regressori per l'algoritmo GS, al variare di ϕ.

I risultati dell'algoritmo sono raffigurati in Tabella 5.4. Ogni caso riporta la selezione delle variabili sulla base della soglia stimata con l'indice di Youden. Il tasso di corretta classificazione e la Specificità sono massimi quando $\phi = 0.1$, rispettivamente 90% e 94%, poichè la bassa probabilità a-priori di inserimento nel modello porta all'esclusione di un'elevata quantità di regressori. Non si registrano risultati significativi per i casi $\phi = 0.3$ e $\phi = 0.5$, con

i tassi di corretta classificazione che assumono un valore intorno al 70% in entrambi i casi.

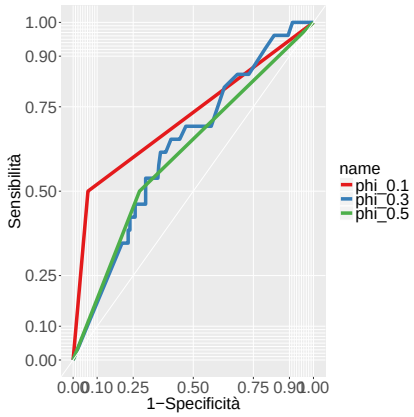
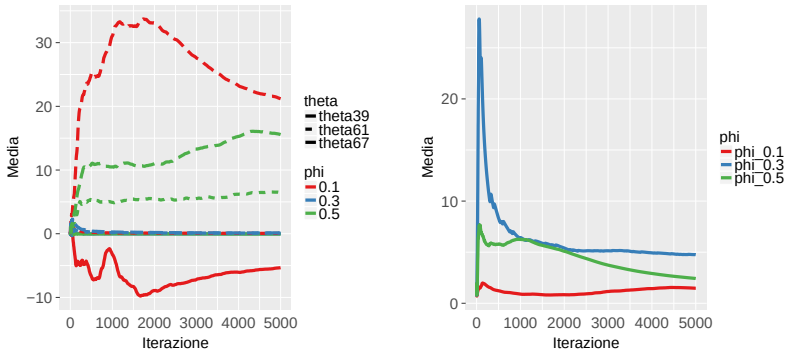


Figura 5.13: Curva ROC per le prestazioni dell'algoritmo GS, al variare di ϕ.

Per valutare la bontà dell'algoritmo viene presa in considerazione la curva ROC, la quale è presentata per i tre casi in Figura 5.13. Il massimo raggiunto dalla statistica AUC è pari a 0.71 quando $\phi = 0.1$. Dalle tre curve si evince che il problema principale è la scarsa Sensibilità della selezione delle variabili.



(a) $\theta_j, j = 4, 18, 19$ **(b)** σ^2

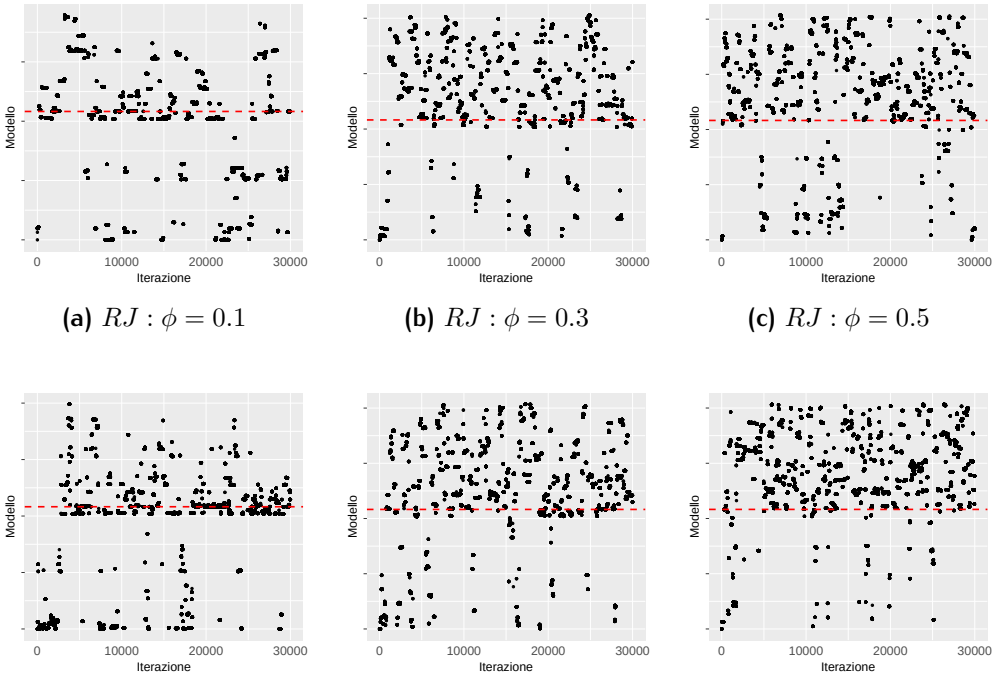
Figura 5.14: Comportamento della media dei parametri θ_j, $j = 39, 61, 67$, e σ^2 nell'algoritmo GS, al variare di ϕ.

La rappresentazione grafica della media dei parametri $\boldsymbol{\theta}$ e σ^2 (Figura 5.14) evidenzia l'influenza di ϕ; le stime dei parametri sono diverse ai vari livelli del'iperparametro di $\boldsymbol{\gamma}$. I parametri stimati sotto l'ipotesi $\phi = 0.1$ non arrivano a convergenza entro le 5000 iterazioni, consigliando la costruzione di

una catena più lunga, mentre nei casi $\phi = 0.3$ e $\phi = 0.5$ le stime tendono alla loro media. La media del parametro σ^2 assume valori diversi nei tre casi, con un valore minimo quando $\phi = 0.1$.

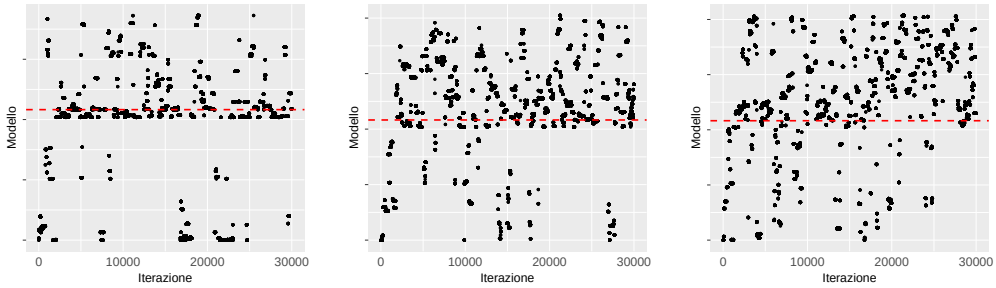
Vengono ora presentati i risultati degli algoritmi RJ e MT-RJ.

La Figura 5.15 mostra i modelli visitati dagli algoritmi RJ e MT-RJ, codificati secondo (5.2), al variare di ϕ e k. Il *mixing* della catena aumenta all'aumentare della probabilità a-priori d'inclusione delle variabili nel modello, ed è maggiore per l'algoritmo MT-RJ, giungendo al massimo quando $k = 10$. In particolare, la probabilità di accettazione di un nuovo stato della catena nell'algoritmo RJ passa dal 29% quando $\phi = 0.1$ al 67% quando $\phi = 0.5$, mentre MT-RJ accetta il salto verso il modello proposto nel 76% delle iterazioni post burn-in. I grafici evidenziano la caratteristica degli algoritmi di ridurre l'esplorazione della funzione obiettivo ad un fascia di modelli comprendente il modello generatore dei dati, $\hat{\boldsymbol{\gamma}}$ (linea rossa).



(a) $RJ : \phi = 0.1$ **(b)** $RJ : \phi = 0.3$ **(c)** $RJ : \phi = 0.5$

(d) $MT : k = 10,\ \phi = 0.1$ **(e)** $MT : k = 10,\ \phi = 0.3$ **(f)** $MT : k = 10,\ \phi = 0.5$



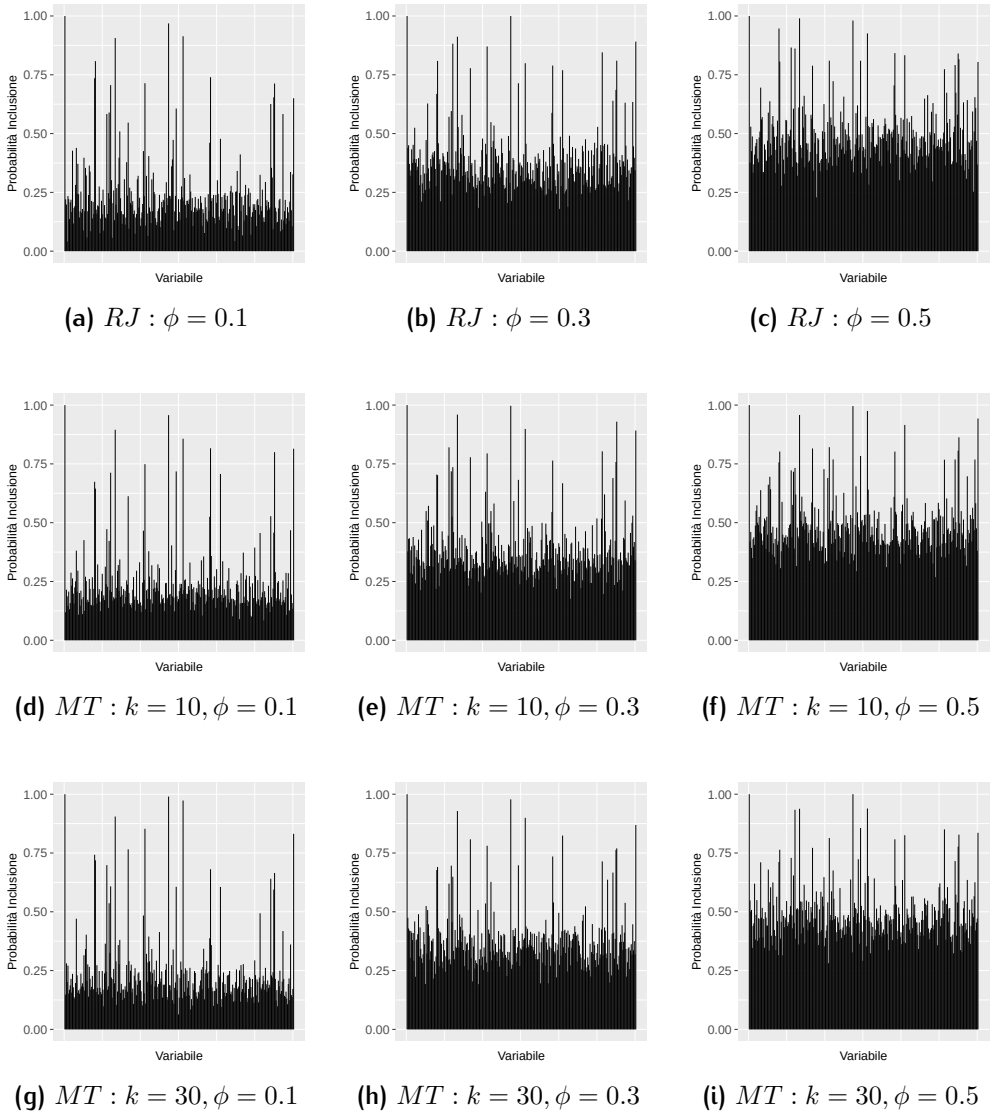
(a) $MT : k = 30,\ \phi = 0.1$ **(b)** $MT : k = 30,\ \phi = 0.3$ **(c)** $MT : k = 30,\ \phi = 0.5$

Figura 5.15: Modelli visitati dagli algoritmi RJ e MT-RJ al variare di k e ϕ.

Le probabilità a-posteriori di inclusione delle variabili π_j sono mostrate in Figura 5.16. Come nel caso del Gibbs Sampling, aumentando ϕ, una caratteristica dei due algoritmi è quella di aggiungere nuovi regressori a quelli già presenti nel modello stimato con un valore minore dell'iperparametro. In questo modo le variabili le inserite nel modello quando $\phi = 0.1$ fanno parte anche del modello stimato con $\phi = 0.3$, il quale è compreso a sua volta nel modello con $\phi = 0.5$. Non si notano particolari differenze dovute al numero k di proposte valutate ad ogni iterazione.

Basandosi sul criterio di Youden per la scelta ottimale della soglia oltre la quale le variabili vengono inserite nel modello, i risultati degli algoritmi sono riportati in Tabella 5.5. Il tasso di errata classificazione delle variabili è minimo (4%) nel caso $\phi = 0.3$ dell'algoritmo RJ, con una Specificità del 98%. La Sensibilità massima si registra nell'algoritmo MT-RJ con $k = 10$: quando $\phi = 0.3$ tale statistica arriva al 96%. Si notano altri due casi in cui il tasso di errata classificazione raggiunge il 5%: quando $k = 10$ e $\phi = 0.1$ o $k = 30$ e $\phi = 0.5$ l'algoritmo MT-RJ restituisce lo stesso modello a meno di un regressore; la Sensibilità e Specificità dei due modelli sono rispettivamente del 77% e del 96%.

