

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



Stima non distorta in mediana del modello di Rasch

Relatore Prof. Nicola Sartori
Dipartimento di Scienze Statistiche

Correlatore Dott. Euloge Clovis Kenne Pagui
Dipartimento di Scienze Statistiche

Laureando Edoardo Michielon
Matricola N 1103862

Anno Accademico 2015/2016

Ai miei genitori

Indice

| | |
|--|-----------|
| Introduzione | 5 |
| 1 Letteratura di riferimento | 9 |
| 1.1 Richiami di teoria dell'inferenza | 9 |
| 1.1.1 Notazione indiciale | 9 |
| 1.1.2 Momenti e cumulanti | 10 |
| 1.1.3 Espansioni asintotiche ed efficient score | 13 |
| 1.1.4 Espansioni di Edgeworth e di Cornish Fisher | 15 |
| 1.2 Stimatori non distorti in media | 17 |
| 1.3 Stimatori non distorti in mediana | 20 |
| 2 Il modello di Rasch | 27 |
| 2.1 Introduzione al modello di Rasch | 27 |
| 2.2 Modello di Rasch ad uno o due parametri | 28 |
| 2.3 Metodi di stima alternativi e problemi associati | 31 |
| 3 Stima non distorta in mediana del modello di Rasch | 37 |
| 3.1 Modello di Rasch ad un parametro | 37 |
| 3.1.1 Matrice dei dati | 37 |
| 3.1.2 Verosimiglianza e momenti nulli | 38 |
| 3.1.3 Calcolo dei cumulanti k_1 , k_2 e k_3 | 40 |
| 3.1.4 Algoritmo di stima | 45 |
| 3.2 Modello di Rasch a due parametri | 46 |
| 3.2.1 Matrice dei dati | 46 |
| 3.2.2 Verosimiglianza e momenti nulli | 48 |
| 3.2.3 Calcolo dei cumulanti k_1 , k_2 e k_3 | 51 |

| | | |
|----------|--|------------|
| 3.2.4 | Algoritmo di stima | 55 |
| 4 | Studi di simulazione | 61 |
| 4.1 | Simulazioni per il modello logistico ad un parametro | 61 |
| 4.2 | Simulazioni per il modello logistico a due parametri | 65 |
| 5 | Applicazione ai dati dei test d'ingresso | 69 |
| | Conclusioni | 75 |
| A | Rcpp | 79 |
| A.1 | Introduzione a Rcpp | 79 |
| A.2 | Un primo esempio | 80 |
| A.3 | RcppArmadillo | 83 |
| B | Codice R per le funzioni di stima | 85 |
| C | Codice R per la modificazione | 97 |
| | Bibliografia | 103 |
| | Ringraziamenti | 109 |

Introduzione

"There is nothing as practical as a good theory"

K. Lewin

Il principale strumento per l'inferenza nel paradigma decisionale frequentista è la funzione di verosimiglianza che, a livello asintotico, gode di proprietà desiderabili fra cui la non distorsione del corrispondente stimatore. Tuttavia per campioni con ridotta numerosità le stime derivate dall'equazione di verosimiglianza, detta *score function*, possono condurre ad errori elevati. Un'altra criticità di questo approccio è costituito dalle ipotesi restrittive alla base, la cui violazione comporta la perdita di buona parte dell'impianto teorico a supporto della metodologia. Neyman e Scott (1948) portarono alla luce una tipologia di modelli in cui il numero di parametri di disturbo, ovvero quelli non di interesse per l'inferenza, crescono all'aumentare della numerosità del campione e le stime di massima verosimiglianza dell'intero vettore dei parametri risultano non consistenti.

Il modello di Rasch rientra per certi aspetti nella tipologia descritta da Neyman e Scott. Inizialmente nato per la valutazione della capacità di lettura dei bambini (Rasch, 1960), il modello è ancora un riferimento in statistica e psicometria per i dati generati da test attitudinali. La sua peculiarità è la specifica della relazione che lega la probabilità di rispondere in modo corretto sia con l'abilità del candidato sia con la difficoltà della domanda, rappresentata da uno o più parametri. All'aumentare, quindi, dei soggetti o delle domande considerate cresce di pari passo il numero di parametri introdotti; inoltre il modello, per costruzione stessa, può presentare casi di perfetta separabilità dei dati che implica la non esistenza delle stime di massima verosimiglianza. Vari autori hanno studiato le proprietà degli stimatori classici applicati in

questo contesto e proposto soluzioni alternative, si veda Andersen (1980) per una trattazione completa.

Una possibile soluzione a questo problema è lo stimatore non distorto in media di Firth (1993) e Kosmidis e Firth (2009). Attraverso una semplice modificazione della *score function* gli autori definiscono un nuovo stimatore che non si basa sull'esistenza della stima di massima verosimiglianza e che risulta non distorto al secondo ordine. Nel caso della regressione logistica, quindi anche per il modello di Rasch, è dimostrata l'esistenza ed unicità della stima. Basandosi su questo lavoro Kenne Pagui *et al.* (2016a) hanno sviluppato uno stimatore non distorto in mediana ridefinendo la modificazione originale. Il nuovo stimatore ha fra i suoi vantaggi l'invarianza per trasformazioni che operano sugli elementi di θ in modo separato..

L'obiettivo del lavoro di tesi consiste nello studio del nuovo stimatore non ancora presente in letteratura, in particolare si sviluppa la stima non distorta in mediana per il modello di Rasch. L'obiettivo primario è l'applicazione della teoria relativa alle famiglie esponenziali al contesto specifico e la creazione di una funzione scritta nel linguaggio R capace di sfruttare al meglio la caratteristica sparsità nei dati per stimare in modo efficiente entrambe le specifiche del modello. In secondo luogo si vuole verificare tramite simulazione le proprietà dello stimatore relativamente agli altri metodi di stima utilizzati in letteratura.

Nel capitolo 1 si introducono gli aspetti teorici utili per le analisi successive. All'interno della teoria classica dell'inferenza sono richiamate la definizione e le proprietà dei momenti nulli e dei cumulanti, gli sviluppi asintotici delle principali quantità di verosimiglianza, dai quali si ricava la definizione di *efficient score*, e le approssimazioni di distribuzioni attraverso le espansioni di Edgeworth e di Cornish-Fisher. Per meglio evidenziare analogie e differenze sono introdotti i lavori di Firth (1993) e Kenne Pagui *et al.* (2016a) uno di seguito all'altro.

Il Capitolo 2 è dedicato all'analisi del modello di Rasch a partire dal lavoro originale del suo autore per arrivare alle due specifiche con cui oggi è maggiormente noto. Sono presentate le funzioni di verosimiglianza associate e le interpretazioni dei parametri inseriti. Si discute in seguito dei vari metodi presenti in letteratura per la stima del metodo, mettendo in evidenza le

criticità cui vanno incontro.

Il nucleo di questo studio è proposto nel Capitolo 3, dove si sviluppano i calcoli necessari per la stima del modello di Rasch tramite lo stimatore non distorto in mediana. Il capitolo è diviso in due parti che rispecchiano le due specifiche introdotte; ogni paragrafo affronta il problema a partire dalla costruzione della matrice del disegno, prosegue col calcolo della verosimiglianza e della modificazione e ha come risultato finale la proposta di un algoritmo di stima implementabile.

Il Capitolo 4 è dedicato agli studi di simulazione condotti per verificare le proprietà distributive del nuovo stimatore. Per la prima specifica del modello di Rasch il confronto è relativo ad altri quattro metodi adatti per la stima, mentre per la seconda versione l'unico stimatore presente in letteratura è quello non distorto in media che sarà preso come *golden standard*.

Per concludere, nel Capitolo 5 è proposto un esempio di applicazione del metodo a dei dati reali. Si sono analizzate le risposte al quesito relativo alla comprensione linguistica del test d'ingresso per il Dipartimento di Scienze Statistiche, con l'obiettivo di valutare la difficoltà delle domande e la abilità dei candidati.

Capitolo 1

Letteratura di riferimento

"Although this may seem a paradox, all exact science is dominated by the idea of approximation"

B. Russel

Nel presente capitolo si vuole richiamare all'attenzione gli aspetti della teoria classica dell'inferenza che saranno utilizzati lungo tutto il lavoro di tesi. Costituisce, insieme al Capitolo 2, un riferimento teorico per il corpo centrale della tesi sviluppato nei Capitoli 3 e 4. Si introduce la notazione indiciale adottata per la scrittura delle formule e si passano in rassegna varie quantità di verosimiglianza e sviluppi asintotici. Infine si presentano i lavori di Firth (1993) e Kenne Pagui *et al.* (2016a) sulle correzioni dello stimatore di massima verosimiglianza in media ed in mediana, rispettivamente.

1.1 Richiami di teoria dell'inferenza

1.1.1 Notazione indiciale

Si adotta per l'intero testo la notazione indiciale per rappresentare vettori, matrici e tensori. Questa convenzione permette di rappresentare oggetti algebrici tramite i loro elementi, generalizzando la scrittura $a = [a_r]$ per $r = 1, \dots, p$. Ad esempio:

| | |
|----------------------|--|
| θ_r, l^s | indicano un elemento di un vettore ($p \times 1$) |
| i_{rs}, i^{rs} | indicano un elemento di una matrice ($p \times p$) |
| l_{rst}, ν^{rst} | indicano un elemento di un tensore ($p \times p \times p$) |

Un indice posto come pedice è detto *covariant index* ed uno posto come apice *contravariant index*; questi ultimi nel caso di matrici indicano che bisogna considerare l'elemento (r, s) -esimo della matrice inversa. Questa notazione e la rispettiva nomenclatura derivano dalla geometria differenziale. Il vantaggio è la forte semplificazione nella rappresentazione di oggetti multivariati che possono essere trattati nello stesso modo di analoghi oggetti univariati. Ad esempio il prodotto esterno fra due vettori colonna $x = [x_i]$ e $y = [y_j]$ in \mathbb{R}^p $A = [a_{ij}] = xy^T = [x_i y_j]$ si può scrivere in modo più compatto come $a_{ij} = x_i y_j$ lasciando implicito il range di variazione degli indici. In questo contesto si è soliti adottare anche la convenzione di somma di Einstein per la quale se un indice si presenta due o più volte in un termine di una espressione occorre sommare rispetto ad esso, salvo il caso che sia esplicitamente indicato il contrario. Sia il simbolo di sommatoria sia il range degli indici rimane implicito. Con questa convenzione si può scrivere, per esempio, la forma quadratica $x^T A x$ nel seguente modo compatto $a_{ij} x^i y^j$. Tutti gli indici che si ripetono all'interno dell'espressione sono detti indici muti (*dummy index*) per differenziarli dagli indici liberi ed è sempre possibile rinominare gli indici muti senza alterare il risultato.

1.1.2 Momenti e cumulanti

Sia Y una variabile casuale unidimensionale con densità $p_Y(y)$, la funzione generatrice dei momenti (m.g.f.) di Y è

$$M_Y(t) = \mathbb{E}[e^{tY}], \quad t \in \mathbb{R}.$$

Se Y è tale che $M_Y(t)$ esiste finita in un intorno di $t = 0$, allora vale che Y ha momenti finiti $\mu_r = \mathbb{E}[Y^r]$ per ogni $r = 1, 2, \dots$ e $\mu_r = \frac{\partial^r}{\partial t^r} M_Y(t)|_{t=0}$. Questo risultato si può dimostrare attraverso l'espansione in serie di potenze di $M_Y(t)$

$$M_Y(t) = 1 + \mu_1 t + \mu_2 \frac{t^2}{2!} + \mu_3 \frac{t^3}{3!} + \mu_4 \frac{t^4}{4!} + \dots$$

Se $M_Y(t) < +\infty$ per $|t| < t_0$ con $t_0 > 0$, allora si definisce la funzione generatrice dei cumulanti (c.g.f.)

$$K_Y(t) = \log M_Y(t).$$

Analogamente a prima si sviluppa in serie la funzione $K_Y(t)$

$$K_Y(t) = k_1 t + k_2 \frac{t^2}{2!} + k_3 \frac{t^3}{3!} + k_4 \frac{t^4}{4!} + \dots$$

da cui si ricavano i cumulanti di ordine r di Y : $k_r = \frac{\partial^r}{\partial t^r} K_Y(t)|_{t=0}$.