Nonostante la Tabella 5.5 mostri che, scegliendo un cut-off ottimale, l'algoritmo RJ è in grado di pareggiare le prestazioni dell'algoritmo MT-RJ, le statistiche AUC relative alle curve ROC in Figura 5.17 suggeriscono un miglioramento dei risultati quando viene presa in considerazione più di una proposta. In quest'ultimo caso l'AUC non è mai minore di quella calcolata



(a) $RJ : \phi = 0.1$ **(b)** $RJ : \phi = 0.3$ **(c)** $RJ : \phi = 0.5$

(d) $MT : k = 10, \phi = 0.1$ **(e)** $MT : k = 10, \phi = 0.3$ **(f)** $MT : k = 10, \phi = 0.5$

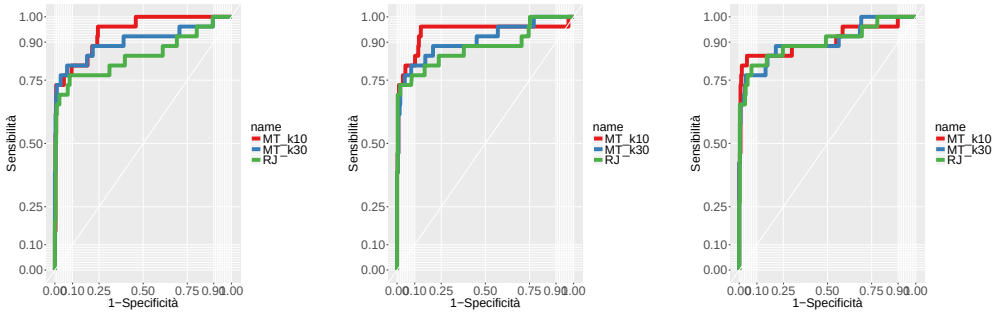
(g) $MT : k = 30, \phi = 0.1$ **(h)** $MT : k = 30, \phi = 0.3$ **(i)** $MT : k = 30, \phi = 0.5$

Figura 5.16: Probabilità post burn-in di inclusione delle variabili per gli algoritmi RJ e MT-RJ, al variare di k e ϕ.

a partire dall'algoritmo RJ. In particolare, con $k = 10$, questa assume valore pari a 0.94 quando $\phi = 0.1$ e $\phi = 0.3$. Se $\phi = 0.5$ i due algoritmi si equivalgono, con un valore dell'AUC pari a 0.9. I risultati sono nettamente migliori se paragonati a quelli del Gibbs Sampling 5.4, con valori della statistica AUC di venti punti percentuali in più per ogni caso trattato.

		$\phi = 0.1$		$\phi = 0.3$		$\phi = 0.5$	
		$\hat{\gamma} = 0$	$\hat{\gamma} = 1$	$\hat{\gamma} = 0$	$\hat{\gamma} = 1$	$\hat{\gamma} = 0$	$\hat{\gamma} = 1$
RJ	$\gamma^* = 0$	254	21	270	5	256	19
	$\gamma^* = 1$	6	20	7	19	5	21
MT-RJ k=10	$\gamma^* = 0$	265	10	238	37	259	16
	$\gamma^* = 1$	6	20	1	25	5	21
MT-RJ k=30	$\gamma^* = 0$	256	19	253	22	265	10
	$\gamma^* = 1$	5	21	5	21	6	20

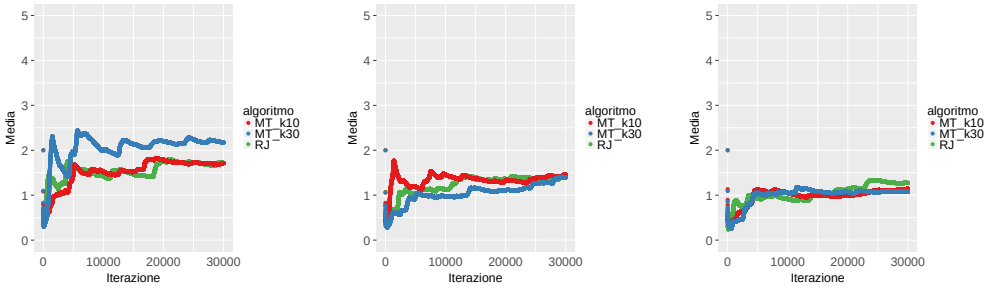
Tabella 5.5: Risultati della selezione dei regressori per gli algoritmi RJ e MT-RJ, al variare di k e ϕ.



(a) $\phi = 0.1$ **(b)** $\phi = 0.3$ **(c)** $\phi = 0.5$

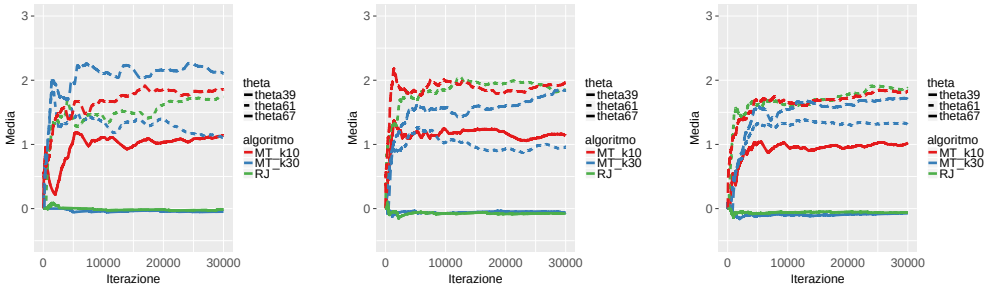
Figura 5.17: Curva ROC per le prestazioni degli algoritmi RJ e MT-RJ, al variare di k e ϕ.

Il comportamento delle medie a-posteriori dei parametri σ^2 e $\boldsymbol{\theta}$ è riportato in Figura 5.18 e Figura 5.19. La stima di σ^2 ottenuta dai due algoritmi tende ad un valore simile quando $\phi = 0.3$ e $\phi = 0.5$, tuttavia questa ha variabilità maggiore nell'algoritmo MT-RJ con $k = 30$. Il parametro ϕ non influisce sulle medie a-posteriori dei parametri θ_j, $j = 39, 61, 67$, le quali dipendono dal numero di proposte k.



(a) $\phi = 0.1$ **(b)** $\phi = 0.3$ **(c)** $\phi = 0.5$

Figura 5.18: Comportamento della media di σ^2 negli algoritmi RJ e MT-RJ, al variare di k e ϕ.



(a) $\phi = 0.1$ **(b)** $\phi = 0.3$ **(c)** $\phi = 0.5$

Figura 5.19: Convergenza di $\theta_j, j = 39, 61, 67$ negli algoritmi RJ e MT-RJ, al variare di k e ϕ.

5.2 Applicazione a dati reali

Vengono trattati due dataset reali:

- **Pima Indians**: dataset per l'analisi del diabete di tipo 2 su un gruppo di nativi Americani, composto da 768 osservazioni e 8 variabili esplicative, relative all'età, all'indice di massa corporea (BMI) e all'obesità, a gravidanze passate e, infine, a caratteristiche del sangue e del plasma;

- **BMKA**: dataset composto da 74 osservazioni e 516 variabili relative all'espressione di alcuni geni in vari tipi di tumore a diversi stadi;

Per valutare la bontà del modello, i dataset vengono divisi in dataset di stima e dataset di verifica, mantenendo invariate le frequenze delle due classi della variabile risposta. L'algoritmo viene stimato sul primo dataset e testato sul secondo.

La funzione predittiva per la previsione di nuove osservazioni è un voto di maggioranza, ed è definita come

$$\begin{aligned}\mathbb{P}(y_{new} = 1 \mid \mathbf{x}_{new}, \boldsymbol{\theta}) &= \frac{1}{M}\sum_{i=1}^{M} \mathbf{I}\left(p(y_{new} \mid \mathbf{x}_{new}, \mathbf{y}, \mathbf{x}, \boldsymbol{\theta}^{(i)}) > 0.5\right) \\ &= \frac{1}{M}\sum_{i=1}^{M} \mathbf{I}\left(\frac{e^{\mathbf{x}_{new}\boldsymbol{\theta}^{(i)}}}{1 + e^{\mathbf{x}_{new}\boldsymbol{\theta}^{(i)}}} > 0.5\right) \end{aligned} \tag{5.8}$$

dove M è il numero di modelli post burn-in con le quali vengono ottenute le nuove previsioni, $\boldsymbol{\theta}^{(i)}$ è il vettore di parametri relativo all'i-esimo modello e $p(\cdot)$ è la funzione logistica.

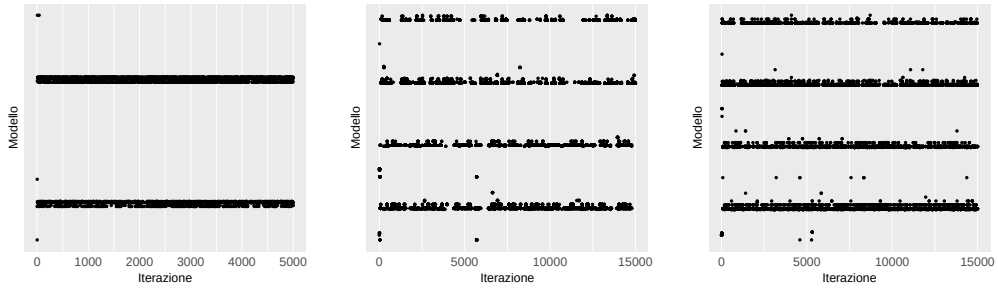
La curva ROC viene usata come strumento per valutare la bontà del modello e per il confronto tra gli algoritmi, mentre la soglia oltre la quale classificare le nuove osservazione nella classe $\mathbf{Y} = 1$ è scelta secondo il criterio di Youden.

5.2.1 Caso 1: $p < n$

Il primo dataset preso in esame è il dataset **Pima Indians**. 500 osservazioni sono riservate alla stima dei modelli, le restanti 268 alla validazione. La percentuale di osservazioni della classe $\mathbf{Y} = 1$ è pari al 35%. A causa del ristretto numero di variabili esplicative, ci si aspettano risultati simili dai tre algoritmi. Il Gibbs Sampling è costruito su 5000 iterazioni con periodo di burn-in pari a 1000, mentre le catene di RJ e MT-RJ hanno lunghezza 15000 e periodo di burn-in pari a 5000. Il numero di proposte differenti per l'algoritmo MT-RJ è $k = 5$. Il parametro ϕ è posto pari a 0.5; mentre $\pi(\boldsymbol{\theta}) \sim \mathsf{N}(0, \sigma^2\mathbf{I}_p)$. L'ipotesi a-priori relativa all'iperparametro associato alla variabilità dei regressori è $\sigma^2 \sim \mathsf{IG}(2, 1)$. I parametri v_0 e v_1 relativi alla distribuzione a-priori Spike and Slab sono posti rispettivamente pari a 0.1 e 100.

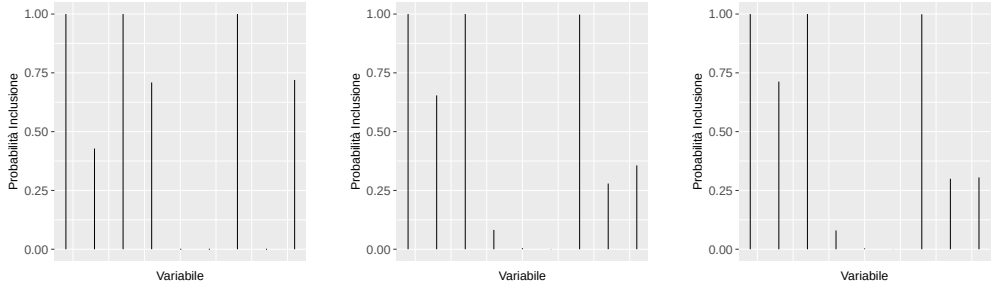
In Figura 5.20 vengono mostrati i modelli visitati dai tre algoritmi. I modelli RJ e MT-RJ si comportano in maniera simile, a differenza del Gibbs

Sampling che prevede un numero minore di salti trans-dimensionali. Tutti e tre i metodi evidenziano delle fasce di modelli ben definite.



(a) $GS : \phi = 0.5$ **(b)** $RJ : \phi = 0.5$ **(c)** $MT : k = 5, \phi = 0.5$

Figura 5.20: Modelli visitati dagli algoritmi GS, RJ e MT-RJ.



(a) GS **(b)** RJ **(c)** $MT : k = 5$

Figura 5.21: Probabilità post burn-in di inclusione delle variabili negli algoritmi GS, RJ e MT-RJ.

La Figura 5.21 mostra le probabilità di inclusione a-posteriori delle variabili: l'intercetta entra nel modello con probabilità 1 in tutti e tre i casi, mentre sono presenti differenze tra la selezione dei regressori nel Gibbs Sampling e negli algoritmi RJ e MT-RJ, i quali portano alle stesse conclusioni. Le variabili `Concentrazione del glucosio nel sange` e `BMI` risultano significative in tutti e tre gli algoritmi, con probabilità di inserimento vicina a 1. I risultati mostrano delle discrepanze relativamente alle variabili `Numero di gravidanze passate`, `Pressione del sangue diastolico` e `Età`:

la prima entra nel modello stimato dagli algoritmi RJ e MT-RJ, mentre le ultime due sono inserite nel modello solo dal Gibbs Sampling.

Vengono valutati ora i metodi sul dataset adibito alla verifica. La funzione predittiva è valutata su 200 modelli scelti casualmente tra quelli post burn-in. Le curve ROC sono mostrate in Figura 5.22. La stima dell'AUC per l'algoritmo Gibbs Sampling è 0.75, mentre risulta uguale per gli algoritmi RJ e MT-RJ, pari a 0.78. Questi due metodi, dunque, migliorano leggermente la qualità delle previsioni di nuove osservazioni, incrementando la percentuale di casi correttamente classificati a parità di Specificità.

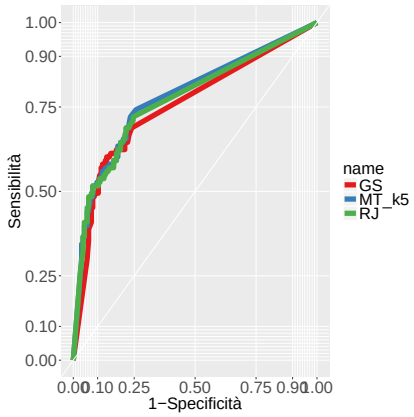


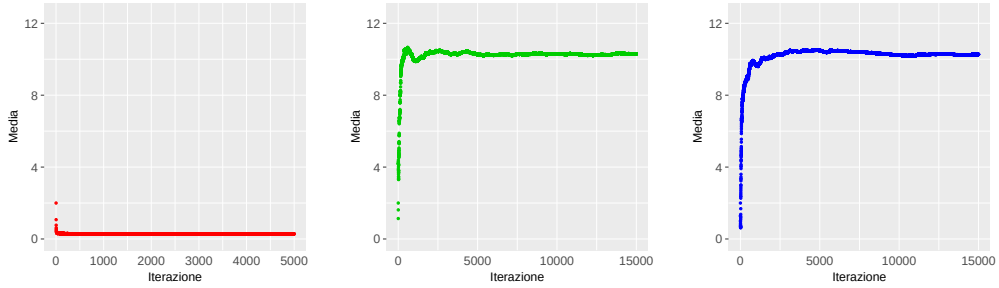
Figura 5.22: Curve ROC per le prestazioni degli algoritmi RJ e MT-RJ.

La tabella di confusione per la previsione delle nuove osservazioni è riportata in Tabella 5.6. Il tasso di errata classificazione delle nuove osservazioni è minimo per il modello MT-RJ (23%), con una Specificità pari all'86%. L'algoritmo RJ e il Gibbs Sampling riportano risultati simili, con un tasso di errata di classificazione intorno al 25%. Tuttavia, questi due metodi migliorano la Sensibilità del test, alzandola dal 60% nel caso dell'algoritmo MT, al 72%.

	GS		RJ		$MT_{k=5}$	
	$\mathbf{Y}_{pred}=0$	$\mathbf{Y}_{pred}=1$	$\mathbf{Y}_{pred}=0$	$\mathbf{Y}_{pred}=1$	$\mathbf{Y}_{pred}=0$	$\mathbf{Y}_{pred}=1$
$\mathbf{Y}_{new}=0$	132	43	134	41	151	24
$\mathbf{Y}_{new}=1$	26	67	26	67	37	56

Tabella 5.6: Risultati della previsione dei modelli RJ e MT-RJ.

La media post burn-in del parametro σ^2 è differente tra gli algoritmi. Questa è pari a 0.28 nel caso del Gibbs Sampling, minore rispetto alla stima degli algoritmi RJ e MT-RJ, che suggeriscono un valore intorno a 10. Il comportamento di tale parametro è mostrato in Figura 5.23. In tutti e tre i casi il parametro arriva a convergenza già dopo la millesima iterazione.



(a) GS **(b)** RJ **(c)** $MT : k = 5$

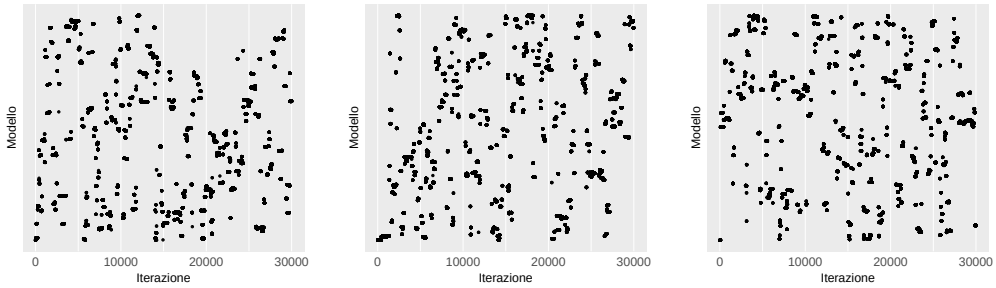
Figura 5.23: Convergenza del parametro σ^2 negli algoritmi GS, RJ e MT-RJ.

5.2.2 Caso 2: $p > n$

Il secondo dataset analizzato è **BMKA**, composto da un numero ristretto di osservazioni. Di queste, 50 sono adibite alla stima dei modelli e le restanti 24 alla verifica. La variabile risposta assume valore 1 nel 67% dei casi. L'alto numero di variabili scoraggia l'uso del Gibbs Sampling per motivi computazionali, in quanto i tempi di calcolo non sono indifferenti. Vengono presentati, dunque, i risultati relativi agli algoritmi RJ e MT-RJ. Le catene sono costruite su 30000, con un periodo di burn-in pari a 5000. Due valori del numero di proposte, $k = 10$ e $k = 30$, sono analizzati nell'algoritmo MT-RJ. In modo tale da esplorare al massimo la distribuzione obiettivo, la scelta a-priori del parametro che regola la probabilità di inclusione delle variabili ricade su $\phi = 0.5$. Infine, le ipotesi sulle distribuzioni a-priori di $\boldsymbol{\theta}$ e σ^2 sono

$$\pi(\boldsymbol{\theta}) \sim \mathsf{N}(0, \sigma^2 \mathbf{I}_p), \quad \sigma^2 \sim \mathsf{IG}(2, 1). \tag{5.9}$$

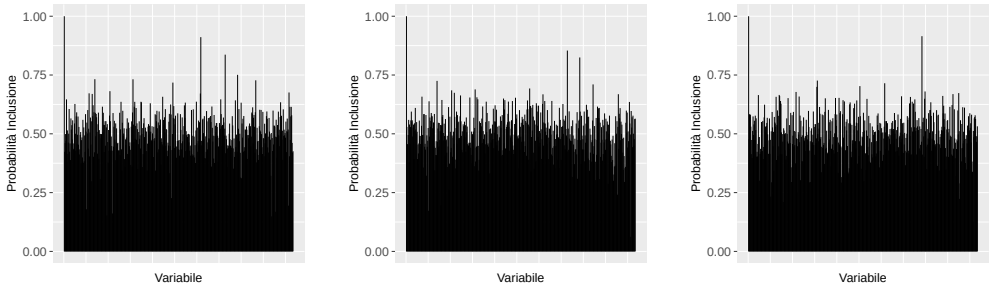
In Figura 5.24 vengono mostrati i modelli visitati dai due algoritmi. Anche dopo un periodo iniziale di assestamento, le catene non restringono l'esplorazione della distribuzione a-posteriori di $\boldsymbol{\gamma}$ ad una fascia di modelli e la



(a) RJ **(b)** $MT : k = 10$ **(c)** $MT : k = 30$

Figura 5.24: Modelli visitati dagli algoritmi RJ e MT-RJ.

probabilità di accettazione post burn-in è vicina a 1. Neanche l'aumento delle diverse proposte k prese in considerazione ad ogni iterazione aumenta la capacità di individuare zone della distribuzione target a più alta densità.



(a) RJ **(b)** $MT : k = 10$ **(c)** $MT : k = 30$

Figura 5.25: Probabilità post burn-in di inclusione delle variabili negli algoritmi RJ e MT-RJ.

L'intercetta entra nel modello con probabilità 1. Non si evincono differenze importanti tra le probabilità di inclusione a-posteriori delle variabili dei tre modelli, le quali sono mostrate in Figura 5.25. Sono stati analizzati i regressori relativi alle probabilità superiori a 0.7: questi risultano essere gli stessi in tutti i casi considerati e, in particolare, i geni 155, 363 e 391 entrano in almeno due dei tre modelli stimati.

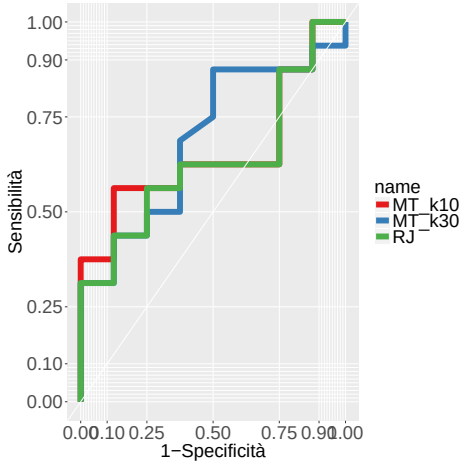
Vengono ora mostrati i risultati della classificazione per le 24 osservazioni adibite alla previsione, usando la funzione predittiva descritta in (5.8).

	RJ		$MT_{k=10}$		$MT_{k=30}$	
	$\mathbf{Y}_{pred}=0$	$\mathbf{Y}_{pred}=1$	$\mathbf{Y}_{pred}=0$	$\mathbf{Y}_{pred}=1$	$\mathbf{Y}_{pred}=0$	$\mathbf{Y}_{pred}=1$
$\mathbf{Y}_{new}=0$	6	2	7	1	4	4
$\mathbf{Y}_{new}=1$	7	9	7	9	2	14

Tabella 5.7: Risultati della previsione dei modelli RJ e MT-RJ.

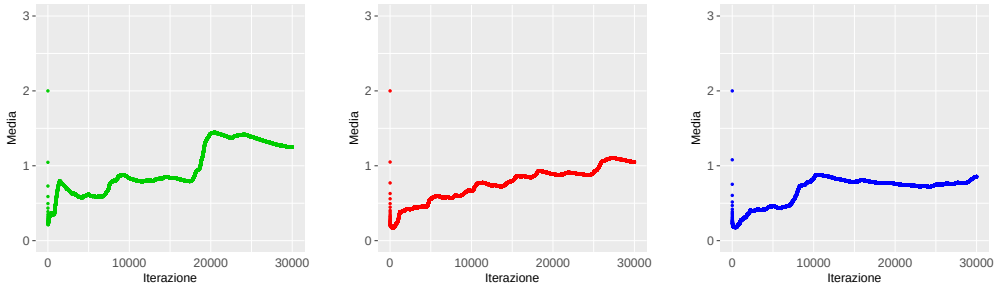
Scegliendo l'indice di Youden come soglia ottimale per la classificazione delle nuove osservazioni, i risultati dei modelli sono riportati in Tabella 5.7. Il tasso di errata classificazione è minimo (25%) quando l'algoritmo MT-RJ analizza $k = 30$ proposte ad ogni iterazione. Questo classificatore ha un'elevata Sensibilità (88%), tuttavia identifica correttamente la classe $\mathbf{Y} = 1$ solo nel 50% dei casi. La maggiore flessibilità del metodo Multiple-Try permette all'algoritmo di raggiungere valori di Sensibilità e Specificità più elevati rispetto all'algoritmo RJ. La massima Specificità si ottiene quando $k = 10$, ed è pari all'88%.

Figura 5.26: Curve ROC per le prestazioni degli algoritmi RJ e MT-RJ.



Il miglioramento della previsione di nuove osservazioni è confermato anche dalle statistiche AUC relative alle curve ROC raffigurate in Figura 5.26. In generale, il classificatore migliore risulta essere l'algoritmo MT-RJ con $k = 30$: l'area sotto la curva raggiunge il 69%, 6 punti percentuali in più rispetto all'algoritmo RJ, che presenta $AUC = 0.63$. Nonostante l'estensione Multiple-Try porti ad una maggiore capacità previsiva, l'analisi delle curve ROC suggerisce la possibilità di un un'ulteriore ottimizzazione: un valore

più elevato del parametro k potrebbe condurre ad un miglioramento delle prestazioni.



(a) $RJ : \phi = 0.5$ **(b)** $MT : k = 10, \phi = 0.5$ **(c)** $MT : k = 30, \phi = 0.5$

Figura 5.27: Convergenza del parametro σ^2 degli algoritmi RJ e MT-RJ.

Le stime delle medie a-posteriori dei parametri θ_j, $j = 1, \ldots, 516$ assumono valori appartenenti all'intervallo $(-1, 1)$. I grafici dei parametri relativi ai regressori da inserire nel modello risultano in linee rette molto vicine allo zero con pochissima variabilità, ragione per cui questi non vengono mostrati. La poca variabilità dei parametri è visibile in Figura 5.27, che suggerisce un valore basso della media a-posteriori nei tre casi.

Capitolo 6

Estensioni e generalizzazioni

Le classi di modelli presentate in questo lavoro ammettono ulteriori generalizzazioni e approfondimenti. Vengono brevemente presentati e analizzati i punti più critici dei metodi.