I cumulanti si possono riscrivere in funzione dei momenti o dei momenti centrali

$$\begin{aligned} k_1 &= \mu_1, \\ k_2 &= \mu_2 - \mu_1^2 = \bar{\mu}_2, \\ k_3 &= \mu_3 - 3\mu_1\mu_2 + 2\mu_1^3 = \bar{\mu}_3, \\ k_4 &= \mu_4 - 3\mu_2^2 - 4\mu_1\mu_3 + 12\mu_1^2\mu_2 - 6\mu_1^4 = \bar{\mu}_4 - 3\bar{\mu}_2^2, \end{aligned}$$

da cui emerge che i primi quattro momenti sono misure descrittive di posizione, variabilità, asimmetria e curtosi. Cumulanti di ordine maggiore forniscono ulteriori indicazioni se confrontati con i rispettivi della Normale. Spesso è conveniente lavorare con i cumulanti standardizzati, definiti da

$$\rho_r = \frac{k_r}{k_2^{r/2}}, \quad r = 3, 4, \dots \quad (1.1)$$

Sia $S_n = \sum Y_i$ dove Y_i sono copie indipendenti ed identicamente distribuite (i.i.d.) di Y , allora si possono ottenere le funzioni generatrici dei momenti e dei cumulanti di S_n dalle relazioni

$$M_{S_n}(t) = (M_Y(t))^n \quad \text{e} \quad K_Y(t) = nK_Y(t). \quad (1.2)$$

Se invece si applica una trasformazione affine ad Y tale che $X = (Y + a)/b$,

allora le nuove funzioni sono date dalle seguenti formule

$$M_X(t) = e^{\frac{a}{b}t} M_Y\left(\frac{t}{b}\right) \quad \text{e} \quad K_X(t) = \frac{a}{b}t + K_Y\left(\frac{t}{b}\right). \quad (1.3)$$

Sia la funzione generatrice dei momenti sia quella dei cumulanti definiscono in modo univoco la distribuzione di Y . Esse rivestono un ruolo importante nel calcolo di distribuzioni sia in modo esatto che approssimato (Pace & Salvan, 1997, Capitolo 3).

Sia $l(\theta) = \log p_Y(y; \theta)$ la log-verosimiglianza associata ad un modello statistico con densità $p_Y(y; \theta)$, $\theta \in \Theta \subseteq \mathbb{R}$ e si chiami la generica derivata parziale di ordine m di $l(\theta)$ con

$$U_{R_m} = U_{r_1 \dots r_m}(\theta) = \frac{\partial^m}{\partial \theta_{r_1} \dots \partial \theta_{r_m}} l(\theta),$$

ad esempio, U_r , $r = 1, \dots, p$ indica il generico elemento del vettore dello *score* $U(\theta)$, mentre U_{rs}, U_{rst} indicano le derivate di ordine maggiore di $l(\theta)$. Si introducono le seguenti notazioni dei momenti nulli per indicare i valori attesi delle derivate di $l(\theta)$ sotto ipotesi di corretta specificazione del modello.

$$\begin{aligned} \nu_a &= \mathbb{E}[U_a(\theta)], \\ \nu_{ab} &= \mathbb{E}[U_{ab}(\theta)], \\ \nu_{a,b} &= \mathbb{E}[U_a(\theta)U_b(\theta)], \\ \nu_{abc} &= \mathbb{E}[U_{abc}(\theta)], \\ \nu_{a,bc} &= \mathbb{E}[U_a(\theta)U_{bc}(\theta)], \\ \nu_{a,b,c} &= \mathbb{E}[U_a(\theta)U_b(\theta)U_c(\theta)]. \end{aligned}$$

Con a, b, c, \dots indici che assumo valore in $1, \dots, p$. Queste quantità sono simmetriche sia rispetto alla permutazione dei multindici R_m , sia rispetto alla permutazione degli indici r_1, \dots, r_m fra loro. Ad esempio vale che $\nu_{a,bc} = \nu_{bc,a} = \nu_{cb,a}$.

I momenti nulli sono regolati da una serie di relazioni note come le relazioni di Bartlett (Bartlett, 1953), dove ognuna è ottenuta differenziando la

precedente rispetto ad un elemento di θ diverso. Ad esempio, si ha

$$\nu_a = 0, \quad (1.4)$$

$$\nu_{ab} + \nu_{a,b} = 0, \quad (1.5)$$

$$\nu_{abc} + \nu_{ab,c} + \nu_{ac,b} + \nu_{bc,a} + \nu_{a,b,c} = 0. \quad (1.6)$$

Infatti, sfruttando la possibilità di scambiare gli operatori di differenziazione ed integrazione, vale che

$$\begin{aligned} \frac{\partial}{\partial \theta^b} \nu_a &= \nu_{ab} + \nu_{a,b}, \\ \frac{\partial}{\partial \theta^c} \nu_{ab} &= \nu_{abc} + \nu_{ab,c}, \\ \frac{\partial}{\partial \theta^c} \nu_{a,b} &= \nu_{ac,b} + \nu_{a,bc} + \nu_{a,b,c}. \end{aligned}$$

1.1.3 Espansioni asintotiche ed efficient score

Si presentano brevemente i risultati degli sviluppi asintotici di quantità di verosimiglianza quali $\hat{\theta} - \theta$, $l(\hat{\theta}) - l(\theta)$ e per la funzione *score* profilo $\partial l_P(\tau) / \partial \tau$ utili per la presentazione degli stimatori nei Paragrafi 1.2 e 1.3. Il riferimento principale per questa parte è Pace & Salvan (1997).

Sia \mathcal{F} un modello parametrico con densità $p_Y(y; \theta)$, $\theta \in \Theta \subseteq \mathbb{R}$ e si assuma che esista la stima di massima verosimiglianza (m.l.e) $\hat{\theta}$ ottenuta come soluzione del sistema di equazioni di stima $U_a(\theta) = 0$, $a = 1, \dots, p$. Sotto condizioni di regolarità si può sviluppare $l(\hat{\theta})$ in serie di Taylor intorno a θ

$$0 = U_a(\hat{\theta}) = U_a + (\hat{\theta} - \theta)^b U_{ab} + \frac{1}{2} (\hat{\theta} - \theta)^{bc} U_{abc} + O_p(n^{-1/2}), \quad (1.7)$$

dove $(\hat{\theta} - \theta)^{R_m} = (\hat{\theta} - \theta)^{r_1} \dots (\hat{\theta} - \theta)^{r_m}$.

Nella formula (1.7) si può separare la parte stocastica dei coefficienti di U_{R_m} dalla parte non stocastica. A tale scopo di consideri la decomposizione $U_{R_m} = \nu_{R_m} + H_{R_m}$ dove H_{R_m} è la variabile centrata definita da $H_{R_m} =$

$U_{R_m} - \nu_{R_m}$, e si ricordi la relazione $\nu_{ab} = -i_{ab}$

$$0 = U^a + (\hat{\theta} - \theta)^b(-i_{ab} + H_{ab}) + \frac{1}{2}(\hat{\theta} - \theta)^{bc}(\nu_{abc} + H_{abc}) + O_p(n^{-1/2}), \quad (1.8)$$

dove si è adottata la notazione $U^r = i^{rs}U_s$. Dalla (1.8) si può ricavare l'espansione asintotica per la quantità $(\hat{\theta} - \theta)^a$

$$(\hat{\theta} - \theta)^a = U^a + H_b^a(\hat{\theta} - \theta)^b + \frac{1}{2}\nu_{bc}^a(\hat{\theta} - \theta)^{bc} + O_p(n^{-1}),$$

che può essere riordinata, tramite semplici passaggi algebrici, per ottenere una forma esplicita in

$$(\hat{\theta} - \theta)^a = U^a + \frac{1}{2}\nu_{bc}^a U^b U^c + H_b^a U^b + O_p(n^{-1}), \quad (1.9)$$

dove valgono le notazioni $\nu_{bc}^a = i^{at}\nu_{bct}$ e $H_b^a = i^{at}H_{bt}$. Grazie all'applicazione dello sviluppo di Taylor stocastico si ottiene lo sviluppo anche di $l(\hat{\theta})$ centrato, che può essere riscritto grazie alla (1.9) in

$$\begin{aligned} l(\hat{\theta}) - l(\theta) &= (\hat{\theta} - \theta)^a U_a + \frac{1}{2}(\hat{\theta} - \theta)^{ab} U_{ab} + \frac{1}{6}(\hat{\theta} - \theta)^{abc} U_{abc} + O_p(n^{-1/2}) \\ &= \frac{1}{2}i_{ab} U^{ab} + \frac{1}{6}(\nu_{abc} U^c + 3H_{ab}) U^{ab} + O_p(n^{-1/2}). \end{aligned}$$

Infine, si ottiene lo sviluppo della funzione *score* profilo che sarà chiave nella teoria degli stimatori non distorti in mediana. A tal proposito, si partiziona il parametro θ in $\theta = (\tau, \zeta)$, dove τ rappresenta il vettore di parametri di interesse di dimensione $k < p$. Sia r l'indice relativo al vettore τ , e a, b, c, \dots gli indici relativi al parametro di disturbo ζ . Si può ottenere lo sviluppo per la score profilo definita da $U_P(\tau) = [U_P(\tau)_r] = [U_r(\tau, \hat{\zeta}_\tau)]$

$$[U_P(\tau)_r] = U_r + U_{ra}(\hat{\zeta}_\tau - \zeta)^a + \frac{1}{2}U_{rab}(\hat{\zeta}_\tau - \zeta)^{ab} + O_p(n^{-1/2}). \quad (1.10)$$

Sostituendo lo sviluppo di $(\hat{\zeta}_\tau - \zeta)$, ricavabile sulla falsa riga della (1.9) dal modello con τ fissato, in (1.10) e ricordando che $U_{ra} = H_{ra} - i_{ra}$, si ottiene

$$[U_P(\tau)_r] = U_r - i_{ra}k^{ab}U_b + H_{ra}k^{ab}U_b - \frac{1}{2}i_{ra}k^{ab}k^{ce}k^{df}\nu_{bcd}U_eU_f$$

$$-i_{ra}k^{ab}k^{cd}H_{bc}U_d + \frac{1}{2}\nu_{rab}k^{ac}k^{bd}U_cU_d + O_p(n^{-1/2}), \quad (1.11)$$

dove si introduce la notazione k^{ab} per indicare il generico elemento dell'inversa della matrice $[i_{ab}]$ relativa al blocco (ζ, ζ) . In particolare, se ci si ferma al termine dominante di $[U_P(\tau)]_r$ in (1.11), si riconosce la quantità

$$\bar{U} = [\bar{U}_P(\tau)]_r = [U_r - i_{ra}k^{ab}U_b] = [U_r - \gamma_{rb}U_b], \quad (1.12)$$

detta *efficient score*.

1.1.4 Espansioni di Edgeworth e di Cornish Fisher

Sia la variabile casuale (v.c.) $S_n^* = (S_n - n\mu)/\sqrt{n\sigma^2}$ somma standardizzata di n copie i.i.d. di Y , dove $\mathbb{E}[Y] = k_1 = \mu$ e $\mathbb{V}[Y] = k_2 = \sigma^2$. Data la standardizzazione vale che $k_1(S_n^*) = 0$, $k_2(S_n^*) = 1$ e per gli ordini $r = 3, 4, \dots$, sfruttando le relazioni (1.1), (1.2) e (1.3) (Pace & Salvan, 1997, Capitolo 10, §10.6)

$$k_r(S_n^*) = k_r\left(\frac{S_n}{\sqrt{n\sigma^2}}\right) = \frac{k_r(S_n)}{\sqrt{n\sigma^2}} = \frac{nk_r}{n^{r/2}\sigma^r} = n^{1-r/2}\rho_r.$$

Per t fissato, valgono gli sviluppi in serie delle funzioni $K_{S_n^*}(t)$ e $M_{S_n^*}(t)$ al divergere di n

$$K_{S_n^*}(t) = \frac{1}{2}t^2 + \frac{\rho_3}{6\sqrt{n}}t^3 + \frac{\rho_4}{24n}t^4 + O(n^{-3/2}), \quad (1.13)$$

$$\begin{aligned} M_{S_n^*}(t) &= \exp\{K_{S_n^*}(t)\} \\ &= e^{\frac{1}{2}t^2} \left\{ 1 + \frac{\rho_3}{6\sqrt{n}}t^3 + \frac{\rho_4}{24n}t^4 + \frac{\rho_3^2}{72n}t^6 + O(n^{-3/2}) \right\}. \end{aligned} \quad (1.14)$$

L'inversione della (1.14) è possibile grazie alle proprietà dei polinomi di Hermite, composti da una successione di funzioni $H_r(y)$ che soddisfano l'equazione ricorsiva $H_{r+1}(y) = yH_r(y) - H_r'(y)$, dati i valori iniziali $H_0(y) = 1$ e $H_0'(y) = 0$. Ne risulta che i polinomi $H_r(y)$ sono di grado r e sono ortogonali fra loro rispetto al prodotto scalare con funzione peso $\phi(y)$, dove $\phi(y)$ è la densità di una v.c. Normale standard. Un risultato notevole per i polinomi

di Hermite è la seguente identità

$$\int_{-\infty}^{+\infty} e^{ty} H_r(y) \phi(y) dy = t^r e^{\frac{1}{2}t^2}. \quad (1.15)$$

Grazie alla (1.15) si può invertire la (1.14) per ottenere l'espansione di Edgeworth della densità di S_n^*

$$p_{S_n^*}(y) = p_{S_n^*}^E(y) + O(n^{-3/2}), \quad (1.16)$$

dove

$$p_{S_n^*}^E(y) = \phi(y) \left\{ 1 + \frac{1}{6\sqrt{n}} \rho_3 H_3(y) + \frac{1}{24n} \rho_4 H_4(y) + \frac{1}{72n} \rho_3^2 H_6(y) \right\}.$$

Si dimostra che l'errore commesso nell'approssimazione può essere limitato in valore assoluto da una costante indipendente da y . Integrando lo sviuppo per la densità di S_n^* si ottiene il rispettivo sviluppo per la funzione di ripartizione

$$F_{S_n^*}(y) = F_{S_n^*}^E(y) + O(n^{-3/2}), \quad (1.17)$$

dove

$$F_{S_n^*}^E(y) = \Phi(y) - \phi(y) \left\{ \frac{1}{6\sqrt{n}} \rho_3 H_2(y) + \frac{1}{24n} \rho_4 H_3(y) + \frac{1}{72n} \rho_3^2 H_5(y) \right\}.$$

A sua volta, l'inversione di $F_{S_n^*}^E(y)$ in (1.17) rende possibile ottenere l'approssimazione per il quantile di S_n^* in funzione dei quantili di una Normale standard. Sia y_α il quantile α -esimo di S_n^* e sia z_α quello di $Z \sim N(0, 1)$, si può dimostrare che

$$y_\alpha = z_\alpha + \frac{\rho_3}{6\sqrt{n}}(z_\alpha^2 - 1) + \frac{\rho_4}{24n}(z_\alpha^3 - 3z_\alpha) - \frac{\rho_3^2}{36n}(2z_\alpha^3 - 5z_\alpha) + O(n^{-3/2}). \quad (1.18)$$

La relazione (1.18) è detta espansione di Cornish-Fisher e si può usare per approssimare i quantili della distribuzione scelta. In particolare sarà usata in questo lavoro per approssimare la mediana della *score* o della *score* profilo, fissando quindi $\alpha = 0.5$.

1.2 Stimatori non distorti in media

In problemi regolari di stima lo stimatore di massima verosimiglianza ha una distribuzione asintotica centrata sul vero valore del parametro. Tuttavia le proprietà asintotiche possono mal rispecchiare il comportamento per campioni finiti. Il problema di eliminare la distorsione nelle stime di massima verosimiglianza è stato affrontato da più autori in letteratura, col principale obiettivo di accelerare la convergenza a zero della distorsione all'aumentare dell'informazione campionaria. Un primo metodo si basa sullo sviluppo in serie della distorsione $b(\theta) = \mathbb{E}[\hat{\theta}] - \theta$

$$b(\theta) = \frac{b_1(\theta)}{n} + \frac{b_2(\theta)}{n^2} + \dots$$

per cui si può rimuovere il termine $O(n^{-1})$ dallo stimatore tramite la semplice operazione $\hat{\theta}_{\text{BC}} = \hat{\theta} - b_1(\hat{\theta})/n$, ottenendo $\mathbb{E}[\hat{\theta}_{\text{BC}}] = \theta + O(n^{-2})$. In alternativa, si può approssimare il termine $b_1(\theta)/n$ attraverso l'utilizzo di jackknife o bootstrap, assumendo la possibilità computazionale di ripetere più volte la stima del modello. Caratteristica comune a questi due approcci standard è quella di correggere la stima di massima verosimiglianza, con l'implicita necessità che la stima esista per il campione osservato, o per tutti i sottocampioni nel caso del jackknife o bootstrap.

Firth (1993) ha approciato il problema sotto un altro punto di vista, orientando l'attenzione non tanto sull'eliminazione della distorsione quanto sulla sua prevenzione. Ha dimostrato che è sufficiente una modifica della funzione *score* (invece che della stima) per ridurre la distorsione dello stimatore e garantire la finitezza delle stime in alcuni casi in cui $\hat{\theta}$ può essere infinito.

Per introdurre l'idea sottostante si consideri il caso in cui θ sia scalare e l'usuale equazione di stima sia $U(\theta) = \partial l(\theta)/\partial \theta = 0$. La distorsione di $\hat{\theta}$ deriva dalla combinazione della non distorsione della funzione *score* ($\mathbb{E}[U(\theta)] = 0$) e dalla curvatura positiva della stessa ($\partial^2 U(\theta)/\partial \theta^2 \neq 0$). La soluzione è, come mostrato in Figura 1.1, indurre in $U(\theta)$ una distorsione verso il basso, se $b(\theta)$ è positiva di una quantità pari a $i(\theta)b(\theta)$. In tal modo

la nuova equazione di stima risulta

$$U^*(\theta) = U(\theta) - i(\theta)b(\theta) = 0, \quad (1.19)$$

la cui soluzione θ^* ha come proprietà $\mathbb{E}[\theta^*] = \theta + O(n^{-2})$. Nel caso in cui θ sia un vettore allora $U(\theta) = [U_r(\theta)] = [\partial l(\theta)/\partial \theta_r] = 0$, $r = \dots, p$ e $i(\theta)$ è da intendersi come la matrice di informazione di Fisher.

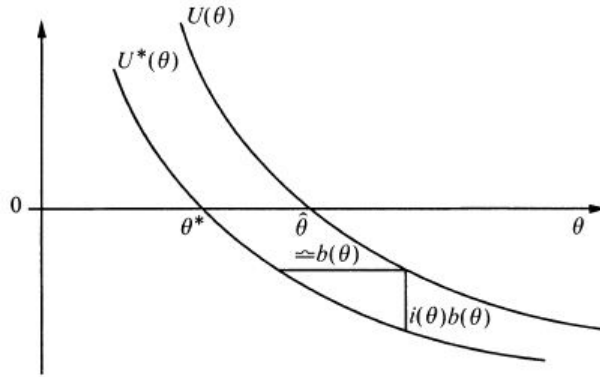


Figura 1.1: Modifica della funzione *score* (Fig. 1, Firth, 1993).

Per formalizzare maggiormente il ragionamento euristico precedente si introduce il problema con la notazione già presentata nel Paragrafo 1.1. Si cerca quel vettore $A(\theta)$, che può dipendere anche dai dati e che sia di ordine $O_p(1)$, tale che la nuova equazione di stima abbia la forma

$$U_r^*(\theta) = U_r(\theta) - A_r(\theta). \quad (1.20)$$

Il termine A_r rimuove il termine di primo ordine della distorsione se soddisfa la relazione

$$\mathbb{E}[A(\theta)] = -i(\theta) \frac{b_1(\theta)}{n} + O(n^{-1}),$$

che porta a due possibili scelte di A a seconda se si scelga di usare l'informazione attesa o osservata

$$A^{(E)} = -i(\theta) \frac{b_1(\theta)}{n} \quad \text{o} \quad A^{(O)} = -j(\theta) \frac{b_1(\theta)}{n}.$$

Poichè la modifica ha un effetto nullo a livello asintotico, in quanto è di ordine

$O_p(1)$, allora la distribuzione asintotica di θ^* non si discosta da quella di $\hat{\theta}$. I due stimatori asintoticamente condividono quindi la matrice di covarianza e gli *standard error* sono calcolabili dall'inversione dell'informazione attesa $i(\theta)$. Nelle famiglie esponenziali con parametrizzazione canonica in θ vale che $A^{(E)}(\theta) = A^{(O)}(\theta)$ il cui r -esimo elemento è

$$A_r = -nk_{r,a} \frac{b_1^a}{n} = -\frac{1}{2} \frac{k_{rab}}{k_{a,b}}, \quad (1.21)$$

dove i pedici r, a, b di k indicano le derivate di $K(\theta)$ nei rispettivi parametri. Usando le relazioni di Bartlett (1.6) si può riscrivere la (1.21) come

$$A_r = \frac{1}{2} \text{tr} \left[i^{-1}(\theta) \left(\frac{\partial i(\theta)}{\partial \theta_r} \right) \right] = \frac{\partial}{\partial \theta_r} \left[\frac{1}{2} \log |i(\theta)| \right].$$

L'equazione di stima $U^*(\theta) = U(\theta) + A(\theta)$ definisce la rispettiva log-verosimiglianza penalizzata $l^*(\theta) = l(\theta) + \frac{1}{2} \log |i(\theta)|$ da cui si ricava la verosimiglianza penalizzata

$$L^*(\theta) = L(\theta) |i(\theta)|^{1/2}, \quad (1.22)$$

dove $L(\theta) = e^{l(\theta)}$. In (1.22) la penalizzazione assume la forma della distribuzione a priori di Jeffreys, dando alla stima non distorta al primo ordine θ^* l'interpretazione di moda a posteriori.

Nel caso della regressione logistica con probabilità di successo $\pi_i = \exp(\eta_i)/(1 + \exp(\eta_i))$, dove $\eta_i = \sum_r x_{ir} \theta_r$, è già dimostrato in letteratura che la distorsione delle stime di massima verosimiglianza porta ad una sovrastima in valore assoluto dei veri valori di θ . C'è bisogno quindi di una penalizzazione con effetto di contrazione (*shrinkage*) dei parametri verso lo zero. Data la matrice d'informazione $i(\theta) = X^T W X$, dove $X = x_{ir}$ e $W = \text{diag}\{\pi_i(1 - \pi_i)\}$, è facile dimostrare che il determinante di $i(\theta)$ è massimizzato per $\pi = 1/2$ a cui corrisponde il valore $\theta = 0$, quindi la a priori di Jeffreys opera la riduzione della distorsione dell'ordine $O(n^{-1})$ tramite l'effetto *shrinkage* auspicato.

Infine, Firth (1993) presenta la modifica nel caso di modelli lineari gene-

ralizzati con link canonico e parametro di dispersione ϕ

$$A_r(\theta) = \frac{1}{2\phi} \sum_{i=1}^n \frac{k_{3i}}{k_{2i}} h_i x_{ir}, \quad (1.23)$$

dove k_{ti} indica il t -esimo cumulante di Y_i e h_i l' i -esimo elemento diagonale della matrice $\text{hat } H = W^{1/2} X (X^\top W X)^{-1} X^\top W^{1/2}$. Mentre nel contesto delle famiglie esponenziali nella parametrizzazione non canonica o per famiglie non esponenziali la modifica A_r assume forma

$$A_r(\theta) = A_r(\theta)^{(E)} = \nu^{a,b} \frac{(\nu_{r,a,b} + \nu_{r,ab})}{2}. \quad (1.24)$$

Il limite principale nel metodo di Firth è il fatto che la riduzione della distorsione è strettamente legata alla parametrizzazione scelta, che risulta cruciale in questo aspetto. Per la regressione logistica, principale applicazione in questo studio, si consiglia la penalizzazione nella parametrizzazione canonica.

1.3 Stimatori non distorti in mediana

Il lavoro di Kenne Pagui *et al.* (2016a) si sviluppa a partire dalla modifica della funzione *score* proposta da Firth (1993), ma considera la mediana invece della media come principale indice di centratura della funzione. Così come in Firth, la nuova modifica non dipende dalle stime di massima verosimiglianza e può quindi fornire stime finite anche quando $\hat{\theta}$ non lo è. Sotto ipotesi di monotonicità, la non distorsione in mediana della funzione *score* implica la non distorsione in mediana al terzo ordine del rispettivo stimatore. Con tale definizione si intende che lo stimatore ha uguale probabilità di sovrastimare o sottostimare il vero valore del parametro, o, in altri termini, questo è la mediana della distribuzione dello stimatore. Per un'introduzione del concetto di stimatori non distorti in mediana, si veda Read (1985).

Si presenta ora il lavoro partendo dal caso semplice in cui θ è scalare, per mostrare in seguito come si possono generalizzare i risultati quando θ è

un vettore di p componenti di cui si cerca la stima indipendentemente dalla presenza o meno di parametri di disturbo.

Tramite l'espansione di Cornish-Fisher (1.18) (si veda anche Pace & Salvan, 1997) si ottiene lo sviluppo asintotico per la mediana sotto θ della funzione *score*

$$M_\theta[l(\theta)] = -\frac{1}{6} \frac{\nu_{\theta,\theta,\theta}}{i(\theta)} + O(n^{-3/2}),$$

dove $\nu_{\theta,\theta,\theta}$ prende il posto di $\nu_{a,b,c}$ nella definizione nel Paragrafo 1.1.2. Pertanto si definisce la nuova funzione *score* modificata

$$\tilde{U}(\theta) = U(\theta) + \frac{1}{6} \frac{\nu_{\theta,\theta,\theta}}{i(\theta)}, \quad (1.25)$$

che risulta non distorta in mediana al terzo ordine. Sia $\tilde{\theta}$ lo stimatore definito dalla soluzione della corrispondente equazione di stima $\tilde{U}(\theta) = 0$. Per costruzione vale che $M_\theta[\tilde{U}(\theta)] = O(n^{-3/2})$ e, per monotonicità di $\tilde{U}(\theta)$, $P_\theta[\tilde{U}(\theta) \leq 0] = 1/2 + O(n^{-3/2})$ implica che $P_\theta[\tilde{\theta} \leq \theta] = 1/2 + O(n^{-3/2})$, quindi lo stimatore $\tilde{\theta}$ è non distorto in mediana al terzo ordine. La quantità $\nu_{\theta,\theta,\theta}/i(\theta)$ è di ordine $O(1)$ quindi lo stimatore $\tilde{\theta}$ si distribuisce asintoticamente come $\tilde{\theta} \sim N(\theta, i^{-1}(\theta))$ e si possono calcolare gli intervalli alla Wald nel modo usuale.