6.1 Convergenza del parametro γ

Quando il parametro $\boldsymbol{\gamma}$ ha dimensione elevata, lo studio della convergenza del modello è di difficile implementazione. In questo lavoro, la convergenza degli algoritmi è stata studiata solo graficamente. L'enumerazione dei modelli secondo (5.2) semplifica l'analisi allo studio della convergenza per un catena a stati discreti. In quest'ottica in Brooks, Giudici e Philippe (2003), gli autori presentano alcuni metodi non parametrici per la convergenza di $\boldsymbol{\gamma}$, tra i quali:

- **Test del Chi-quadro**: nel caso in cui si disponga di un numero J di catene di lunghezza B, a partire dal numero di volte in cui il modello i-mo viene visitato dalla catena j-ma, N_i^j, $i = 1, \ldots, 2^p$ e $j = 1, \ldots, J$, è possibile costruire una matrice di incidenza di dimensioni $J \times 2^p$, della forma

$$\begin{bmatrix} N_1^1 & \ldots & N_{2^p}^1 \\ \vdots & & \vdots \\ N_1^J & \ldots & N_{2^p}^J \end{bmatrix}. \tag{6.1}$$

Sotto l'assunzione di omogeneità delle catene, è possibile calcolare il valore medio della frequenza di ogni modello nelle J catene. Definito il numero atteso di volte che l'algoritmo visita l'i-mo modello $E_i = \sum_{j=1}^{J} N_i^j/J$, il test diventa

$$\chi_t^2 = \sum_{i=1}^{2^p} \sum_{j=1}^{J} \frac{(N_i^j - E_i)^2}{E_i}. \tag{6.2}$$

Sotto l'ipotesi nulla di omogeneità, il test si distribuisce come una distribuzione χ^2 con $J(2^p - 1)$ gradi di libertà.

La validità dell'approssimazione asintotica del test è garantita se le frequenze attese E_i, $i = 1, \ldots, 2^p$, assumono valore maggiore di 5 (Agresti, 1990). Gli algoritmi RJ e MT-RJ hanno la caratteristica di concentrarsi sui modelli maggiormente supportati dai dati, tuttavia, nel caso in cui il numero p di variabili sia particolarmente grande, la matrice (6.1) è composta da un elevato numero di zeri: difficilmente l'ipotesi su E_i, $i = 1, \ldots, 2^p$ è rispettata. Per ovviare a tale problema è necessario l'inserimento di uno schema che raggruppi i modelli con minor frequenza, alzando il valore del numero atteso di frequenze dei modelli e garantendo la validità asintotica del test;

- **Tasso di convergenza diretto**: dopo B iterazioni è possibile costruire una matrice $\mathbf{P}$ di transizione per i modelli visitati. L'elemento $[i, j]$ della matrice è calcolato come $\mathbf{P}^{[i,j]} = N_{ij}/B$, dove N_{ij} è il numero di transizioni dal modello i al modello j occorsi nella catena. Anche quando il numero di variabili esplicative è ridotto, molti salti tra modelli non sono effettuabili e la matrice $\mathbf{P}$ comprende un elevato numero di zeri. E' possibile permutare le righe di zeri in fondo e calcolare il secondo autovalore della matrice, il quale è un indice della convergenza della catena.

Il problema principale dei metodi proposti è l'elevata dimensionalità di $\boldsymbol{\gamma}$, poichè la capacità di memoria dei software non è sufficiente quando il valore $2^p \to \infty$ e l'enumerazione diventa infattibile. Una possibile soluzione del problema è quella di creare un numero m di cluster di modelli con densità

a-posteriori simile e calcolare i test di convergenza su questi. L'inserimento dei cluster di modelli permette di sostituire il valore 2^p con il valore 2^m e la costruzione del test del Chi-quadro e della matrice di transizione diventano realizzabili.

6.2 Generalizzazione della funzione proposta $q(\gamma' \mid \gamma)$

Il punto debole degli algoritmi 2 e 6 è la funzione proposta $q(\boldsymbol{\gamma}' \mid \boldsymbol{\gamma})$, poichè l'esplorazione della distribuzione obiettivo avviene solo tramite spostamenti locali. Questo giustifica una sua generalizzazione, in grado di proporre nuovi candidati in punti più distanti dello spazio, aumentando l'efficienza degli algoritmi. Inoltre, la scelta di tale funzione influenza direttamente il costo computazionale degli algoritmi.

La funzione generalizzata per la proposta di nuovi modelli è del tipo

$$q(\boldsymbol{\gamma}' \mid \boldsymbol{\gamma}) = w_d, \qquad \text{se} \quad \sum_{j=1}^{p} \left|\gamma_j' - \gamma_j\right| = d, \tag{6.3}$$

dove w_d è una probabilità che dipende dal numero d di variabili modificate nel modello ed è minima in $d = p/2$. In generale, il numero di modelli possibili in seguito alla modifica di d variabili è $\binom{p}{d}$ ed è possibile porre $w_d = 1/\binom{p}{d}$, assegnando uguale probabilità a tutte le proposte. Inoltre, tale funzione è simmetrica in $\boldsymbol{\gamma}$ e $\boldsymbol{\gamma}'$, dunque i pesi si riducono a $w(\boldsymbol{\gamma}' \mid \boldsymbol{\gamma}) = \pi(\boldsymbol{\gamma}', \sigma^2 \mid \mathbf{y}, \mathbf{X}, \boldsymbol{\omega})$.

Nel caso in cui il numero di regressori sia particolarmente elevato, è di interesse poter esplorare la distribuzione obiettivo in più direzioni, aggiungendo flessibilità all'algoritmo. In Casarin, Craiu e Leisen (2010) gli autori formalizzano un'estensione dell'algoritmo Multiple-Try che permette l'uso di k funzioni proposta differenti, senza compromettere l'ergodicità della catena.

Siano $q_i(\boldsymbol{\gamma}^{(i)} \mid \boldsymbol{\gamma})$, $i = 1, \ldots, k$, un set di funzioni per i nuovi candidati. I pesi $w_i(\boldsymbol{\gamma}^{(i)} \mid \boldsymbol{\gamma})$, $i = 1, \ldots, k$, sono definiti come in (1.8) ed hanno la forma

$$w_i(\boldsymbol{\gamma}^{(i)} \mid \boldsymbol{\gamma}) = \pi(\boldsymbol{\gamma}^{(i)} \mid \mathbf{y}) q_i(\boldsymbol{\gamma} \mid \boldsymbol{\gamma}^{(i)}) \xi(\boldsymbol{\gamma}, \boldsymbol{\gamma}^{(i)}). \tag{6.4}$$

Algorithm 7 MT-RJ Marginalizzato: generalizzazione

Inizializzazione di $\boldsymbol{\gamma}_0, \sigma_0^2, \boldsymbol{\omega}_0$, simulando dalla distribuzione a-priori del parametro;

for $b = 2$ to B **do**:

for $i = 1$ to k **do**

Generare $\boldsymbol{\gamma}^{(i)}$ secondo la densità $q_i(\boldsymbol{\gamma}^{(i)} \mid \boldsymbol{\gamma}^{(b-1)}) = 1/\binom{p}{i}$, se $\sum_{j=1}^{p} \left|\gamma_j^{(i)} - \gamma_j^{(b-1)}\right| = i$;

Calcolare $w_i(\boldsymbol{\gamma}^{(i)} \mid \boldsymbol{\gamma}^{(b-1)})$;

end for

Porre $\boldsymbol{\gamma}^{(j)} = \boldsymbol{\gamma}'$ secondo la densità di probabilità

$$\bar{w}_k = \frac{w_j(\boldsymbol{\gamma}^{(j)} \mid \boldsymbol{\gamma}^{(b-1)})}{\sum_{i=1}^{k} w_i(\boldsymbol{\gamma}^{(i)} \mid \boldsymbol{\gamma}^{(b-1)})};$$

Generare $\mathbf{v}^{(i)}$ dalla densità $q_i(\mathbf{v}^{(i)} \mid \boldsymbol{\gamma}^{(j)})$ e calcolare $w_i(\mathbf{v}^{(i)} \mid \boldsymbol{\gamma}^{(j)})$ per $i = \ldots, k$, $i \neq j$; porre $\mathbf{v}^{(j)} = \boldsymbol{\gamma}^{(b-1)}$ e calcolare $w_j(\mathbf{v}^{(j)} \mid \boldsymbol{\gamma}^{(j)})$;

Calcolare

$$\alpha\left(\boldsymbol{\gamma}^{(b-1)}, \boldsymbol{\gamma}^{(j)}\right) = \min\left\{1, \frac{\sum_{i=1}^{k} w_i(\boldsymbol{\gamma}^{(i)} \mid \boldsymbol{\gamma}^{(b-1)})}{\sum_{i=1}^{k} w_i(\mathbf{v}^{(i)} \mid \boldsymbol{\gamma}^{(j)})}\right\}; \tag{6.5}$$

Con probabilità $\alpha\left(\boldsymbol{\gamma}^{(b-1)}, \boldsymbol{\gamma}^{(j)}\right)$:

Porre $\hat{\boldsymbol{\gamma}}^{(b)} = \boldsymbol{\gamma}^{(j)}$;

Simulare dalle distribuzioni condizionate:

$\hat{\boldsymbol{\theta}}^{(b)} \mid \mathbf{y}, \mathbf{X}_{\gamma^{(b)}}, \boldsymbol{\gamma}^{(b)}, \sigma^2_{(b-1)}, \boldsymbol{\omega}^{(b-1)} \sim \mathsf{N}(\boldsymbol{\Sigma}^*_\theta(\mathbf{X}'_{\gamma^{(b)}}\mathbf{W}\tilde{\tilde{\mathbf{y}}}), \boldsymbol{\Sigma}^*_\theta)$;

$\hat{\sigma}^2_{(b)} \mid \boldsymbol{\theta}^{(b)}, \boldsymbol{\gamma}^{(b)} \sim \mathsf{IG}\left(\lambda + \frac{p_\gamma}{2}, \nu + \frac{1}{2}\sum_{j=1}^{p_\gamma} \theta_j^2\right)$;

$\hat{\boldsymbol{\omega}}^{(b)} \mid \mathbf{X}_{\gamma^{(b)}}, \boldsymbol{\theta}^{(b)}, \boldsymbol{\gamma}^{(b)} \sim \mathsf{PG}(1, \mathbf{X}_{\gamma^{(b)}}\boldsymbol{\theta}^{(b)})$;

Definire, altrimenti:

$\hat{\boldsymbol{\gamma}}^{(b)} = \boldsymbol{\gamma}^{(b-1)}, \qquad \hat{\boldsymbol{\theta}}^{(b)} = \boldsymbol{\gamma}^{(b-1)}$;

$\hat{\sigma}^2_{(b)} = \sigma^2_{(b-1)}, \qquad \hat{\boldsymbol{\omega}}^{(b)} = \boldsymbol{\omega}^{(b-1)}$.

end for

Le possibilità per la scelta di $\xi(\boldsymbol{\gamma}, \boldsymbol{\gamma}^{(i)})$ sono le stesse presentate in precedenza.

Per assicurare la reversibilità dell'algoritmo è necessario, dopo aver scelto

il candidato $\boldsymbol{\gamma}^{(j)}$ tra i k proposti, generare $k-1$ valori ausiliari $\boldsymbol{v}^{(i)}$, dalla distribuzione $q_i(\boldsymbol{v}^{(i)} \mid \boldsymbol{\gamma}^{(j)})$. La probabilità di accettazione segue l'equazione (4.1), con l'eccezione che i pesi tengono conto di differenti funzioni per le proposte.

L'algoritmo 7, costruito con distribuzioni differenti per le proposte è un caso particolare di un altro tipo di Multiple-Try, formalizzato in Casarin, Craiu e Leisen (2010), l'*Interacting Multiple-Try*. Nell'articolo gli autori considerano una popolazione di N catene e propongono un metodo tale per cui le k proposte di ogni catena dipendono dalla stato di tutte le altre al passo precedente. Nel caso più generale, si assume che ogni catena abbia un numero di proposte diverse tra loro pari a k_n, $n = 1, \ldots, N$ ($k_n = 1$ implica l'algoritmo MH), aumentando decisamente la flessibilità dell'algoritmo. Sia $\boldsymbol{\theta}^{(b)} = \left\{\theta_n^{(b)}\right\}_{n=1}^{N}$ il vettore di stati delle N catene all'iterazione b. Secondo l'Interacting MT, il vettore $\boldsymbol{\theta}^{(b)}$ dipende dai valori assunti da $\boldsymbol{\theta}^{(b-1)}$. L'algoritmo è presentato in 8, dove $\tilde{f}_n(\theta) = (\boldsymbol{\theta}^{(1:n-1)}, \theta, \boldsymbol{\theta}^{(i+1:N)})$ è una funzione che mappa la proposta per l'n-ma catena che dipende dai valori delle altre $N-1$ catene. La dipendenza dello stato $\boldsymbol{\theta}^{(b)}$ può essere estesa al vettore di stati corrente, tuttavia, tale schema è di difficile implementazione (Casarin, Craiu e Leisen, 2010).