La nuova proposta ha come principale vantaggio l'essere invariante a riparametrazioni, proprietà ereditata dall'invarianza delle statistiche d'ordine. Più formalmente sia $\omega(\theta)$ una riparametrazione $\mathbb{R} \rightarrow \mathbb{R}$ con inversa $\theta(\omega)$ derivabile almeno una volta in ω . Le quantità in (1.25) nella nuova parametrizzazione sono $\nu_{\omega,\omega,\omega}^\Omega(\omega) = \nu_{\theta,\theta,\theta}^\Theta(\theta(\omega))[\theta'(\omega)]^3$ e $i^\Omega(\omega) = i^\Theta(\theta(\omega))[\theta'(\omega)]^2$. Quindi $\tilde{U}(\theta)$ si comporta analogamente a $U(\theta)$ in modo tensoriale e la nuova funzione *score* modificata è $\tilde{U}(\theta(\omega))\theta'(\omega)$. Queste proprietà fanno in modo che la soluzione dell'equazione $\tilde{U}(\omega) = 0$ è ricavabile da $\tilde{\omega} = \omega(\tilde{\theta})$ ed è anch'essa non distorta in mediana al terzo ordine. Al contrario, la modifica (1.24) per la non distorsione in media nel caso scalare ha forma

$$U^*(\theta) = U(\theta) + \frac{\nu_{\theta,\theta,\theta} + \nu_{\theta,\theta\theta}}{2i(\theta)}.$$

Come mostrato in Kosmidis & Firth (2010) $U^*(\theta)$ non si comporta in modo

tensoriale rispetto a riparametrizzazioni, per la presenza al numeratore del momento nullo $\nu_{\theta, \theta\theta}$. Essendo legata alla parametrizzazione scelta, gli autori consigliano di calcolare la modifica necessaria nella parametrizzazione dove la distribuzione degli stimatori è meglio approssimata dalla distribuzione Gaussiana.

Nel caso in cui Y appartenga ad una famiglia esponenziale con parametrizzazione canonica θ , quindi con densità della forma $p_Y(y; \theta) = \exp\{\theta t(y) - K(\theta)\}h(y)$, la funzione *score* modificata è

$$\tilde{U}(\theta) = U(\theta) + \frac{1}{6} \frac{K_{\theta\theta\theta}}{K_{\theta\theta}}, \quad (1.26)$$

dove $K_{\theta\theta\theta} = \partial^3 K(\theta)/\partial\theta^3$ e $K_{\theta\theta} = \partial^2 K(\theta)/\partial\theta^2$. Nel caso discreto, l'espansione di Cornish-Fisher prevede anche dei termini oscillatori (si veda Cai & Wang, 2009), tuttavia questa differenziazione fra caso continuo e discreto non sarà presa in considerazione nel calcolo della modifica. Intuitivamente si può vedere la correzione in (1.26) tramite il terzo cumulante come la correzione della asimmetria della funzione *score* per accelerare la convergenza alla distribuzione Gaussiana e garantire la non distorsione in mediana al terzo ordine. In questo caso particolare, la funzione $\tilde{U}(\theta)$ può essere intesa come derivata della log verosimiglianza penalizzata $\tilde{l}(\theta) = l(\theta) + 1/6 \log K_{\theta\theta}$ a cui corrisponde la verosimiglianza penalizzata

$$L^*(\theta) = L(\theta)|i(\theta)|^{1/6}. \quad (1.27)$$

La (1.27) mostra che la non distorsione in mediana penalizza la verosimiglianza per un fattore pari a $|i(\theta)|^{1/6}$ che è di minore entità rispetto alla priori di Jeffreys in (1.22), in altri termini $\tilde{\theta}$ ha un minor effetto di *shrinkage* rispetto a θ^* . Questo implica che $\tilde{\theta}$ assume valore intermedio fra la stima θ^* non distorta in media e la stima $\hat{\theta}$ di massima verosimiglianza.

Il primo problema nella trasposizione della modifica al caso in cui θ sia un vettore in \mathbb{R}^p è legato alla definizione esatta di mediana in più dimensioni. Sebbene vi siano più definizioni possibili (si veda Oja, 2013) nessuna si presta alla generalizzazione della formula (1.25), o per poca praticabilità o

per scarse proprietà campionarie dimostrate in studi simulativi. L'approccio scelto per la stima di $\tilde{\theta}$ è quello di impostare un sistema di p equazioni, una per ogni elemento θ_r , $r = 1, \dots, p$. In ogni equazione si considera l' r -esimo elemento come unico parametro di interesse ed i rimanenti $p - 1$ come parametri di disturbo, quindi il sistema consta di p equazioni di funzioni *score* profilo da risolvere simultaneamente, senza utilizzare però stime vincolate dei parametri di disturbo. Viene quindi utilizzata l'*efficient score* (1.12) come approssimazione della funzione *score* profilo tramite il suo termine dominante. Con questi accorgimenti il vettore della funzione *score* modificato è

$$\begin{aligned}\tilde{U}_r(\theta) &= U_r(\theta) - \gamma_{ra} U_a(\theta) - k_{1r} + \frac{1}{6} \frac{k_{3r}}{k_{2r}}, \quad r = 1, \dots, p \\ &= \bar{U}_r(\theta) - k_{1r} + \frac{1}{6} \frac{k_{3r}}{k_{2r}},\end{aligned}\tag{1.28}$$

dove i termini k_{jr} , $j = 1, 2, 3$ sono i cumulanti di ordine j riportati più avanti nella formula (1.30). L'inserimento di k_{1r} è necessario per la centratura della funzione all'interno dello sviluppo di Edgeworth. Infine la stima $\tilde{\theta}$ è ottenuta dall'equazione $\tilde{U}(\theta) = 0$.

Sia $\bar{U}(\theta)$ l'*efficient score* già introdotta in (1.12), si dimostra che $\bar{U}(\theta) = 0$ e $U(\theta) = 0$ hanno la stessa soluzione, ovvero la stima di massima verosiglianza $\hat{\theta}$. Il principale vantaggio di $\bar{U}(\theta)$ è che la matrice Hessiana attesa associata $H(\theta)$ è diagonale, a differenza di $i(\theta)$, infatti vale che $H(\theta) = \mathbb{E}_\theta [-\partial \bar{U}(\theta) / \partial \theta^\top] = \{diag(i^{-1}(\theta))\}^{-1}$. Applicare la modifica su $\bar{U}(\theta)$ invece che su $U(\theta)$ rende quindi equivalente risolvere il sistema delle equazioni di stima congiuntamente o componente per componente. I due vettori sono legati dalla trasformazione lineare

$$\bar{U}(\theta) = \bar{A}(\theta)U(\theta),\tag{1.29}$$

dove $\bar{A}(\theta)$ è una matrice non singolare di ordine p . Dalla derivazione di (1.29) vale $H(\theta) = \bar{A}(\theta)i(\theta)$ da cui per inversione si ha che $\bar{A}(\theta) = H(\theta)i^{-1}(\theta)$. Infine poichè la modifica rimane di ordine $O(1)$ allora $\tilde{\theta}$ ha distribuzione asintotica uguale a $\hat{\theta}$. La stima congiunta di $\tilde{\theta}$ a partire dalla equazione $\tilde{U}(\theta) = 0$ porta ad un risultato che ben approssima la stima di ogni elemento di $\tilde{\theta}_p$ ottenuto

come soluzione di p equazioni di stima $\tilde{U}_P(\theta_r) = 0$, dove

$$\tilde{U}_P(\theta_r) = U_P(\theta_r) - k_{1\theta_r} + \frac{1}{6} \frac{k_{3\theta_r}}{k_{2\theta_r}},$$

$U_P(\theta_r) = \partial l(\theta_r, \hat{\theta}_{(-r)}) / \partial \theta_r$ e $\hat{\theta}_{(-r)}$ è la stima di massima verosimiglianza di $\theta_s, s = 1, \dots, r-1, r+1, \dots, p$ per un dato valore di θ_r . In particolare, vale la relazione dimostrata in Kenne Pagui *et al.* (2016a)

$$\tilde{\theta}_r - \tilde{\theta}_{rP} = O_p(n^{-3/2}),$$

che ancora una volta sfrutta la diagonalità di $H(\theta)$ per la quale $\mathbb{E}_\theta \left[\partial \tilde{U}_r / \partial \theta_s \right] = O(1)$, per $r, s = 1, \dots, p, r \neq s$. Ne risulta che la non distorsione in mediana per ogni componente di $\tilde{\theta}_r$ al terzo ordine è implicata dalla proprietà analoga di $\tilde{\theta}_{rP}$.

Si dimostra brevemente il calcolo dei cumulanti presenti in (1.28). In particolare, per k_{1r} si veda Pace & Salvan, (1997) in cui si fa uso delle identità di Bartlett in (1.6), mentre k_2 e k_3 sono i termini dominanti degli sviluppi dei cumulanti per la *score* profilo, quindi sono i rispettivi cumulanti relativi all'*efficient score*, si veda Mc Cullagh & Tibshirani, (1990)

$$\mathbb{E} [U_P(\theta_r)] = k_{1r} + O(n^{-1}),$$

dove

$$\begin{aligned} k_{1r} &= k^{ab}(\nu_{r,ab} - i_{rc}k^{cd}\nu_{b,ad}) + \frac{1}{2}k^{ab}(\nu_{rab} - i_{rc}k^{cd}\nu_{abd}) \\ &= -\frac{1}{2}k^{ab}\{(\nu_{r,ab} + \nu_{r,a,b}) - i_{rc}k^{cd}(\nu_{d,ab} + \nu_{d,a,b})\} \\ &= -\frac{1}{2}\bar{A}_{rc}\nu^{ab}(\nu_{c,a,b} + \nu_{c,ab}), \end{aligned}$$

e per gli altri cumulanti valgono le relazioni

$$\begin{aligned} k_{2r} &= [\mathbb{V}\{\bar{U}\}]_{rr} = [\mathbb{V}\{A_{rs}U_s\}]_{rr} \\ &= [A i A^\top]_{rr} = [H i^{-1} i i^{-1} H]_{rr} \\ &= (i^{rr})^{-1} i^{rr} (i^{rr})^{-1} = 1/i^{rr}, \\ k_{3r} &= \mathbb{E} [\bar{U}_r \bar{U}_r \bar{U}_r] \\ &= \mathbb{E} [\bar{A}_{ra} U_a \bar{A}_{rb} U_b \bar{A}_{rc} U_c] \end{aligned} \tag{1.30}$$

$$= \bar{A}_{ra} \bar{A}_{rb} \bar{A}_{rc} \nu_{abc}.$$

Nella formula (1.30) la notazione indiciale è implicita per gli indici a, b, c ma non per r , quindi per ogni componente θ_r il calcolo di k_{1r} e k_{3r} consta di tre sommatorie per gli indici liberi.

Per concludere si richiama che, per ogni $r = 1, \dots, p$, \bar{U}_r si comporta in modo tensoriale rispetto a riparametrizzazioni che mantengono l'interesse, quindi $\tilde{\theta}$ è equivariante rispetto a riparametrizzazioni che operano sugli elementi di θ in modo separato. Seppur ridotta, la proprietà permane per un'ampia classe di trasformazioni, ad esempio $\omega = \exp\{\theta\}$.

Capitolo 2

Il modello di Rasch

"Sed quis custodiet ipsos custodes?"

Giovenale

2.1 Introduzione al modello di Rasch

Il modello di Rasch è stato introdotto dall'omonimo statistico danese nel 1960 nell'ambito dei test per l'abilità di lettura ad alta voce, un indicatore usato per misurare le tre componenti di precisione, ritmo e fluidità strettamente correlate con la comprensione del testo (Fuchs *et al.*, 2001). L'obiettivo iniziale fu quello di costruire un modello per la probabilità di un bambino s di compiere un errore nel test i contenente N_i parole. Rasch propose un modello binomiale con parametro $\theta_{si} = \delta_i/\gamma_s$, ed approssimò la variabile $Y_{si} \sim \text{Binomial}(\theta_{si}, N_i)$ con una $\text{Poisson}(\lambda_{si})$ fissato N_i grande a sufficienza (Fienberg, 2006). Il numero di errori atteso è quindi

$$\lambda_{si} = \frac{N_i \delta_i}{\gamma_s} = \frac{t_i}{\gamma_s},$$

dove δ_i rappresenta la "difficoltà del test" e γ_s la "abilità del bambino". Cressie e Holland (1983) hanno sottolineato che l'abilità del soggetto non è direttamente osservabile e si può solo stimare una "probabilità manifesta" a cui corrisponde una proporzione di bambini di una data popolazione che ottiene un punteggio analogo nel test. Il modello è utilizzato col principale scopo di

valutare la difficoltà dei test stessi ed i primi lavori sul modello si focalizzano sulla stima dei parametri δ_i utili per la costruzione della curva caratteristica degli item (*ICC*) (un esempio è in Figura 2.1). Nella forma attuale il modello è specificato tramite la distribuzione binomiale, forma adottata dallo stesso Rasch nei suoi studi successivi. I dati si presentano generalmente sotto forma di una matrice a due entrate dove per riga sono riportati i soggetti, per colonna gli item, o test, e le celle assumono il valore di $y_{si} = 1$ se l' s -esimo soggetto ha risposto correttamente all' i -esimo quesito e $y_{si} = 0$ nel caso contrario, con $s = 1, \dots, S$ e $i = 1, \dots, I$. Questa struttura evidenzia la forte simmetria del modello fra il ruolo degli item e quello dei soggetti, proprietà che ha forti riscontri nei metodi di stima. Negli anni successivi alla pubblicazione, il modello di Rasch si è imposto come riferimento nella psicometria: viene infatti utilizzato in vari test sull'intelligenza ed abilità per l'analisi degli item (si veda ad esempio Lord, 1968). Vi sono anche studi meno convenzionali; ad esempio Wright (1980) ha mostrato esempi di applicazione nel settore bancario e nel disegno degli esperimenti; Perline & Wainer (1980) hanno sviluppato un modello di Rasch per le dichiarazioni offensive di criminali per prevedere la recidiva trattandola come un tratto latente.

2.2 Modello di Rasch ad uno o due parametri

Nella sua specifica base il modello logistico ad un parametro ipotizza che ogni cella della matrice dei dati sia rappresentabile come $Y_{si} \sim \text{Bernoulli}(\pi_{si})$, dove, utilizzando il link canonico, vale che

$$\text{logit}(\pi_{si}) = \gamma_s + \alpha_i, \quad s = 1, \dots, S, \quad i = 1, \dots, I.$$

Per garantire l'identificabilità del modello si pone come vincolo $\alpha_1 = 0$, alternativamente si può imporre un vincolo simile sui γ_s , tuttavia lo stesso Rasch propose di inserire il vincolo fra i parametri relativi agli item per permettere la confrontabilità delle stime su campioni di soggetti diversi che affrontano lo stesso test. In questo caso, γ_s mantiene la sua interpretazione di "abilità del candidato", mentre α_i assume il ruolo di "facilità relativa del test" in quanto all'aumentare di α_i aumenta π_{si} (rispetto a prima vale che

$\alpha_i = -\delta_i$), il valore è confrontabile solo rispetto al primo item preso come riferimento. La Figura 2.1 riporta un esempio della curva caratteristica degli item al variare del parametro α_i . Si nota come i diversi valori di α_i comportino una semplice traslazione della curva col risultato intuitivo che, a parità di γ_s , maggiore è la facilità del test maggiore è la probabilità che il soggetto risponda giusto.

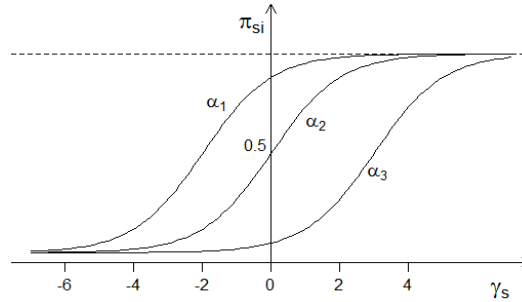


Figura 2.1: Esempio di ICC per il modello logistico ad un parametro con $\alpha_1 > \alpha_2 = 0 > \alpha_3$

Il modello ha verosimiglianza per $\gamma = (\gamma_1, \dots, \gamma_S)$ e $\alpha = (\alpha_1, \dots, \alpha_I)$

$$\begin{aligned}
 L(\gamma, \alpha) &= \prod_{s=1}^S \prod_{i=1}^I \pi(\gamma_s, \alpha_i)^{y_{si}} (1 - \pi(\gamma_s, \alpha_i))^{(1-y_{si})} \\
 &= \exp \left(\sum_{s=1}^S \sum_{i=1}^I \gamma_s y_{si} + \sum_{s=1}^S \sum_{i=1}^I \alpha_i y_{si} \right) / \prod_{s=1}^S \prod_{i=1}^I (1 + e^{\gamma_s + \alpha_i}) \\
 &= \exp \left(\sum_{s=1}^S \gamma_s y_{s\cdot} + \sum_{i=1}^I \alpha_i y_{\cdot i} \right) / \prod_{s=1}^S \prod_{i=1}^I (1 + e^{\gamma_s + \alpha_i}). \quad (2.1)
 \end{aligned}$$

dove $y_{s\cdot}$ e $y_{\cdot i}$ sono rispettivamente le somme di riga e di colonna della matrice dei dati y . Poichè il modello è una famiglia esponenziale $y_{s\cdot}$ e $y_{\cdot i}$ sono anche le statistiche sufficienti minimali per i parametri γ_s e α_i .

Il modello logistico a due parametri introduce un nuovo insieme di parametri, $\beta = (\beta_1, \dots, \beta_I)$, relativi agli item. Mantenendo le stesse ipotesi distributive la nuova specifica è

$$\text{logit}(\pi_{si}) = \beta_i \gamma_s + \alpha_i, \quad s = 1, \dots, S, \quad i = 1, \dots, I.$$

Si può a volte trovare sotto la riparametrizzazione $\text{logit}(\pi_{si}) = \beta_i(\gamma_s + \alpha_i)$ che non cambia nella sostanza l'interpretazione dei parametri. Questa versione fu proposta in modo indipendente anche da Birnbaum (1957) come caso particolare di un *latent trait model*. Si può vedere il modello ad un parametro logistico come un caso particolare in cui tutti i β_i sono uguali fra loro. I nuovi parametri β_i sono una misura di potere discriminante dell'item ovvero della capacità dell'item di separare soggetti capaci e non. All'aumentare del valore di β_i la ICC tende ad essere più ripida (si veda la Figura 2.2), infatti la sua derivata è $\partial\pi_{si}/\partial\gamma_s = \beta_i\pi_{si}/(1 - \pi_{si})$, pertanto per valori alti di β_i basta una piccola variazione di γ_s per far variare drasticamente la probabilità di rispondere correttamente all'item. L'introduzione del nuovo parametro è tale

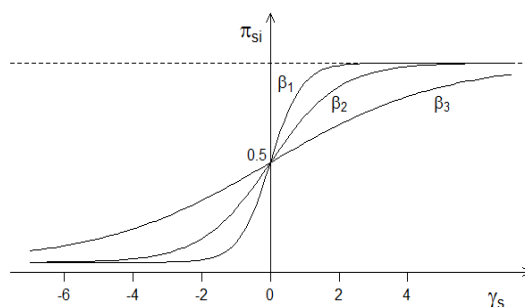


Figura 2.2: Esempio di ICC per un modello logistico a due parametri con $\beta_1 > \beta_2 = 1 > \beta_3$ e $\alpha_1 = \alpha_2 = \alpha_3 = 0$

da rendere il modello una famiglia esponenziale curva per cui non si può riscrivere la verosimiglianza come nel caso precedente

$$\begin{aligned}
 L(\gamma, \alpha, \beta) &= \prod_{s=1}^S \prod_{i=1}^I \pi(\gamma_s, \alpha_i, \beta_i)^{y_{si}} (1 - \pi(\gamma_s, \alpha_i, \beta_i))^{(1-y_{si})} \\
 &= \exp \left(\sum_{s=1}^S \sum_{i=1}^I \beta_i \gamma_s y_{si} + \sum_{i=1}^I \alpha_i y_{\cdot i} \right) / \prod_{s=1}^S \prod_{i=1}^I (1 + e^{\beta_i \gamma_s + \alpha_i}) .
 \end{aligned} \tag{2.2}$$

Il modello è identificabile se si pongono i due vincoli $\alpha_1 = 0$ e $\beta_1 = 1$.

2.3 Metodi di stima alternativi e problemi associati

La stima del modello di Rasch è un tema affrontato da diversi autori in letteratura; si veda Wright & Master (1982) per una rassegna più dettagliata dei metodi classici o Linacre (2004) per quelli più recenti. Tutti i metodi qui presentati si riferiscono alla stima del modello logistico ad un parametro.

La stima di massima verosimiglianza dell'intero vettore $\theta = (\gamma, \alpha)$ presenta problemi dovuti principalmente all'alto numero di parametri da stimare congiuntamente e al diverso ordine asintotico con cui aumenta l'informazione relativa agli stessi. Inoltre, è frequente che alcune osservazioni siano sulla frontiera dello spazio campionario, con la conseguenza che non esistono le stime di massima verosimiglianza (si veda Andersen 1980, Teorema 6.3).

Una prima soluzione è quella di considerare parametri di interesse solo quelli relativi agli item e parametri di disturbo i rimanenti. Tuttavia la stima di massima verosimiglianza degli α_i va incontro a quello che in letteratura è noto come il problema di Neyman-Scott (1948). Dato che al crescere del numero S di candidati nel campione cresce in ugual misura anche il numero di parametri γ_s da stimare, questi vengono chiamati parametri incidentali, mentre gli α_i , il cui numero è costante, sono detti parametri strutturali. Il problema di stima quindi non è riconducibile alle ipotesi di regolarità che caratterizzano le usuali proprietà asintotiche degli stimatori di massima verosimiglianza. Neyman e Scott (1948) dimostrarono che, nel caso di presenza di parametri incidentali gli stimatori per i parametri strutturali sono non consistenti. Nel caso di Rasch, Andersen (1980, Teorema 6.1) dimostrò che

Teorema 2.3.1.

Per valori moderati di I , gli stimatori di massima verosimiglianza per $\alpha_1, \dots, \alpha_I$ sono non consistenti e hanno una distorsione asintotica approssimata pari a $I/(I - 1)$.

Sebbene la distorsione tenda a perde di importanza al crescere di I si possono comunque definire dei nuovi stimatori non distorti per ogni I come segue

$$\alpha_i^* = \frac{I - 1}{I} \alpha_i. \quad (2.3)$$

La consistenza degli stimatori in (2.3) è stata dimostrata da Haberman (1977) per $S \rightarrow \infty$ e I grande a sufficienza (si nota che in pratica vale molto spesso che $S \gg I$). Una soluzione alternativa per la stima dei α_i per valori di I moderati fu proposta dallo stesso Rasch (1960) ed è nota come *conditional ML-estimation*.

Si presenta il metodo di stima tramite un semplice esempio (preso da Pace & Salvan, 1997, Esempio 4.12). Sia un modello di Rasch con S individui e $I = 2$ item, le osservazioni y_{si} sono delle Bernoulli indipendenti con probabilità di successo

$$\pi_{s1} = \frac{e^{\gamma_1}}{1 + e^{\gamma_1}}, \quad \pi_{s2} = \frac{e^{\gamma_2 + \alpha_2}}{1 + e^{\gamma_2 + \alpha_2}}. \quad (2.4)$$

In (2.4) si è posto il vincolo $\alpha_1 = 0$ quindi α_2 rappresenta la differenza relativa fra la difficoltà del secondo item rispetto al primo in scala logit, infatti $\alpha_2 = \text{logit}(\pi_{s1}) - \text{logit}(\pi_{s2})$. La verosimiglianza del modello segue da (2.1). La statistica $u = (u_1, \dots, u_S)$, dove $u_s = (y_{s1} + y_{s2})$, è sufficiente minimale per il vettore $\gamma = (\gamma_1, \dots, \gamma_S)$ per ogni valore di α_2 . Condizionandosi a (u_1, \dots, u_S) le y_{s2} sono estrazioni indipendenti da una Bernoulli con probabilità $\pi_{s2}^* = \pi_{s2}^*(u_s)$ che assume tre possibili valori:

$$\begin{aligned} \pi_{s2}^*(u_s = 0) &= \mathbb{P}[Y_{s2} = 1 \mid Y_{s1} + Y_{s2} = 0] = 0, \\ \pi_{s2}^*(u_s = 1) &= \mathbb{P}[Y_{s2} = 1 \mid Y_{s1} + Y_{s2} = 1] \\ &= \frac{\mathbb{P}[Y_{s1} = 0, Y_{s2} = 1]}{\mathbb{P}[Y_{s1} = 1]\mathbb{P}[Y_{s2} = 0] + \mathbb{P}[Y_{s1} = 0]\mathbb{P}[Y_{s2} = 1]} \\ &= \frac{(1 - \pi_{s1})\pi_{s2}}{\pi_{s1}(1 - \pi_{s2}) + (1 - \pi_{s1})\pi_{s2}} \\ &= \frac{1}{1 + e^{\alpha_2}}, \\ \pi_{s2}^*(u_s = 2) &= \mathbb{P}[Y_{s2} = 1 \mid Y_{s1} + Y_{s2} = 2] = 1. \end{aligned}$$

Se si supponga, senza perdita di generalità, che le prime n_1 unità siano tali che $u_s = 1$, la verosimiglianza condizionata è data da

$$L_c(\alpha_2) = \exp\left(\alpha_2 \sum_{s=1}^{n_1} y_{s2}\right) / [1 + e^{\alpha_2}]^{n_1}.$$

Si può generalizzare il risultato a $I > 2$ item seguendo l'idea appena proposta. Considerando che le statistiche assumo valori $u_s = r$, $r = 0, \dots, I$ e che n_r sia il numero di u_s tali che $u_s = r$, la verosimiglianza condizionata in questo caso assume la forma di (Andersen, 1980, Formula 6.20).

$$L_c(\alpha) = \exp\left(\sum_{i=2}^I \alpha_i \sum_{s=1}^{n_{i-1}} y_{si}\right) / \prod_{r=0}^I \left[\exp\left(\sum_{i=1}^I \alpha_i \sum_{s=1}^{n_r} y_{si}\right) \right]^{n_r}.$$

Le relative proprietà asintotiche di questi stimatori sono state dimostrate da Andersen nel teorema seguente (si veda Andersen, 1980, Teorema 6.2 o Andersen, 1970)

Teorema 2.3.2.