La probabilità di accettazione definita in (6.6) rispetta la condizione di reversibilità, come mostrato in Casarin, Craiu e Leisen (2010). Gli autori, inoltre, presentano uno schema per la scelta del numero di catene N e il numero di proposte k_n. Poichè il costo computazionale è alto nel caso in cui i due valori siano elevati, la ricerca del compromesso migliore tra onere di calcolo e flessibilità dell'algoritmo diventa di fondamentale importanza:

- se il numero di catene è $5 \leq N \leq 20$, è possibile porre $k_n = N$, permettendo alle catene di interagire in maniera diretta tra loro e di esplorare la distribuzione obiettivo in varie direzioni;

- se il numero di catene è elevato, per esempio $N = 100$, l'analisi della distribuzione obiettivo avviene in maniera soddisfacente scegliendo un numero di proposte pari a 1, senza perdita di efficienza. In generale, gli autori suggeriscono una scelta del rapporto $k_n/N \in [5\%, 20\%]$.

Quando N è elevato, è possibile evitare l'uso di tutte le catene; i risultati ottenuti nell'articolo, dimostrano che l'uso ad ogni iterazione di un sottoinsieme casuale composto da M stati porta ad un miglioramento dei risultati. In particolare, una scelta ottimale di del numero di catene da usare per generare il nuovo stato $\boldsymbol{\gamma}^{(b)}$ è $M = k_n$ (Casarin, Craiu e Leisen, 2010).

Algorithm 8 Interacting MT

Inizializzazione di $\boldsymbol{\theta}_0$, simulando dalla distribuzione a-priori del parametro;

for $b = 2$ to B **do**:
  for $n = 1$ to N **do**
    Generare $\boldsymbol{\theta}_n^{(1)}, \ldots, \boldsymbol{\theta}_n^{(k_n)} \sim q_i(\boldsymbol{\theta}_n^{(i)} \mid \tilde{f}_n(\boldsymbol{\theta}_n^{(b-1)}))$, calcolare $w_i(\boldsymbol{\theta}_n^{(i)} \mid \boldsymbol{\theta}_n^{(b-1)}), \quad i = 1, \ldots, k_n$;
    Porre $\boldsymbol{\theta}_n^{(j)} = \boldsymbol{\theta}_n'$ secondo la densità di probabilità

$$\bar{w}_n = \frac{w_j(\boldsymbol{\theta}_n^{(j)} \mid \boldsymbol{\theta}_n^{(b-1)})}{\sum_{i=1}^{k_n} w_i(\boldsymbol{\theta}_n^{(i)} \mid \boldsymbol{\theta}_n^{(b-1)})};$$

    Generare $\mathbf{v}_n^{(1)}, \ldots, \mathbf{v}_n^{(j-1)}, \mathbf{v}_n^{(j+1)}, \ldots, \mathbf{v}_n^{(k_n)} \sim q(\mathbf{v}_n^{(i)} \mid \tilde{f}_n(\boldsymbol{\theta}_n^{(j)}))$ e porre $\mathbf{v}_n^{(j)} = \boldsymbol{\theta}_n^{(b-1)}$ e calcolare $w_i(\mathbf{v}_n^{(i)} \mid \boldsymbol{\theta}_n^{(j)}), \quad i = 1, \ldots, k_n$;
    Porre $\boldsymbol{\theta}_n^{(b)} = \boldsymbol{\theta}_n^{(j)}$ con probabilità

$$\alpha(\boldsymbol{\theta}_n^{(b-1)}, \boldsymbol{\theta}_n^{(j)}) = \min\left(1, \frac{\sum_{i=1}^{k_n} w_i(\boldsymbol{\theta}_n^{(i)} \mid \boldsymbol{\theta}_n^{(b-1)})}{\sum_{i=1}^{k_n} w_i(\mathbf{v}_n^{(i)} \mid \boldsymbol{\theta}_n^{(j)})}\right), \tag{6.6}$$

    $\boldsymbol{\theta}_n^{(b)} = \boldsymbol{\theta}_n^{(b-1)}$ altrimenti;
  end for
end for

Un ultimo ragionamento viene fatto sulla scelta di $q(\boldsymbol{\gamma}' \mid \boldsymbol{\gamma})$ negli algoritmi 6 e 7. Quando il numero di variabili esplicative è molto elevato, è desiderabile avere un numero elevato di candidati k ad ogni iterazione. Per abbattere il costo computazionale, alto nel caso in cui molte proposte vengano valutate, è possibile scegliere una funzione per i candidati indipendente dallo stato precedente, $q(\boldsymbol{\gamma}' \mid \boldsymbol{\gamma}) = q(\boldsymbol{\gamma}')$, e sfruttare lo schema presentato in (4.2) e

(4.3), il quale permette di riciclare ad ogni iterazione un numero pari a $k-1$ proposte scartate nel passo in avanti. Sotto tali ipotesi è possibile aumentare di molto il numero k di valori analizzati ad ogni iterazione, senza aggravare i tempi di calcolo. Una proposta che soddisfa tali caratteristiche è

$$q(\boldsymbol{\gamma}') = \frac{1}{2^p}. \tag{6.7}$$

In questo modo vengono proposti k modelli in maniera casuale ed ognuno di questi ha la stessa probabilità, permettendo il riciclo dei $k-1$ valori senza intaccare l'ergodicità della catena.

Conclusioni

Il lavoro svolto in questa tesi ha portato alla luce nuove tecniche per l'analisi di dati binari nel caso in cui il numero di variabili esplicative sia particolarmente elevato. E' stato analizzato inizialmente un metodo base, il Gibbs Sampling. A partire da questo una rassegna di estensioni e metodi più sofisticati sono stati presentati. I risultati ottenuti sono incoraggianti, con un miglioramento progressivo delle prestazioni da parte dei modelli testati.

L'analisi di dati dicotomici è senza dubbio uno dei campi maggiormente studiati e allo stesso tempo da studiare. Gli algoritmi formalizzati hanno il vantaggio di non essere ristretti al campo biologico, ma generalizzabili a qualsiasi ambito: nell'era dei *Big Data* questi hanno il potenziale per prendere piede e trovare sempre più applicazioni.

Il limite principale, come è stato sottolineato precedentemente, è di natura computazionale: senza sottostare ad ipotesi restrittive relative alla funzione per le proposte da valutare ad ogni iterazione, il costo computazionale è particolarmente oneroso. E' stata formalizzata un'approssimazione quadratica della distribuzione target per evitare di valutare tale funzione per ogni stato proposto, teoricamente salvando non poco tempo di calcolo. Inoltre, è stato nominato anche l'algoritmo di Chambers per un'efficiente inversione delle matrici in seguito all'inserimento o eliminazione di una riga/colonna, passaggio ricorrente in questa classe di modelli. Tuttavia, questi metodi non sono stati implementati in questo lavoro e può essere uno spunto per miglioramenti futuri.

Un ulteriore limite è dato dalla difficoltà dello studio della convergenza del vettore di stati $\boldsymbol{\gamma}$ quando questo assume dimensioni particolarmente elevate. In questo caso, i metodi presenti in letteratura non permettono un'analisi

soddisfacente della convergenza e necessitano di revisioni e generalizzazioni.

In conclusione, i risultati mostrano la validità dei metodi presentati e allo stesso tempo la possibilità di aumentare la loro efficienza. Questa classe di modelli, insomma, necessita e merita ulteriori approfondimenti futuri.

Bibliografia

Agresti, Alan (1990). *Categorical Data Analysis*. Wiley, p. 572.

Baker, Stuart G e Barnett S Kramer (2007). «Peirce, Youden, and Receiver Operating Characteristic Curves». In: *The American Statistician* 61.4, pp. 343–346. DOI: 10.1198/000313007X247643. eprint: https://doi.org/10.1198/000313007X247643. URL: https://doi.org/10.1198/000313007X247643.

Brooks, S. P., P. Giudici e G. O. Roberts (2003). «Efficient Construction of Reversible Jump Markov Chain Monte Carlo Proposal Distributions». In: *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* 65.1, pp. 3–55. ISSN: 13697412, 14679868. URL: http://www.jstor.org/stable/3088825.

Brooks, Stephen, Paolo Giudici e Anne Philippe (apr. 2003). «Nonparametric Convergence Assessment for MCMC Model Selection». In: *Journal of Computational and Graphical Statistics* 12. DOI: 10.1198/1061860031347.

Brooks, Stephen P. (1998). «Markov Chain Monte Carlo Method and Its Application». In: *Journal of the Royal Statistical Society. Series D (The Statistician)* 47.1, pp. 69–100. ISSN: 00390526, 14679884. URL: http://www.jstor.org/stable/2988428.

Casarin, Roberto, Radu V. Craiu e Fabrizio Leisen (2010). «Interacting Multiple Try Algorithms with Different Proposal Distributions». In: *arXiv e-prints*, arXiv:1011.1170, arXiv:1011.1170. arXiv: 1011.1170 [stat.CO].

Chambers, John M. (1971). «Regression Updating». In: *Journal of the American Statistical Association* 66.336, pp. 744–748. ISSN: 01621459. URL: http://www.jstor.org/stable/2284221.

Fan, Y e S A Sisson (2010). «Reversible jump Markov chain Monte Carlo». In: *arXiv e-prints*, arXiv:1001.2055, arXiv:1001.2055. arXiv: 1001.2055 [stat.ME].

George, Edward I. e Robert E. McCulloch (1997). «Approaches for Bayesian Variable Selection». In: *Statistica Sinica* 7.2, pp. 339–373. ISSN: 10170405, 19968507. URL: http://www.jstor.org/stable/24306083.

Green, Peter J. (1995). «Reversible jump Markov chain Monte Carlo computation and Bayesian model determination». In: *Biometrika* 82, pp. 711–732.

Green, PJ (2003). «Introducing Highly Structured Stochastic Systems». English. In: *Highly Structured Stochastic Systems*. A cura di PJ Green, NL Hjort e S Richardson. United Kingdom: Oxford University Press, pp. 1 –12.

Hastings, W. K. (1970). «Monte Carlo Sampling Methods Using Markov Chains and Their Applications». In: *Biometrika* 57.1, pp. 97–109. ISSN: 00063444. URL: http://www.jstor.org/stable/2334940.

Lamnisos, Demetris, Jim E. Griffin e Mark F. J. Steel (2009). «Transdimensional Sampling Algorithms for Bayesian Variable Selection in Classification Problems With Many More Variables Than Observations». In: *Journal of Computational and Graphical Statistics* 18.3, pp. 592–612. ISSN: 10618600. URL: http://www.jstor.org/stable/25651265.

Liu, Jun S., Wing Hung Wong e Augustine Kong (1994). «Covariance Structure of the Gibbs Sampler with Applications to the Comparisons of Estimators and Augmentation Schemes». In: *Biometrika* 81.1, pp. 27–40. ISSN: 00063444. URL: http://www.jstor.org/stable/2337047.

Malsiner-Walli, Gertraud e Helga Wagner (2016). «Comparing Spike and Slab Priors for Bayesian Variable Selection». In: *Austrian Journal of Statistics* 40.4, 241–264. DOI: https://doi.org/10.17713/ajs.v40i4.215. URL: https://www.ajs.or.at/index.php/ajs/article/view/vol40\%2C\%20no4\%20-\%202.

Martino, Luca (2018). «A Review of Multiple Try MCMC algorithms for Signal Processing». In: *arXiv e-prints*, arXiv:1801.09065, arXiv:1801.09065. arXiv: 1801.09065 [stat.CO].

Pandolfi, Silvia, Francesco Bartolucci e Nial Friel (gen. 2010). «A generalization of the Multiple-try Metropolis algorithm for Bayesian estimation and model selection.» In: *Journal of Machine Learning Research - Proceedings Track* 9, pp. 581–588.

Papathomas, Michail, Petros Dellaportas e Vassilis Vasdekis (gen. 2009). «A general proposal construction for reversible jump MCMC». In:

Polson, Nicholas G., James G. Scott e Jesse Windle (2012). «Bayesian inference for logistic models using Polya-Gamma latent variables». In: *arXiv e-prints*, arXiv:1205.0310, arXiv:1205.0310. arXiv: 1205.0310 [stat.ME].

Sanger, F., S. Nicklen e A. R. Coulson (1977). «DNA sequencing with chain-terminating inhibitors». In: *Proceedings of the National Academy of Sciences* 74.12, pp. 5463–5467. ISSN: 0027-8424. DOI: 10.1073/pnas.74.12.5463. eprint: https://www.pnas.org/content/74/12/5463.full.pdf. URL: https://www.pnas.org/content/74/12/5463.

Thisted, Ronald A. (1988). *Elements of Statistical Computing: Numerical Computation*. London, UK, UK: Chapman & Hall, Ltd. ISBN: 0-412-01371-1.