Se non ci sono due α_i uguali e se le equazioni di verosimiglianza hanno stime finite, allora gli stimatori CML $\alpha_2, \dots, \alpha_I$ sono consistenti e asintoticamente normali.

Questo risultato, seppur notevole, è limitato dall'ipotesi iniziale secondo cui le stime devono esistere finite, ovvero che non esista un item per il quale tutti i soggetti hanno risposto in modo corretto o incorretto.

Data la simmetria del modello è possibile applicare la stessa tecnica di stima anche per i soggetti, condizionandosi questa volta ai totali di colonna. Gli stimatori $\hat{\gamma}_s$ ottenuti dalla verosimiglianza condizionata godono delle stesse proprietà degli analoghi stimatori per gli item. Tuttavia presentano diversi problemi, primo fra tutti è che la stima degli item prima e dei soggetti poi sullo stesso insieme di dati non è tecnicamente corretto (Rasch, 1980). Poiché si lavora in due fasi non è possibile la stima della informazione incrociata dei due blocchi, inoltre la doppia stima dei parametri in modo condizionato incorre negli stessi problemi delle stime di massima verosimiglianza che necessitano che entrambe le dimensioni dei dati, S e I , tendano a $+\infty$, e, per ultimo, rimane non risolto il problema di finitezza delle stime già accennato.

Sebbene Rasch proponesse il modello con “i parametri specifici per gli individui” (1980, pagina 9), quelli che oggi sono chiamati effetti fissi, un approccio al problema proposto negli anni successivi fu quello di considerare

i γ_s come effetti casuali, ancora una volta in riferimento al modello con un parametro logistico. Il metodo di stima è incentrato sull'ipotesi che i soggetti si possano considerare come parte di una popolazione più ampia e per tanto i relativi parametri siano da trattare come estrazioni da una distribuzione parametrica $\phi(\gamma)$. Sebbene questo approccio sia predominante nella *Item Response Theory* (IRT), De Boeck (2008) ha mostrato che considerare anche i parametri degli item come effetti casuali è teoricamente corretto e coerente, in quanto si può pensare alle domande di un test come un campione estratto dalla popolazione di tutte le possibili domande sulla materia. Il principale vantaggio di questo approccio risiede nel diminuire il numero di parametri totali da stimare, riducendo gli S parametri γ_s a quelli caratterizzanti la popolazione dei soggetti. Una spiegazione approfondita dei modelli ad effetti casuali esula dallo scopo di questo lavoro e quindi si presenta solamente la specifica del modello. Si supponga per il momento che gli α_i siano noti o stimati con efficienza e si voglia solo stimare la distribuzione della popolazione dei soggetti $\phi(\gamma)$, sia u_s la statistica $u_s = \sum_i y_{si}$. Allora la distribuzione di u_s per un individuo estratto a caso da $\phi(\gamma)$ è (Andersen, 1980)

$$\begin{aligned} f(u_s) &= \int f(u_s|\gamma)\phi(\gamma)d\gamma \\ &= \exp\left(\sum_{i=1}^I \alpha_i \sum_{s:u_s=\sum_i y_{si}} y_{si}\right) \int \{e^{\gamma u_s} / \prod_i e^{\gamma u_s + \alpha_i}\} \phi(\gamma) d\gamma, \end{aligned}$$

da cui si può derivare la verosimiglianza

$$L(\theta) = \prod_{r=0}^I f(r)^{n_r},$$

dove la scelta di $\phi(\gamma)$ è spesso la distribuzione Gaussiana. Infine, poichè i valori di α_i sono ignoti si può procedere alla stima congiunta di tutto il vettore dei parametri specificando un modello logistico ad effetti casuali $\text{logit}(\pi_{si}) = \gamma_s + \alpha_i = \mu + R_s + \alpha_i$, con $\gamma_s \sim N(\mu, \sigma_S^2)$ e $R_s \sim N(0, \sigma_S^2)$. La stima del modello è ottenibile attraverso la massima verosimiglianza ristretta (REML).

La proposta di questa tesi consiste nel presentare un metodo di stima del modello di Rasch basato sul lavoro di Kenne Pagui *et al.* (2016). Il metodo è in grado di stimare congiuntamente tutto il vettore di parametri sia nel modello

logistico ad uno che a due parametri, mantenendo i parametri come effetti fissi, in accordo allo spirito originario con cui il modello è stato proposto. Inoltre, si vedrà che il metodo non soffre del problema della possibile non esistenza delle stime.

Capitolo 3

Stima non distorta in mediana del modello di Rasch

"Who refuses to do arithmetic is doomed to talk nonsense"

J. McCarthy

3.1 Modello di Rasch ad un parametro

3.1.1 Matrice dei dati

Nella formulazione più semplice del modello di Rasch si introduce un solo insieme di parametri per i soggetti e gli item, rispettivamente $\gamma = (\gamma_1, \dots, \gamma_S)$ e $\alpha = (\alpha_1, \dots, \alpha_I)$. Tutte le n combinazioni possibili sono rappresentate nella matrice del disegno X di dimensioni $n \times p$, con $n = S \times I$ e $p = S + I - 1$, dove è già stato imposto il vincolo di identificabilità dei parametri ($\alpha_1 = 0$). La classica forma della matrice è quella in Figura 3.1. La sparsità della matrice è un elemento chiave nello sviluppo di tutto il seguente capitolo e merita attenzione. È dovuta al fatto che il parametro relativo ad un soggetto si “attiva” solo un numero di volte pari alle domande cui il soggetto risponde, mentre un parametro relativo ad un item compare una sola volta per ogni soggetto.

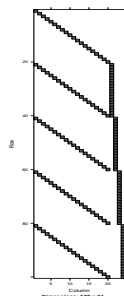


Figura 3.1: Struttura della matrice dei dati per il modello di Rasch ad un parametro.

La variabile risposta è un vettore di dimensione $n \times 1$, ottenuta vettorizzando le entrate della matrice originaria $S \times I$, ha quindi struttura

$$y_{si} = \underbrace{(y_{11}, \dots, y_{S1})}_{item1}, \underbrace{(y_{12}, \dots, y_{S2})}_{item2}, \dots, \underbrace{(y_{1I}, \dots, y_{SI})}_{itemI}.$$

3.1.2 Verosimiglianza e momenti nulli

Si ripresentano brevemente le formule di verosimiglianza, presentata nella (2.1), e le rispettive quantità necessarie al calcolo della modificazione (1.28). Siano $h = 1, \dots, n$ l'indice che scorre le righe di X , $i, j = 1, \dots, I$ gli indici relativi agli item, $s, t = 1, \dots, S$ quelli relativi ai soggetti e $r, a, b, c = 1, \dots, p$ gli indici del vettore dei parametri $\theta = (\gamma, \alpha)$, in particolare si ha una corrispondenza biunivoca fra h e le coppie (s, i) . Sia $\eta_h = x_h \theta$ il predittore lineare e $F(\cdot)$ la funzione *link* inverso del modello logistico tale che $\pi_h = F(\eta_h)$. Si ottiene quindi

$$L(\theta) = \prod_{h=1}^n \pi_h^{y_h} (1 - \pi_h)^{1 - y_h},$$

$$l(\theta) = \sum_{h=1}^n y_h \eta_h - \sum_{h=1}^n \log(1 + e^{\eta_h}),$$

$$U_\alpha(\theta) = \sum_{h=1}^n X_{ha} (y_h - F(\eta_h)),$$

$$U_{ab}(\theta) = - \sum_{h=1}^n X_{ha} X_{hb} F'(\eta_h),$$

$$U_{abc}(\theta) = - \sum_{h=1}^n X_{ha} X_{hb} X_{hc} F'(\eta_h) (1 - 2F(\eta_h)).$$

Con questa specifica diventa immediato il calcolo dei momenti nulli

$$\begin{aligned} \nu_{a,bc} &= 0, \\ \nu_{a,b,c} &= -\nu_{abc} = -\mathbb{E}[\partial^3 l(\theta) / \partial \theta_a \partial \theta_b \partial \theta_c] = -\mathbb{E}[U_{abc}(\theta)] \\ &= \sum_h X_{ha} X_{hb} X_{hc} F'(\eta_h) (1 - 2F(\eta_h)), \end{aligned} \quad (3.1)$$

dove la (3.1) deriva da

$$\begin{aligned} U_{abc}(\theta) &= \frac{\partial^3}{\partial \theta_a \partial \theta_b \partial \theta_c} l(\theta) \\ &= \frac{\partial^3}{\partial \theta_a \partial \theta_b \partial \theta_c} (\theta_1 \sum_h X_{h1} y_h + \dots + \theta_p \sum_h X_{hp} y_h - \sum_h \log(1 + e^{\eta_h})) \\ &= \frac{\partial^2}{\partial \theta_b \partial \theta_c} (\sum_h X_{ha} y_h - \sum_h X_{ha} \frac{e^{\eta_h}}{1 + e^{\eta_h}}) \\ &= \frac{\partial}{\partial \theta_c} \left(- \sum_h X_{ha} X_{hb} \frac{e^{\eta_h}}{(1 + e^{\eta_h})^2} \right) \\ &= - \sum_h X_{ha} X_{hb} X_{hc} (1 - e^{\eta_h}) \frac{e^{\eta_h}}{(1 + e^{\eta_h})^3} \\ &= - \sum_h X_{ha} X_{hb} X_{hc} F'(\eta_h) (1 - 2F(\eta_h)). \end{aligned}$$

La matrice delle derivate seconde è notoriamente non stocastica nel modello logistico ed è divisibile in quattro blocchi in funzione della separazione delle due componenti $\theta = (\gamma, \alpha)$.

Dalla Figura 3.4 emerge come l'ipotesi di indipendenza fra soggetti ed l'indipendenza fra gli item porti ad avere due blocchi diagonali sulla diagonale. Gli unici blocchi non sparsi sono $U_{\gamma\alpha}$ (e relativo trasposto) poiché un soggetto s risponde a tutte le domande e a sua volta ogni domanda è

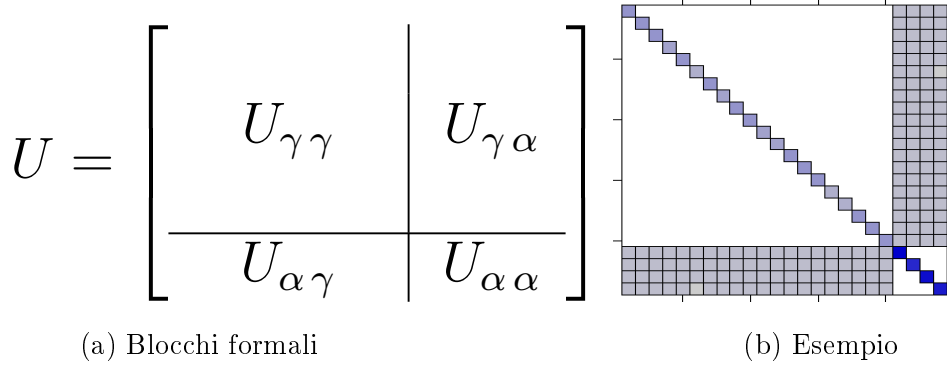


Figura 3.4: Divisione a blocchi della matrice delle derivate seconde U .

sottoposta a tutti i partecipanti. Riassumendo, si ha

$$\begin{aligned} i_{\gamma_s, \gamma_t} &= 0, & s, t &= 1, \dots, S, & s \neq t, \\ i_{\alpha_i, \alpha_j} &= 0, & i, j &= 2, \dots, I, & i \neq j, \\ i_{\gamma_s, \alpha_i} &\neq 0, & \forall s, i. \end{aligned}$$

3.1.3 Calcolo dei cumulanti k_1 , k_2 e k_3

Il cuore del problema computazionale nel calcolo della modificazione (1.28) è l'insieme dei tre cicli annidati nella (1.30), necessari per ogni elemento del vettore di parametri. Data la particolare natura del modello di Rasch si riesce a sfruttare in modo opportuno la sparsità delle matrici coinvolte per guadagnare in tempi computazionali. In particolare, si ricorda che fissato il primo indice il tensore $\nu_{a,b,c}$ si riduce ad una matrice con caratteristiche ben precise a seconda se l'indice a si riferisca ad un parametro relativo a soggetti, γ_s , o ad un item, α_i . In dettaglio:

1. Se a si riferisce a γ_s

allora le matrici di riferimento sono della stessa forma della Figura 3.5 e gli unici elementi non nulli della matrice $\nu_{a,b,c}$ sono

$$\nu_{\gamma_s, \gamma_s, \gamma_s}, \quad 2\nu_{\gamma_s, \gamma_s, \alpha_i}, \quad \nu_{\gamma_s, \alpha_i, \alpha_i}, \quad i = 2, \dots, I.$$

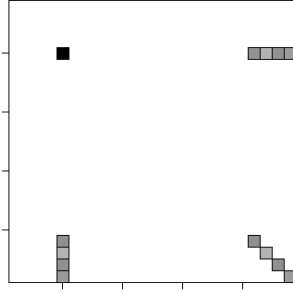


Figura 3.5: Particolare del tensore $\nu_{a,b,c}$ con a corrispondente a γ_5 .

2. Se a si riferisce ad α_i

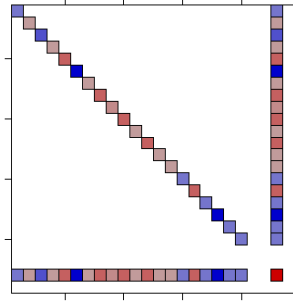


Figura 3.6: Particolare del tensore $\nu_{a,b,c}$ con a corrispondente a α_4 .

allora le matrici di riferimento sono della stessa forma della Figura 3.6 e gli unici elementi non nulli della matrice $\nu_{\alpha,b,c}$ sono

$$\nu_{\alpha_i, \gamma_s, \gamma_s}, \quad 2\nu_{\alpha_i, \gamma_s, \alpha_i}, \quad \nu_{\alpha_i, \alpha_i, \alpha_i}, \quad s = 1, \dots, S.$$

Se si esplicitano i conti dalla relazione di $\nu_{a,b,c}$, fissando s e i , e ricordando che $F'(\eta_h)(1 - 2F(\eta_h)) = \pi_{si}(1 - \pi_{si})(1 - 2\pi_{si})$, si mostra che

$$\begin{aligned} \nu_{\gamma_s, \gamma_s, \gamma_s} &= -\nu_{\gamma_s \gamma_s \gamma_s} = \sum_{i=1}^I \pi_{si}(1 - \pi_{si})(1 - 2\pi_{si}), \\ \nu_{\gamma_s, \gamma_s, \alpha_i} &= -\nu_{\gamma_s \gamma_s \alpha_i} = \pi_{si}(1 - \pi_{si})(1 - 2\pi_{si}), \\ \nu_{\gamma_s, \alpha_i, \alpha_i} &= -\nu_{\gamma_s \alpha_i \alpha_i} = \pi_{si}(1 - \pi_{si})(1 - 2\pi_{si}), \\ \nu_{\alpha_i, \gamma_s, \gamma_s} &= -\nu_{\alpha_i \gamma_s \gamma_s} = \pi_{si}(1 - \pi_{si})(1 - 2\pi_{si}), \\ \nu_{\alpha_i, \gamma_s, \alpha_i} &= -\nu_{\alpha_i \gamma_s \alpha_i} = \pi_{si}(1 - \pi_{si})(1 - 2\pi_{si}), \\ \nu_{\alpha_i, \alpha_i, \alpha_i} &= -\nu_{\alpha_i \alpha_i \alpha_i} = \sum_{s=1}^S \pi_{si}(1 - \pi_{si})(1 - 2\pi_{si}). \end{aligned}$$

Si semplifica di conseguenza anche il calcolo (1.30) delle componenti utili per la modifica, riportato nelle formule di seguito. I cambi di notazione sono dovuti a preservare il più possibile la chiarezza a discapito forse della semplicità di lettura. Ricordando che la matrice $\bar{A}(\theta) = H(\theta)i^{-1}(\theta)$ e quindi $\bar{A}_{\theta,r,a} = i^{\theta_r a} (i^{\theta_r \theta_r})^{-1}$ in cui non è implicita la sommatori in r , si ha

$$\begin{aligned}
k_{1\theta_r} &= -\frac{1}{2} \bar{A}_{\theta_r,a} \nu^{bc} (\nu_{a,b,c} + \nu_{a,bc}) \\
&= -\frac{1}{2} \sum_{a=1}^p \sum_{b=1}^p \sum_{c=1}^p \bar{A}_{\theta_r,a} \nu^{bc} (\nu_{a,b,c}) \\
&= -\frac{1}{2} \left[\sum_{s=1}^S \sum_{b=1}^p \sum_{c=1}^p \bar{A}_{\theta_r,\gamma_s} \nu^{bc} \nu_{\gamma_s,b,c} + \sum_{i=2}^I \sum_{b=1}^p \sum_{c=1}^p \bar{A}_{\theta_r,\alpha_i} \nu^{bc} \nu_{\alpha_i,b,c} \right] \\
&= -\frac{1}{2} \left[\sum_{s=1}^S \bar{A}_{\theta_r,\gamma_s} \left(\nu^{\gamma_s \gamma_s} \nu_{\gamma_s,\gamma_s,\gamma_s} + \sum_{i=2}^I \nu^{\gamma_s \alpha_i} \nu_{\gamma_s,\gamma_s,\alpha_i} + \sum_{i=2}^I \nu^{\alpha_i \alpha_i} \nu_{\gamma_s,\alpha_i,\alpha_i} \right) + \right. \\
&\quad \left. + \sum_{i=2}^I \bar{A}_{\theta_r,\alpha_i} \left(\sum_{s=1}^S \nu^{\gamma_s \gamma_s} \nu_{\alpha_i,\gamma_s,\gamma_s} + \sum_{s=1}^S \nu^{\alpha_i \gamma_s} \nu_{\alpha_i,\alpha_i,\gamma_s} + \nu^{\alpha_i \alpha_i} \nu_{\alpha_i,\alpha_i,\alpha_i} \right) \right],
\end{aligned} \tag{3.2}$$

$$k_{2\theta_r} = (i^{rr})^{-1},$$

$$\begin{aligned}
k_{3\theta_r} &= \bar{A}_{\theta_r,a} \bar{A}_{\theta_r,b} \bar{A}_{\theta_r,c} \nu_{a,b,c} \\
&= \sum_{a=1}^p \sum_{b=1}^p \sum_{c=1}^p \bar{A}_{\theta_r,a} \bar{A}_{\theta_r,b} \bar{A}_{\theta_r,c} \nu_{a,b,c} \\
&= \sum_{s=1}^S \sum_{b=1}^p \sum_{c=1}^p \bar{A}_{\theta_r,\gamma_s} \bar{A}_{\theta_r,b} \bar{A}_{\theta_r,c} \nu_{\gamma_s,b,c} + \sum_{i=2}^I \sum_{b=1}^p \sum_{c=1}^p \bar{A}_{\theta_r,\alpha_i} \bar{A}_{\theta_r,b} \bar{A}_{\theta_r,c} \nu_{\alpha_i,b,c} \\
&= \sum_{s=1}^S \bar{A}_{\theta_r,\gamma_s} \left(\bar{A}_{\theta_r,\gamma_s}^2 \nu_{\gamma_s,\gamma_s,\gamma_s} + 2 \sum_{i=2}^I \bar{A}_{\theta_r,\gamma_s} \bar{A}_{\theta_r,\alpha_i} \nu_{\gamma_s,\gamma_s,\alpha_i} + \sum_{i=2}^I \bar{A}_{\theta_r,\alpha_i}^2 \nu_{\gamma_s,\alpha_i,\alpha_i} \right) + \\
&\quad + \sum_{i=2}^I \bar{A}_{\theta_r,\alpha_i} \left(\sum_{s=1}^S \bar{A}_{\theta_r,\gamma_s}^2 \nu_{\gamma_s,\gamma_s,\gamma_s} + 2 \sum_{s=1}^S \bar{A}_{\theta_r,\alpha_i} \bar{A}_{\theta_r,\gamma_s} \nu_{\alpha_i,\alpha_i,\gamma_s} + \bar{A}_{\theta_r,\alpha_i}^2 \nu_{\alpha_i,\alpha_i,\alpha_i} \right).
\end{aligned}$$

Per meglio comprendere il vantaggio di questo approccio si presenta in Tabella 3.1 un breve studio di simulazione per confrontare i tempi computazionali

di diverse implementazioni. Tutte le funzioni presentate calcolano la modifica del vettore dei parametri $m(\theta) = -k_1 + \frac{1}{6} \frac{k_3}{k_2}$ dati i valori di θ , calcolando ed invertendo internamente anche la matrice $i(\theta)$. Esse quindi costituiscono solamente parte di ogni passo dell'algoritmo ricorsivo di stima che è presentato per intero nel Paragrafo 3.1.4. Si confrontano fra loro cinque funzioni implementate in R (2016a) che producono lo stesso risultato e differiscono solamente per il codice implementato, riportato in Appendice C. Le varie funzioni sono

- **Fun 1** Funzione che implementa il calcolo della modifica nel modo più fedele possibile alla definizione matematica. Prima calcola il tensore $\nu_{a,b,c}$ ed il cumulante k_2 e poi attraverso tre cicli per ogni elemento r -esimo di θ calcola k_{1r} e k_{3r} per un totale di quattro cicli sull'intero vettore dei parametri. La funzione è scritta appositamente per accompagnare il lettore dalla formula matematica alla sua traduzione in codice, e non è considerata come *benchmark*.
- **Fun 2** Funzione di riferimento per il pacchetto `mbrglm` di Kenne Pagui *et al.* (2016b) per la famiglia binomiale. Calcola prima l'intero tensore $\nu_{a,b,c}$ e k_2 e poi attraverso un totale di due cicli i rimanenti vettori k_1 e k_3 . Questo progetto di tesi mira a migliorare l'efficienza di questa funzione in particolare in situazioni in cui i e $\nu_{a,b,c}$ sono sparse.
- **Fun 3** Funzione che calcola k_2 ed in un unico passaggio k_1 e k_3 grazie alle formule presentate in questa sezione, in particolare il calcolo degli elementi di $\nu_{a,b,c}$ non è svolto precedentemente, con guadagni anche in termini di memoria occupata.
- **Fun 4** Analoga a Fun 3 per il calcolo dei tre cumulanti, presenta una versione parallelizzata del ciclo esterno in $r = 1, \dots, p$ grazie ai pacchetti `doParallel` e `foreach` (si veda Calaway, 2015a e 2015b). Ad ogni richiamo della funzione essa distribuisce in modo efficiente i calcoli usando un numero di *core* definito dall'utente, in questo esempio ne sono stati utilizzati due.
- **Fun 5** Analoga a Fun 3 ma con il calcolo di k_1 e k_3 implementati in C++ attraverso il pacchetto `Rcpp` di Eddelbuettel (2016a) presentato

in Appendice A. La funzione presenta una parte che deve essere compilata la prima volta soltanto, per un tempo medio di circa 15 secondi (indipendentemente dalla numerosità campionaria). Questa versione, al netto di qualche espediente tecnico, è quella inserita nella funzione finale `mbrRasch.1pl`.

| Sogg | Item | Parametri | Fun 1 | Fun 2 | Fun 3 | Fun 4 | Fun 5 |
|------|------|-----------|---------|--------|---------|---------|--------|
| 20 | 5 | 24 | 7.44 | 0.14 | 0.12 | 0.91 | 0.00 |
| 50 | 10 | 59 | 231.82 | 1.69 | 0.49 | 1.11 | 0.03 |
| 100 | 20 | 119 | 3826.93 | 35.32 | 2.22 | 2.10 | 0.18 |
| 100 | 30 | 129 | – | 54.56 | 2.79 | 2.42 | 0.28 |
| 150 | 30 | 179 | – | 313.56 | 5.81 | 4.40 | 0.76 |
| 200 | 30 | 229 | – | 999.68 | 11.21 | 7.78 | 1.48 |
| 500 | 50 | 549 | – | – | 291.29 | 169.04 | 51.43 |
| 1000 | 50 | 1049 | – | – | 2025.64 | 1217.48 | 458.36 |

Tabella 3.1: Confronto tempi computazionali in secondi per il calcolo della modifica $m(\theta)$.

Dalla Tabella 3.1 si può notare come la prima funzione sia totalmente inefficiente come era prevedibile. La funzione `mbrglm` è in grado di fornire delle stime fino ad un numero consistente di parametri, ma ha forti problemi di scalabilità. La terza funzione è quella che compie il primo salto di qualità perché in grado di sfruttare in modo ottimale la sparsità della matrice indotta dal modello, comportando una riduzione notevole della complessità dell'algoritmo richiesto al calcolo dei cumulanti. La quarta e la quinta funzione sono miglioramenti della terza che si basano sulle stesse formule. La quarta sfrutta il calcolo parallelo, i cui vantaggi si osservano solo a partire da un numero di parametri sopra 100, evidentemente perché la gestione di più *core* implica dei tempi computazionali fissi utili solo dopo una certa soglia di complessità. Infine, l'ultima funzione è quella che, grazie all'integrazione con C++, risulta la più efficiente e non presenta problemi di scala evidenti fino a 200 parametri.