Appendice A

Codice R degli algoritmi presentati

Codice A.1: Distribuzione target $\pi(\boldsymbol{\gamma}, \boldsymbol{\theta}, \sigma^2 \mid \mathbf{y}, \mathbf{X})$

```
#x matrice dei dati
#y vettore risposta
#w vettore variabile latente Polya-Gamma
#sigma2 valore di $\sigma^2$
#dev matrice di varianza-covarianza a-osteriori per $\theta$
#D matrice di varianza-covarianza a-priori per $\theta$

log.marginal = function(x,y,w,sigma2,dev,D){
  p = ncol(x)
  y.tilde = (y-0.5)/w
  (-0.5)*logdet(sigma2*D)+0.5*logdet(dev)+0.5*t(y.tilde)%*%diag(w)%*%x%*%
      dev%*%t(x)%*%diag(w)%*%y.tilde
}
```

Codice A.2: Pesi esponenziali normalizzati in MT-RJ Marginalizzato

```
#vLogW vettore di log-pesi da normalizzare in forma esponenziale
```

```
self.weights = function(vLogW){
  dMaxW  = max(vLogW)
  vW     = exp(vLogW - dMaxW)
  dNc    = sum(vW)
  dLogNC = log(dNc) + dMaxW
  vW     = vW / dNc
  return(vW)
}
```

Codice A.3: Funzione per le proposte in MT-RJ Marginalizzato: generalizzazione

```
#gamma modello in input
#d numero variabili da modificare

proposal = function(gamma,d){
  p = length(gamma)
  p.s = sample(1:p,d)
  gamma[p.s] = 1-gamma[p.s]
  return(gamma)
}
```

Codice A.4: Spike and Slab Gibbs Sampling

```
library(pgdraw) #pacchetto per generare da distribuzione Polya-Gamma
library(msos) #pacchetto per calcolare il logaritmo del determinante di
    una matrice
library(mvtnorm) #pacchetto per generare da distribuzione multivariata

#B numero iterazioni
#y vettore risposta
#x matrice dei dati
#hyp vettore di iperparametri per $\sigma^2$
#v0, v1 varianze per distribuzione Spike e distribuzione Slab,
    rispettivamente
```

```
#D matrice di varianza-covarianza a-priori per $\theta$

gibbs = function(B,y,x,hyp,v0,v1,D){
  n = length(y)
  p = NCOL(x)

  sigma2 = rep(0,B)
  w = matrix(0, nrow=B, ncol=n)
  gamma = matrix(0, nrow=B, ncol=p)
  theta = matrix(0, nrow=B, ncol=p)
  pi0up = matrix(0, nrow=B, ncol=p)
  gamma.01 = rep(0,p)
  b=matrix(0, nrow=B, ncol=p)

  #inizializzazione di sigma2, theta, pi0up (vettore dei $\pi_j$)
  sigma2[1] = 2
  theta[1,] = rep(0,p)
  pi0up[1,] = rep(0.5,p)

  #iperparametri per $\sigma^2$
  lambda = hyp[1]
  v = hyp[2]

  for(i in 2:B){
    theta.gamma = theta[i-1,]

    #genero $\gamma$
    for(j in 1:p){
      gamma[i,j] = rbinom(1,1,prob=pi0up[i-1,j])
      b[i,j] = gamma[i,j]*v1+(1-gamma[i,j])*v0
    }

    #genero $\omega$
    w[i,] = pgdraw(1,x%*%theta.gamma)

    #genero $\sigma^2$
```

```
    sigma2[i] = 1/rgamma(1,(lambda+p/2),(v+0.5*sum(theta.gamma^2/b[i,])))

    #genero $\beta$
    y.tilde = (y-1/2)/w[i,]
    D.gamma = diag(b[i,])
    dev = chol2inv(chol(t(x)%*%diag(w[i,])%*%x+solve(sigma2[i]*D.gamma)))
    mean = dev%*%(t(x)%*%diag(w[i,])%*%y.tilde)
    theta[i,] = rmvnorm(1, mean=mean, sigma=dev)

    #aggiorno $\pi_j$
    for(j in 1:p){

      #prima proposta
      gamma.01[j] = exp(dbinom(sum(gam.without),p,phi,log=T)-dbinom(sum(
          gam.with),p,phi,log=T)) * exp(-0.5*logdet(sigma2[i]*D.lambda[-j
          ,-j])+0.5*logdet(sigma2[i]*D.lambda)) * exp(logdet(t(x[,-j])%*%
          diag(w[i,])%*%x[,-j]+solve(sigma2[i]*D.lambda[-j,-j]))-abs(
          logdet(dev)))^(-0.5) * exp(0.5*t(y.tilde)%*%diag(w[i,])%*%x[,-j]
          %*%chol2inv(chol(t(x[,-j])%*%diag(w[i,])%*%x[,-j]+solve(sigma2[i
          ]*D.lambda[-j,-j])))%*%t(x[,-j])%*%diag(w[i,])%*%y.tilde-0.5*t(y
          .tilde)%*%diag(w[i,])%*%x%*%dev%*%t(x)%*%diag(w[i,])%*%y.tilde)

      #seconda proposta
      gamma.01[j] = exp(dnorm(theta[i,j],0,sd=sqrt(sigma2[i]*v0),log=T)-
          dnorm(theta[i,j],0,sd=sqrt(sigma2[i]*v1),log=T))

      pi0up[i,j] = 1/(1+((1-pi0up[i-1,j])*gamma.01[j]/pi0up[i-1,j]))
    }

  }

  return(list(theta=theta,gamma=gamma,sigma=sigma2))
}
```

Codice A.5: RJ Marginalizzato

```
#B numero iterazioni
#y vettore risposta
#x matrice dei dati
#w0 vettore iniziale per omega
#phi probabilita d'inclusione delle variabili a-priori
#hyp vettore di iperparametri per $\sigma^2$
#gamma0 modello iniziale
#sigma0 valore iniziale per $\sigma^2$

rj = function(B,y,x,w0,phi,hyp,gamma0,sigma0){
  n = length(y)
  p = NCOL(x)

  gamma = matrix(0,B,p)
  sigma2 = rep(0,B)
  theta = matrix(0,B,p)
  w = matrix(0,B,n)

  #inizializzazione di gamma, sigma2, w
  gamma[1,] = gamma0
  sigma2[1] = sigma0
  w[1,] = w0

  #iperparametri per $\sigma^2$
  lambda = hyp[1]
  v = hyp[2]

  #contatore per nuove proposte accettate
  c = rep(0,B)

  for(i in 2:B){
```

```
#proposta per gamma
p.s = sample(2:p,1)
gamma.s = gamma[i-1,]
gamma.s[p.s] = 1-gamma.s[p.s]

x.gamma = x[,gamma[i-1,]==1] #matrice dei dati condizionata a gamma
D.gamma = diag(1,sum(gamma[i-1,])) #matrice varianza-covarianza a
    priori per theta a componenti indipendenti
#D.gamma = t(x.gamma)%*%x.gamma #matrice varianza-covarianza a priori
    per theta
dev.gamma = chol2inv(chol(t(x.gamma)%*%diag(w[i-1,])%*%x.gamma+1/
    sigma2[i-1]*chol2inv(chol(D.gamma))))

x.s = x[,gamma.s==1] #matrice dei dati condizionata a gamma'
D.s = diag(1,sum(gamma.s)) #matrice varianza-covarianza a priori per
    theta a componenti indipendenti
#D.s = t(x.s)%*%x.s #matrice varianza-covarianza a priori per theta
dev.s = chol2inv(chol(t(x.s)%*%diag(w[i-1,])%*%x.s+1/sigma2[i-1]*
    chol2inv(chol(D.s))))

#probabilita di accettazione
alpha = min(1,exp(dbinom(sum(gamma.s),p,phi,log=T) + log.marginal(x.s,
    y,w[i-1,],sigma2[i-1],dev.s,D.s) - dbinom(sum(gamma[i-1,]),p,phi,
    log=T) - log.marginal(x.gamma,y,w[i-1,],sigma2[i-1],dev.gamma,D.
    gamma)))
#aggiornamento dei parametri
y.tilde = (y-0.5)/w[i-1,]
r=runif(1)

if(r<=alpha){
  gamma[i,] = gamma.best
  c[i] = 1

  x.gamma = x[,gamma[i,]==1]
  D.gamma = diag(1,sum(gamma[i,])) #indipendenza a-priori
  #D.gamma = t(x.gamma)%*%x.gamma #correlazione a-priori
```

```
      dev.bb = chol2inv(chol(t(x.gamma)%*%diag(w[i-1,])%*%x.gamma+1/sigma2
          [i-1]*chol2inv(chol(D.gamma))))             mean.bb = dev.bb%*%(
          t(x.gamma)%*%diag(w[i-1,])%*%y.tilde)
      theta[i,gamma[i,]==1] = rmvnorm(1, mean=mean.bb, sigma=dev.bb)

      sigma2[i] = 1/rgamma(1,(lambda+sum(gamma[i,])/2),(v+0.5*sum(theta[i
          ,]^2)))

          w[i,] = pgdraw(1,x.gamma%*%theta[i,gamma[i,]==1])
        }

        if(r>alpha){
          gamma[i,] = gamma[i-1,]
          theta[i,] = theta[i-1,]
          sigma2[i]=sigma2[i-1]
          w[i,]=w[i-1,]
      }
  }

  return(list(gamma=gamma,sigma=sigma2,theta=theta,c=c))
}
```

Codice A.6: MT-RJ Marginalizzato

```
#B numero iterazioni
#y vettore risposta
#x matrice dei dati
#w0 vettore iniziale per omega
#phi probabilita d'inclusione delle variabili a-priori
#k numero proposte
#hyp vettore di iperparametri per $\sigma^2$
#gamma0 modello iniziale
#sigma0 valore iniziale per $\sigma^2$

rjmt = function(B,y,x,w0,phi,k,hyp,gamma0,sigma0){
```

```
n = length(y)
p = NCOL(x)

gamma = matrix(0,B,p)
sigma2 = rep(0,B)
theta = matrix(0,B,p)
w = matrix(0,B,n)
psi.forward = rep(0,k)
psi.backward = rep(0,k)

#inizializzazione di gamma, sigma2, w
gamma[1,] = gamma0
sigma2[1] = sigma0
w[1,] = w0

#iperparametri per $\sigma^2$
lambda = hyp[1]
v = hyp[2]

#contatore per nuove proposte accettate
c = rep(0,B)

ind=1:k

for(i in 2:B){

  #k proposte per gamma e calcolo dei log-pesi
  draw = matrix(rep(gamma[i-1,],k),k,p,byrow=T)
  for(h in 1:k){
    p.s = sample(2:p,1)
    draw[h,p.s] = 1-draw[h,p.s]

    tmp = x[,draw[h,]==1] #matrice dei dati condizionata a gamma

    D.gamma = diag(1,sum(draw[h,])) #matrice varianza-covarianza a
        priori per theta a componenti indipendenti
```

```
  #D.gamma = t(tmp)%*%tmp #matrice varianza-covarianza a priori per
      theta
  dev.gamma = chol2inv(chol(t(tmp)%*%diag(w[i-1,])%*%tmp+1/sigma2[i-1]
      *chol2inv(chol(D.gamma))))

  psi.forward[h] = dbinom(sum(draw[h,]),p,phi,log=T)+log.marginal(tmp,
      y,w[i-1,],sigma2[i-1],dev.gamma,D.gamma)
}

#scelta di $\gamma^{(j)}$
best = sample(1:k,1,prob=self.weights(psi.forward))
gamma.best = draw[best,]

#k-1 proposte all'indietro
draw.back = matrix(rep(gamma.best,k),k,p,byrow=T)

for(h in ind[-best]){
  p.s = sample(2:p,1)
  draw.back[h,p.s] = 1-draw.back[h,p.s]

  tmp = x[,draw.back[h,]==1] #matrice dei dati condizionata a gamma

  D.gamma = diag(1,sum(draw.back[h,])) #matrice varianza-covarianza a
      priori per theta a componenti indipendenti
  #D.gamma = t(tmp)%*%tmp #matrice varianza-covarianza a priori per
      theta
  dev.gamma = chol2inv(chol(t(tmp)%*%diag(w[i-1,])%*%tmp+1/sigma2[i-1]
      *chol2inv(chol(D.gamma))))

  psi.backward[h] = dbinom(sum(draw.back[h,]),p,phi,log=T)+log.
      marginal(tmp,y,w[i-1,],sigma2[i-1],dev.gamma,D.gamma)
}

D.gamma = diag(1,sum(gamma[i-1,]))
dev.gamma = chol2inv(chol(t(x[,gamma[i-1,]==1])%*%diag(w[i-1,])%*%x[,
    gamma[i-1,]==1]+1/sigma2[i-1]*chol2inv(chol(D.gamma))))
```

```
psi.backward[best] = dbinom(sum(gamma[i-1,]),p,phi,log=T)+log.marginal
    (x[,gamma[i-1,]==1],y,w[i-1,],sigma2[i-1],dev.gamma,D.gamma)

#normalizzazione pesi esponenziali
psi = self.weights(c(psi.backward,psi.forward))

#probabilita accettazione
alpha = min(1,sum(psi[(k+1):(2*k)])/sum(psi[1:k]))

#aggiornamento dei parametri
y.tilde = (y-0.5)/w[i-1,]
r=runif(1)

if(r<=alpha){
  gamma[i,] = gamma.best
  c[i] = 1

  x.gamma = x[,gamma[i,]==1]
  D.gamma = diag(1,sum(gamma[i,])) #matrice varianza-covarianza a
      priori per theta a componenti indipendenti
  #D.gamma = t(x.gamma)%*%x.gamma #matrice varianza-covarianza a
      priori per theta
  dev.bb = chol2inv(chol(t(x.gamma)%*%diag(w[i-1,])%*%x.gamma+1/sigma2
      [i-1]*chol2inv(chol(D.gamma))))
  mean.bb = dev.bb%*%(t(x.gamma)%*%diag(w[i-1,])%*%y.tilde)
  theta[i,gamma[i,]==1] = rmvnorm(1, mean=mean.bb, sigma=dev.bb)

  sigma2[i] = 1/rgamma(1,(lambda+sum(gamma[i,])/2),(v+0.5*sum(theta[i
      ,]^2)))

  w[i,] = pgdraw(1,x.gamma%*%theta[i,gamma[i,]==1])
}

if(r>alpha){
      gamma[i,] = gamma[i-1,]
```

```
          theta[i,] = theta[i-1,]
          sigma2[i]=sigma2[i-1]
          w[i,]=w[i-1,]
        }
  }

  return(list(gamma=gamma,sigma=sigma2,theta=theta,c=c))
}
```

Codice A.7: MT-RJ Marginalizzato: generalizzazione

```
#B numero iterazioni
#y vettore risposta
#x matrice dei dati
#w0 vettore iniziale per omega
#phi probabilita d'inclusione delle variabili a-priori
#k numero proposte differenti
#hyp vettore di iperparametri per $\sigma^2$
#gamma0 modello iniziale
#sigma0 valore iniziale per $\sigma^2$

rjmt_gen = function(B,y,x,w0,phi,k,hyp,gamma0,sigma0){
  n = length(y)
  p = NCOL(x)

  gamma = matrix(0,B,p)
  sigma2 = rep(0,B)
  theta = matrix(0,B,p)
  w = matrix(0,B,n)
  psi.forward = rep(0,k)
  psi.backward = rep(0,k)

  #inizializzazione di gamma, sigma2, w
  gamma[1,] = gamma0
  sigma2[1] = sigma0
```

```
w[1,] = w0

#iperparametri per $\sigma^2$
lambda = hyp[1]
v = hyp[2]

#contatore per nuove proposte accettate
c = rep(0,B)

ind=1:k

for(i in 2:B){

  #k proposte per gamma e calcolo dei log-pesi
  draw = matrix(rep(gamma[i-1,],k),k,p,byrow=T)
  for(h in 1:k){
    draw[h,]=proposal(draw[h,],h)

    tmp = x[,draw[h,]==1]

    D.gamma = diag(1,sum(draw[h,])) #matrice varianza-covarianza a
        priori per theta a componenti indipendenti
    #D.gamma = t(tmp)%*%tmp #matrice varianza-covarianza a priori per
        theta
    dev.gamma = chol2inv(chol(t(tmp)%*%diag(w[i-1,])%*%tmp+1/sigma2[i-1]
        *chol2inv(chol(D.gamma))))

    psi.forward[h] = dbinom(sum(draw[h,]),p,phi,log=T)+log.marginal(tmp,
        y,w[i-1,],sigma2[i-1],dev.gamma,D.gamma)-lchoose(p,h)
  }

  #scelta di $\gamma^{(j)}$
  best=sample(1:k,1,prob=self.weights(psi.forward))
  gamma.best = draw[best,]

  #k-1 proposte all'indietro
```

```
draw.back=matrix(rep(gamma.best,k),k,p,byrow=T)

for(h in ind[-best]){

  draw.back[h,]=proposal(gamma.best,h)

  tmp=x[,draw.back[h,]==1]

  D.gamma=diag(1,sum(draw.back[h,])) #matrice varianza-covarianza a
      priori per theta a componenti indipendenti
  #D.gamma=t(tmp)%*%tmp #matrice varianza-covarianza a priori per
      theta
  dev.gamma=chol2inv(chol(t(tmp)%*%diag(w[i-1,])%*%tmp+1/sigma2[i-1]*
      chol2inv(chol(D.gamma))))

  #peso w in avanti
  psi.backward[h]=dbinom(sum(draw.back[h,]),p,phi,log=T)+log.marginal(
      tmp,y,w[i-1,],sigma2[i-1],dev.gamma,D.gamma)-lchoose(p,h)
}

D.gamma = diag(1,sum(gamma[i-1,]))
dev.gamma = chol2inv(chol(t(x[,gamma[i-1,]==1])%*%diag(w[i-1,])%*%x[,
    gamma[i-1,]==1]+1/sigma2[i-1]*chol2inv(chol(D.gamma))))

psi.backward[best] = dbinom(sum(gamma[i-1,]),p,phi,log=T)+log.marginal
    (x[,gamma[i-1,]==1],y,w[i-1,],sigma2[i-1],dev.gamma,D.gamma)-
    lchoose(p,best)

#normalizzazione pesi esponenziali
psi = self.weights(c(psi.backward,psi.forward))

#probabilita accettazione
alpha = min(1,sum(psi[(k+1):(2*k)])/sum(psi[1:k]))

#aggiornamento dei parametri
y.tilde = (y-0.5)/w[i-1,]
```

```
    r=runif(1)

    if(r<=alpha){
      gamma[i,] = gamma.best
      c[i] = 1

      x.gamma = x[,gamma[i,]==1]
      D.gamma = diag(1,sum(gamma[i,])) #indipendenza a-priori
      #D.gamma = t(x.gamma)%*%x.gamma #correlazione a-priori
      dev.bb = chol2inv(chol(t(x.gamma)%*%diag(w[i-1,])%*%x.gamma+1/sigma2
          [i-1]*chol2inv(chol(D.gamma))))
      mean.bb = dev.bb%*%(t(x.gamma)%*%diag(w[i-1,])%*%y.tilde)
      theta[i,gamma[i,]==1] = rmvnorm(1, mean=mean.bb, sigma=dev.bb)

      sigma2[i] = 1/rgamma(1,(lambda+sum(gamma[i,])/2),(v+0.5*sum(theta[i
          ,]^2)))

      w[i,] = pgdraw(1,x.gamma%*%theta[i,gamma[i,]==1])
    }

    if(r>alpha){
      gamma[i,] = gamma[i-1,]
      theta[i,] = theta[i-1,]
      sigma2[i]=sigma2[i-1]
      w[i,]=w[i-1,]
    }
  }

  return(list(gamma=gamma,sigma=sigma2,theta=theta,c=c))
}
```

Appendice B

Codice R delle simulazioni condotte

Codice B.1: Esempio: Dati simulati

```
### Pacchetti necessari ###
library(pgdraw)
library(msos)
library(mvtnorm)
library(parallel)
library(ggplot2)
library(plotROC)

### Simulazione dataset ###
#n osservazioni
#p variabili
#p0 complessita vero modello
#balance frequenza classe 1

dati.sim = function(n,p,p0,balance){
  x1 = matrix(rnorm(p*n*5),ncol=p))

  beta = c(rep(0,p))
  sed = c(sample(1:p,p0))
  beta[sed] = runif(p0,2,5)
```

```
  z = x1%*%beta
  pr = 1/(1+exp(-z))

  y1 = rbinom(n*5,1,pr)

  uno = which(y1==1)
  uno = sample(uno,round(0.balance*n))
  zero = which(y1==0)
  zero = sample(zero,round((1-balance)*n)))

  x = x1[c(zero,uno),]
  y = y1[c(zero,uno)]

  return(list(x=x,y=y,beta=beta))
}

### Pre-elaborazione dati ###
n = 200
p = 300
p0 = 25

dati = dati.sim(n,p,p0)
x = cbind(rep(1,n),dati$x)
y = dati$y
beta = dati$beta

#vero modello
true.model = rep(0,p)
true.model[which(beta!=0)] = 1

#modello iniziale casuale
gamma0 = rep(0,p+1)
gamma0[1] = 1
gamma0[c(sample(2:(p+1),10))] = 1
```

```
#vettore variabile latente $\omega$ iniziale
w0 = rep(0.5,n)

#iperparametro per beta
sigma0 = 2
v0 = 0.1
v1 = 100
D = diag(1,p,p)
#D = t(x)%*%x

#iperparametri per sigma
lambda = 2
v = 1

rm(dati)

### Elaborazione dati in parallelo ###
cl = makeCluster(12,type="FORK")

tasks = list(
  job1 = function() gibbs(5000,y,x,c(lambda,v),0.1,v0,v1,D),
  job2 = function() gibbs(5000,y,x,c(lambda,v),0.3,v0,v1,D),
  job3 = function() gibbs(5000,y,x,c(lambda,v),0.5,v0,v1,D),
  job4 = function() rj(30000,as.numeric(y.stima),as.matrix(x.stima),w0
      ,0.1,c(lambda,v),gamma0,sigma0),
  job5 = function() rj(30000,as.numeric(y.stima),as.matrix(x.stima),w0
      ,0.3,c(lambda,v),gamma0,sigma0),
  job6 = function() rj(30000,as.numeric(y.stima),as.matrix(x.stima),w0
      ,0.5,c(lambda,v),gamma0,sigma0),
  job7 = function() rjmt(30000,as.numeric(y.stima),as.matrix(x.stima),w0
      ,0.1,10,c(lambda,v),gamma0,sigma0),
  job8 = function() rjmt(30000,as.numeric(y.stima),as.matrix(x.stima),w0
      ,0.3,10,c(lambda,v),gamma0,sigma0),
  job9 = function() rjmt(30000,as.numeric(y.stima),as.matrix(x.stima),w0
      ,0.5,10,c(lambda,v),gamma0,sigma0),
  job10 = function() rjmt(30000,as.numeric(y.stima),as.matrix(x.stima),w0
```

```
      ,0.1,30,c(lambda,v),gamma0,sigma0),
  job11 = function() rjmt(30000,as.numeric(y.stima),as.matrix(x.stima),w0
      ,0.3,30,c(lambda,v),gamma0,sigma0),
  job12 = function() rjmt(30000,as.numeric(y.stima),as.matrix(x.stima),w0
      ,0.5,30,c(lambda,v),gamma0,sigma0),
)

out = clusterApply(
  cl,
  tasks,
  function(f) f()
)

stopCluster(cl)

### Visualizzazione risultati ###

### Modelli visitati ###
d = c(2^(0:(p-1)))
R = lapply(out, function(x) as.character(x$gamma%*%d1+1))
par(mfrow=c(3,3))

pdf("modelli_visitati.pdf")
for(i in 1:length(list)){
        ggplot(data=data.frame(modello=as.numeric(R[[i]]),i=1:length(R[[i
            ]])), aes(x=i, y=modello)) + geom_point() +
  theme_gray(base_size = 22)+ geom_hline(yintercept=as.numeric(true.model%
      *%dq+1),size=1.25,col="red",linetype="dashed") + theme(axis.title.x=
      element_text(size=20),axis.ticks.x=element_blank(),axis.text.x =
      element_text(size=20),axis.title.y=element_text(size=20), axis.text.
      y = element_blank()) + labs(title="",x="Iterazione",y="Modello") +
      scale_y_continuous(breaks = NULL)
dev.off()

### Distribuzione a-posteriori post burn-in di $\gamma$ ###
pdf("gamma_distribuzione.pdf")
```

```
for(i in 1:length(list)){
        ggplot(data=as.data.frame(table(R[[i]][-c(1:1000)])),aes(x=Var1,y=
            Freq)) + geom_bar(stat="identity",fill="black") + ylim(0,70) +
             theme_gray(base_size=22) + theme(axis.title.x=element_text(
            size=20),axis.ticks.x=element_blank(),axis.text.x=element_
            blank(),axis.title.y=element_text(size=20), axis.text.y=
            element_text(size=20)) + labs(title="",x="Modello",y="
            Frequenza")
}
dev.off()

### Probabilita di inclusione a-posteriori delle variabili ###
post.prob = matrix(0,length(list),p)
for(i in 1:3){
        post.prob[i,] = apply(out[[i]]$gamma[-c(1:1000),],2,mean)
}
for(i in 4:length(list)){
        post.prob[i,] = apply(out[[i]]$gamma[-c(1:5000),],2,mean)
}

pdf("prob_posteriori.pdf")
for(i in 1:length(list)){
        ggplot(data=data.frame(Var1=post.prob[i,],i=1:p),aes(y=Var1,x=i,
            ymin=Var1,ymax=0)) +
  geom_linerange() + ylim(0,1) + theme_gray(base_size = 22) + theme(axis.
      title.x=element_text(size=20),axis.ticks.x=element_blank(),axis.text
      .x = element_blank(),axis.title.y=element_text(size=20), axis.text.y
       = element_text(size=20)) + labs(title="",x="Variabile",y="
      Probabilita Inclusione")
}
dev.off()

### Curva ROC ###
roc.data = list()
roc.data[[1]] = melt_roc(data.frame(D=true.model,RJ=post.prob[4,],MT_k10=
    post.prob.mtx[7,],MT_k30=post.prob[10,]),"D",c("RJ","MT_k10","MT_k30")
```

```
    )
roc.data[[2]] = melt_roc(data.frame(D=true.model,RJ=post.prob[5,],MT_k10=
    post.prob.mtx[8,],MT_k30=post.prob[11,]),"D",c("RJ","MT_k10","MT_k30")
    )
roc.data[[3]] = melt_roc(data.frame(D=true.model,RJ=post.prob[6,],MT_k10=
    post.prob.mtx[9,],MT_k30=post.prob[12,]),"D",c("RJ","MT_k10","MT_k30")
    )
roc.data[[4]] = melt_roc(data.frame(D=true.model,phi_0.1=post.prob[1,],phi
    _0.3=post.prob[2,],phi_0.5=post.prob[3,]),"D",c("phi_0.1","phi_0.3","
    phi_0.5"))

pdf("roc_plot.pdf")
for(i in 1:length(list)){
        ggplot(roc.data[[i]],aes(d=D,m=M,color=name)) + geom_roc(n.cuts=0,
            size=3) + style_roc(theme=theme_grey,xlab="1-Specificita",ylab
            ="Sensibilita")+ theme(legend.title=element_text(size=20),
            legend.text=element_text(size=20),axis.text=element_text(size
            =20),axis.title=element_text(size=20)) + scale_color_brewer(
            palette="Set1")
}
dev.off()

### Convergenza $\sigma^2$ ###
sigma_data = list()
sigma_data[[1]] = data.frame(media=c(cumsum(out[[1]]$sigma)/(1:5000),
    cumsum(out[[2]]$sigma)/(1:5000),cumsumout[[3]]$sigma)/(1:5000)),i=rep
    (1:5000),3),algoritmo=as.character(c(rep("RJ",5000),rep("MT_k10",5000)
    ,rep("MT_k30",5000))))
sigma[[2]] = data.frame(media=c(cumsum(out[[4]]$sigma)/(1:30000),cumsum(
    out[[7]]$sigma)/(1:30000),cumsum(out[[10]]$sigma)/(1:30000)),i=rep
    (1:30000,3),algoritmo=as.character(c(rep("RJ",30000),rep("MT_k10"
    ,30000),rep("MT_k30",30000))))
sigma[[3]] = data.frame(media=c(cumsum(out[[5]]$sigma)/(1:30000),cumsum(
    out[[8]]$sigma)/(1:30000),cumsum(out[[11]]$sigma)/(1:30000)),i=rep
    (1:30000,3),algoritmo=as.character(c(rep("RJ",30000),rep("MT_k10"
    ,30000),rep("MT_k30",30000))))
```

```
sigma[[4]] = data.frame(media=c(cumsum(out[[6]]$sigma)/(1:30000),cumsum(
    out[[9]]$sigma)/(1:30000),cumsum(out[[12]]$sigma)/(1:30000)),i=rep
    (1:30000,3),algoritmo=as.character(c(rep("RJ",30000),rep("MT_k10"
    ,30000),rep("MT_k30",30000))))

pdf("sigma_plot.pdf")
for(i in 1:length(sigma_data)){
        ggplot(data=sigma_data[[i]],aes(x=i,y=media)) + ylim(0,5) + geom_
            point(aes(col=algoritmo),size=2)+
  theme_gray(base_size = 22) + theme(legend.title=element_text(size=20),
      legend.text=element_text(size=20),axis.title.x=element_text(size=20)
      ,axis.ticks.x=element_blank(),axis.text.x =element_text(size=20),
      axis.title.y=element_text(size=20),axis.text.y=element_text(size=20)
      ) + labs(title="",x="Iterazione",y="Media") + scale_color_brewer(
      palette="Set1")
}
dev.off()

### Convergenza $\theta$ ###
beta.data = list()
beta.data[[1]] = data.frame(theta=c(rep("theta39",5000),rep("theta61"
    ,5000),rep("theta67",5000),rep("theta39",5000),rep("theta61",5000),rep
    ("theta67",5000),rep("theta39",5000),rep("theta61",5000),rep("theta67"
    ,5000)),media=c(cumsum(out[[1]]$beta[,39])/(1:5000),cumsum(out[[1]]$
    beta[,61])/(1:5000),cumsum(out[[1]]$beta[,62])/(1:5000),cumsum(out
    [[2]]$theta[,39])/(1:5000),cumsum(out[[2]]$theta[,61])/(1:5000),cumsum
    (out[[2]]$theta[,67])/(1:5000),cumsum(out[[3]]$theta[,39])/(1:5000),
    cumsum(out[[3]]$theta[,61])/(1:5000),cumsum(out[[3]]$theta[,67])/
    (1:5000)),i=rep(1:5000,9),algoritmo=as.character(c(rep("RJ",15000),rep
    ("MT_k10",15000),rep("MT_k30",15000))))
beta.data[[2]] = data.frame(theta=c(rep("theta39",30000),rep("theta61"
    ,30000),rep("theta67",30000),rep("theta39",30000),rep("theta61",30000)
    ,rep("theta67",30000),rep("theta39",30000),rep("theta61",30000),rep("
    theta67",30000)),media=c(cumsum(out[[4]]$beta[,39])/(1:30000),cumsum(
    out[[4]]$beta[,61])/(1:30000),cumsum(out[[4]]$beta[,62])/(1:30000),
    cumsum(out[[7]]$theta[,39])/(1:30000),cumsum(out[[7]]$theta[,61])/
```

```
    (1:30000),cumsum(out[[7]]$theta[,67])/(1:30000),cumsum(out[[10]]$theta
    [,39])/(1:30000),cumsum(out[[10]]$theta[,61])/(1:30000),cumsum(out
    [[10]]$theta[,67])/(1:30000)),i=rep(1:30000,9),algoritmo=as.character(
    c(rep("RJ",90000),rep("MT_k10",90000),rep("MT_k30",90000))))
beta.data[[3]] = data.frame(theta=c(rep("theta39",30000),rep("theta61"
    ,30000),rep("theta67",30000),rep("theta39",30000),rep("theta61",30000)
    ,rep("theta67",30000),rep("theta39",30000),rep("theta61",30000),rep("
    theta67",30000)),media=c(cumsum(out[[5]]$beta[,39])/(1:30000),cumsum(
    out[[5]]$beta[,61])/(1:30000),cumsum(out[[5]]$beta[,62])/(1:30000),
    cumsum(out[[8]]$theta[,39])/(1:30000),cumsum(out[[8]]$theta[,61])/
    (1:30000),cumsum(out[[8]]$theta[,67])/(1:30000),cumsum(out[[11]]$theta
    [,39])/(1:30000),cumsum(out[[11]]$theta[,61])/(1:30000),cumsum(out
    [[11]]$theta[,67])/(1:30000)),i=rep(1:30000,9),algoritmo=as.character(
    c(rep("RJ",90000),rep("MT_k10",90000),rep("MT_k30",90000))))
beta.data[[4]] = data.frame(theta=c(rep("theta39",30000),rep("theta61"
    ,30000),rep("theta67",30000),rep("theta39",30000),rep("theta61",30000)
    ,rep("theta67",30000),rep("theta39",30000),rep("theta61",30000),rep("
    theta67",30000)),media=c(cumsum(out[[6]]$beta[,39])/(1:30000),cumsum(
    out[[6]]$beta[,61])/(1:30000),cumsum(out[[6]]$beta[,62])/(1:30000),
    cumsum(out[[9]]$theta[,39])/(1:30000),cumsum(out[[9]]$theta[,61])/
    (1:30000),cumsum(out[[9]]$theta[,67])/(1:30000),cumsum(out[[12]]$theta
    [,39])/(1:30000),cumsum(out[[12]]$theta[,61])/(1:30000),cumsum(out
    [[12]]$theta[,67])/(1:30000)),i=rep(1:30000,9),algoritmo=as.character(
    c(rep("RJ",90000),rep("MT_k10",90000),rep("MT_k30",90000))))

pdf("beta_plot.pdf")
for(i in 1ength(beta_data)){
        ggplot(data=beta.data[[i]],aes(x=i,y=media)) + geom_line(aes(
            linetype=theta, color=algoritmo),size=2) + ylim(c(-0.5,3)) +
  theme_gray(base_size = 22) + theme(legend.title=element_text(size=20),
      legend.text=element_text(size=20),axis.title.x=element_text(size=20)
      ,axis.ticks.x=element_blank(),axis.text.x =element_text(size=20),
      axis.title.y=element_text(size=20), axis.text.y=element_text(size
      =20)) + labs(title="",x="Iterazione",y="Media") + scale_color_brewer
      (palette="Set1")
}
```

```
dev.off()
```

Codice B.2: Esempio: Applicazione a dati reali

```
### Lettura dati e preparazione del dataset ###
dati = read.table(file.choose())
y = as.numeric(y)
x = cbind(rep(1,n),dati[,-which(colnames(dati)==y)])

n = length(y)
p = ncol(x)
balance = table(y)[[1]]/n

#procedimento per dividere il dataset in stima e verifica mantenendo il
    bilanciamento delle classi
#n.stima numero di osservazioni usate per la stima
uno = which(y==1)
uno = sample(uno,round(balance*n.stima))
zero = which(y==0)
zero = sample(zero,round((1-balance)*n.stima)))

x.stima = x[c(uno,zero),]
y.stima = y[c(uno,zero)]
x.test = x[-c(uno,zero),]
y.test = y[-c(uno,zero)]

### Inizializzazione parametri ###
gamma0 = rep(0,p)
gamma0[1] = 1
gamma0[c(sample(1:p,10))] = 1
w0 = rep(0.5,n.stima)
lambda = 2
v = 1
sigma0 = 2
v0 = 0.1
```

```
v1 = 100
D = diag(1,p,p)

### Elaborazione dati in parallelo ###
cl = makeCluster(4,type="FORK")

tasks = list(
  job1 = function() gibbs(5000,y,x,c(lambda,v),0.5,v0,v1,D),
  job2 = function() rj(15000,as.numeric(y.stima),as.matrix(x.stima),w0
      ,0.5,c(lambda,v),gamma0,sigma0),
  job3 = function() rjmt(15000,as.numeric(y.stima),as.matrix(x.stima),w0
      ,0.5,10,c(lambda,v),gamma0,sigma0),
  job4 = function() rjmt(15000,as.numeric(y.stima),as.matrix(x.stima),w0
      ,0.5,30,c(lambda,v),gamma0,sigma0),
)

out = clusterApply(
  cl,
  tasks,
  function(f) f()
)

stopCluster(cl)

### Funzione predittiva###
#n.p numero modelli su cui valutare la previsione
n.p=200

R = list()
R[[1]] = as.character(out[[1]]$gamma%*%d1+1)
R[[2]] = as.character(out[[2]]$gamma%*%d1+1)
R[[3]] = as.character(out[[3]]$gamma%*%d1+1)
R[[4]] = as.character(out[[4]]$gamma%*%d1+1)

pred = list()
pred[[1]] = table(R[[1]][-c(1:1000)])
```

```
pred[[2]] = table(R[[2]][-c(1:5000)])
pred[[3]] = table(R[[3]][-c(1:5000)])
pred[[4]] = table(R[[4]][-c(1:5000)])

beta_pred = list()
beta_pred[[1]] = out[[i]]$theta[sample(which(R[[1]][-c(1:1000)] %in% names
    (pred[[1]]),n.p)+1000,]
for(i in 2:length(pred)){
        beta_pred[[i]]=out[[i]]$theta[sample(which(R[[i]][-c(1:5000)] %in%
             names(pred),n.p)+5000,]
}

prev = list()
for(i in 1:length(pred)){
        prev[[i]] = exp(beta_pred[[i]]%*%t(x.test))/(1+exp(beta_pred[[i]]%
            *%t(x.test)))
}

prob = lapply(prev,function(x) apply(x,2,function(y) sum(y>0.5)/n.p))

roc.data=melt_roc(data.frame(D=y.test,GS=prob[[1]],RJ=prob[[2]],MT_k10=
    prob[[3]],MT_k30=prob[[4]]),"D",c("GS","RJ","MT_k10","MT_k30"))

pdf("roc.pdf")
ggplot(roc.data,aes(d=D,m=M,color=name)) + geom_roc(n.cuts=0,size=3) +
    style_roc(theme = theme_grey, xlab="1-Specificita",ylab="Sensibilita")
     + theme(legend.title=element_text(size=20),legend.text=element_text(
    size=20),axis.text=element_text(size=20),axis.title=element_text(size
    =20)) + scale_color_brewer(palette="Set1")
dev.off()

### Indice di Youden ###
y1=coords(roc(y.test,prob[[1]]),"b")
y2=coords(roc(y.test,prob[[2]]),"b")
y3=coords(roc(y.test,prob[[3]]),"b")
y4=coords(roc(y.test,prob[[4]]),"b")
```

```
table(y.test,prob[[1]]>y1)
table(y.test,prob[[2]]>y2)
table(y.test,prob[[3]]>y3)
table(y.test,prob[[4]]>y4)
```

Ringraziamenti

Arrivati alla fine di questo lavoro e di questo percorso, qualche momento di riflessione è doveroso e naturale. Non posso nominare tutte le persone e le sfide che ho incontrato e che hanno reso interessanti e dato sapore a questi cinque anni e mezzo. Tuttavia, ci sono persone che desidero siano citate in questa tesi.

Il ringraziamento più grande e sincero va ai miei genitori e alla mia famiglia, che mi ha sempre sostenuto di fronte a qualsiasi scelta, facile o difficile che fosse, e senza i quali non sarei qui a scrivere questo lavoro; per l'enorme fiducia che mi è stata concessa e per avermi permesso di intraprendere nuove esperienze fuori casa e all'estero, aiutandomi a capire quale fosse la mia strada.

Un ulteriore ringraziamento va al prof. Bernardi, per avermi permesso di lavorare al meglio, per la disponibilità mostrata e per l'aiuto concessomi, e ad Andrea per i chiarimenti sui particolari e per l'aiuto pratico ricevuto.

Agli amici di una vita di Salzano, per tutte le serate e le partitine memorabili al campetto, e ai compagni di Padova, per le partite a carte che hanno alleviato le lunghe giornate in aula studio.

Infine, un ringraziamento ai sistemisti Paolo Scopelliti e Vincenzo Agosto per il lavoro di supporto e per le risorse computazionali senza le quali questa tesi sarebbe risultata impossibile.