3.1.4 Algoritmo di stima

L'equazione di stima presentata in (1.28) è riscrivibile nella forma

$$\tilde{U}(\theta) = \bar{U}(\theta) + m(\theta) = 0, \quad (3.3)$$

dove

$$m(\theta) = -k_{1\theta} + \frac{1}{6} \frac{k_{3\theta}}{k_{2\theta}}. \quad (3.4)$$

Per isolare la componente di score semplice dalla modifica apportata si può esplicitare la formula dell'*efficient score* attraverso la scrittura $\bar{U}(\theta) = A(\theta)U(\theta)$ con $A = Hi^{-1}$ (si veda Paragrafo 1.3) e quindi vale la riscrittura $A^{-1}(\theta)m(\theta) = i(\theta)\text{diag}(i^{rr})m(\theta) = i(\theta)m_1(\theta)$ con $m_{1r} = [m_r i^{rr}]$ dove si sottolinea che non è implicita una sommatoria nell'indice r . Quindi, si ha

$$\tilde{U}(\theta) = 0, \quad (3.5)$$

$$A(\theta)U(\theta) + m(\theta) = 0,$$

$$U(\theta) + A^{-1}(\theta)m(\theta) = 0,$$

$$U(\theta) + i(\theta)m_1(\theta) = 0. \quad (3.6)$$

Dall'ultima equazione di stima si può procedere alla stima del parametro in via numerica con un'approssimazione dell'algoritmo di *Fisher Scoring*, dove si approssima $\mathbb{E}[\partial\tilde{U}(\theta)/\partial\theta]$ con $i(\theta)$

$$\begin{aligned} \tilde{\theta}^{(k+1)} &= \tilde{\theta}^{(k)} + i^{-1}(\tilde{\theta}^{(k)}) \left[U(\tilde{\theta}^{(k)}) + i(\tilde{\theta}^{(k)})m_1(\tilde{\theta}^{(k)}) \right] \\ &= \tilde{\theta}^{(k)} + m_1(\tilde{\theta}^{(k)}) + i^{-1}(\tilde{\theta}^{(k)})U(\tilde{\theta}^{(k)}). \end{aligned} \quad (3.7)$$

Si può adottare la convenzione che $\tilde{\theta}^{(0)} = 0$ in quanto l'algoritmo ha mostrato stabilità nella convergenza; studi di simulazione mostrano che in media sono sufficienti dai 6 agli 8 passi. Si vuole sottolineare che in questo caso la modifica per la non distorsione in mediana entra solamente in modo additivo nella formula ricorsiva. Inoltre è facile mostrare che si può ottenere la forma chiusa analoga all'algoritmo IWLS inserendo opportunamente la modifica nella *adjusted response* z . Si ottiene così una formulazione elegante

ed analoga al contesto dei modelli lineari generalizzati. In particolare, siano

$$\begin{aligned}
 U(\theta) &= X^\top(y - \pi(\theta)) = X^\top W(\theta)v(\theta), \\
 W(\theta) &= \text{diag}\{\pi(\theta)(1 - \pi(\theta))\}, \\
 v(\theta) &= W^{-1}(\theta)(y - \pi(\theta)), \\
 i(\theta) &= X^\top W(\theta)X.
 \end{aligned} \tag{3.8}$$

Quindi si può riscrivere la (3.7) nel seguente modo

$$\begin{aligned}
 \tilde{\theta}^{(k+1)} &= \tilde{\theta}^{(k)} + i^{-1}(\tilde{\theta}^{(k)})\{U(\tilde{\theta}^{(k)}) + i(\tilde{\theta}^{(k)})m_1(\tilde{\theta}^{(k)})\}, \\
 i(\tilde{\theta}^{(k)})\tilde{\theta}^{(k+1)} &= i(\tilde{\theta}^{(k)})\tilde{\theta}^{(k)} + U(\tilde{\theta}^{(k)}) + i(\tilde{\theta}^{(k)})m_1(\tilde{\theta}^{(k)}) \\
 &= i(\tilde{\theta}^{(k)})\left[\tilde{\theta}^{(k)} + m_1(\tilde{\theta}^{(k)})\right] + U(\tilde{\theta}^{(k)}), \\
 X^\top W(\tilde{\theta}^{(k)})X\tilde{\theta}^{(k+1)} &= X^\top W(\tilde{\theta}^{(k)})X\left[\tilde{\theta}^{(k)} + m_1(\tilde{\theta}^{(k)})\right] + X^\top W(\tilde{\theta}^{(k)})v(\tilde{\theta}^{(k)}) \\
 &= X^\top W(\tilde{\theta}^{(k)})\left[X\left(\tilde{\theta}^{(k)} + m_1(\tilde{\theta}^{(k)})\right) + v(\tilde{\theta}^{(k)})\right] \\
 &= X^\top W(\tilde{\theta}^{(k)})\tilde{z}(\tilde{\theta}^{(k)}),
 \end{aligned} \tag{3.9}$$

ottenendo infine

$$\tilde{\theta}^{(k+1)} = \left(X^\top W(\tilde{\theta}^{(k)})X\right)^{-1} X^\top W(\tilde{\theta}^{(k)})\tilde{z}(\tilde{\theta}^{(k)}). \tag{3.10}$$

3.2 Modello di Rasch a due parametri

3.2.1 Matrice dei dati

La seconda versione del modello di Rasch è un modello non lineare generalizzato (GNLM) perchè presenta due insiemi di parametri relativi agli item, α e β , che entrano nel predittore η in modo additivo e moltiplicativo rispettivamente. Per trattare la non linearità occorre ampliare la teoria relativa ai modelli lineari generalizzati ed in questo paragrafo si presenta solamente la parte relativa all'inferenza. Per motivi che saranno chiariti nel Paragrafo 3.3.4 è sufficiente far dipendere la matrice del disegno dai parametri, tale che $X = X(\theta) = X(\gamma, \alpha, \beta)$, per ottenere un algoritmo di stima di tipo *Fisher-Scoring* analogo al caso più semplice del modello di Rasch ad un parametro.

La nuova matrice del disegno è di dimensione $n \times p$ con $n = SI$ e $p = S + 2(I - 1)$ per la presenza dei due vincoli di identificabilità ($\alpha_1 = 0, \beta_1 = 1$) ed è definita secondo la regola $\{x_{hr}\} = \partial\eta_h / \partial\theta_r$. Poichè $\partial\eta_{si} / \partial\gamma_s = \beta_i$, $\partial\eta_{si} / \partial\alpha_i = 1$ e $\partial\eta_{si} / \partial\beta_i = \gamma_s$, allora la matrice ha una struttura riportata in (3.11). In particolare, è riconoscibile la divisione in blocchi indotta nelle colonne dalla partizione del vettore dei parametri $\theta = (\gamma, \alpha, \beta)$ e nelle righe dalla vettorizzazione della matrice delle variabili risposta $Y = \{y_{si}\}$, analogamente al caso del primo modello. La matrice mantiene la sua caratteristica sparsità.

$$X(\theta) = \left[\begin{array}{cccc|cccc|cccc|cccc} \beta_1 & 0 & \cdots & 0 & 0 & 0 & \cdots & \cdots & 0 & 0 & 0 & \cdots & \cdots & 0 & 0 & 0 & \cdots & \cdots & 0 \\ 0 & \beta_1 & & 0 & 0 & 0 & & & 0 & 0 & 0 & & & 0 & & & & & 0 \\ \vdots & & \ddots & \vdots & \vdots & \vdots & & & \vdots & \vdots & \vdots & & & \vdots & \vdots & \vdots & & & \vdots \\ 0 & 0 & \cdots & \beta_1 & 0 & 0 & \cdots & \cdots & 0 & 0 & 0 & \cdots & \cdots & 0 & 0 & 0 & \cdots & \cdots & 0 \\ \hline \beta_2 & 0 & \cdots & 0 & 1 & 0 & \cdots & \cdots & 0 & \gamma_1 & 0 & \cdots & \cdots & 0 & & & & & 0 \\ 0 & \beta_2 & & 0 & 1 & 0 & & & 0 & \gamma_2 & 0 & & & 0 & & & & & 0 \\ \vdots & & \ddots & \vdots & \vdots & \vdots & & & \vdots & \vdots & \vdots & & & \vdots & \vdots & \vdots & & & \vdots \\ 0 & 0 & \cdots & \beta_2 & 1 & 0 & \cdots & \cdots & 0 & \gamma_S & 0 & \cdots & \cdots & 0 & & & & & 0 \\ \hline \ddots & 0 & \cdots & 0 & 0 & \vdots & 0 & \cdots & 0 & 0 & \vdots & 0 & \cdots & 0 & 0 & \vdots & 0 & \cdots & 0 \\ 0 & \ddots & & 0 & 0 & \vdots & 0 & & 0 & 0 & \vdots & 0 & & 0 & 0 & \vdots & 0 & & 0 \\ \vdots & & \ddots & 0 & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & & & \vdots \\ 0 & 0 & \cdots & \ddots & 0 & \vdots & 0 & \cdots & 0 & 0 & \vdots & 0 & \cdots & 0 & 0 & \vdots & 0 & \cdots & 0 \\ \hline \beta_I & 0 & \cdots & 0 & 0 & \cdots & \cdots & 0 & 1 & 0 & \cdots & \cdots & 0 & \gamma_1 & & & & & \\ 0 & \beta_I & & 0 & 0 & & & 0 & 1 & 0 & & & 0 & \gamma_2 & & & & & \\ \vdots & & \ddots & \vdots & \vdots & & & \vdots & 1 & \vdots & & & \vdots & \vdots & & & & & \vdots \\ 0 & 0 & \cdots & \beta_I & 0 & \cdots & \cdots & 0 & 1 & 0 & \cdots & \cdots & 0 & \gamma_S & & & & & \end{array} \right] \quad (3.11)$$

3.2.2 Verosimiglianza e momenti nulli

Si presentano la verosimiglianza, presentata nella (2.2), e le relative derivate utili per il calcolo dei momenti nulli e dei cumulanti. Si mantiene la notazione indiciale presentata nel Paragrafo 3.1.2.

La verosimiglianza e la log-verosimiglianza sono:

$$\begin{aligned}
 L(\theta) &= \prod_{s=1}^S \prod_{i=1}^I \pi_{si}^{y_{si}} (1 - \pi_{si})^{1-y_{si}}, \\
 l(\theta) &= \sum_{s=1}^S \sum_{i=1}^I [y_{si} \log \pi_{si} + (1 - y_{si}) \log(1 - \pi_{si})] \\
 &= \sum_{s=1}^S \sum_{i=1}^I y_{si} \beta_i \gamma_s + \sum_{i=1}^I \alpha_i \sum_{s=1}^S y_{si} - \sum_{s=1}^S \sum_{i=1}^I \log(1 + e^{\beta_i \gamma_s + \alpha_i}),
 \end{aligned}$$

le derivate prime e seconde sono rispettivamente:

$$\begin{aligned}
 U_{\gamma_s} &= \sum_{i=1}^I \beta_i (y_{si} - \pi_{si}), \\
 U_{\alpha_i} &= \sum_{s=1}^S (y_{si} - \pi_{si}), \\
 U_{\beta_i} &= \sum_{s=1}^S \gamma_s (y_{si} - \pi_{si}),
 \end{aligned}$$

$$\begin{aligned}
 U_{\gamma_s \gamma_s} &= - \sum_{i=1}^I \beta_i^2 \pi_{si} (1 - \pi_{si}), \\
 U_{\alpha_i \alpha_i} &= - \sum_{s=1}^S \pi_{si} (1 - \pi_{si}), \\
 U_{\beta_i \beta_i} &= - \sum_{s=1}^S \gamma_s^2 \pi_{si} (1 - \pi_{si}),
 \end{aligned}$$

$$\begin{aligned}
 U_{\gamma_s \alpha_i} &= - \beta_i \pi_{si} (1 - \pi_{si}), \\
 U_{\gamma_s \beta_i} &= (y_{si} - \pi_{si}) - \beta_i \gamma_s \pi_{si} (1 - \pi_{si}), \\
 U_{\alpha_i \beta_i} &= - \sum_{s=1}^S \gamma_s \pi_{si} (1 - \pi_{si}).
 \end{aligned}$$

Tutti gli altri elementi della forma $U_{\gamma_s \gamma_t}$, U_{α_i, α_j} , U_{β_i, β_j} sono nulli per l'indipendenza fra tutti i soggetti e l'indipendenza fra tutti gli item. Ne risulta che la derivata seconda ha una struttura molto simile a quella del modello

di Rasch ad un parametro. In questo caso in Figura 3.4 si distinguono sei blocchi indotti dalla partizione $\theta = (\gamma, \alpha, \beta)$. Tutti i blocchi sono non stocastici ad eccezione del blocco $U_{\gamma\beta}$ il cui valore atteso cambiato di segno è $i_{\gamma\beta} = -\mathbb{E}[(Y_{si} - \pi_{si}) - \beta_i\gamma_s\pi_{si}(1 - \pi_{si})] = \beta_i\gamma_s\pi_{si}(1 - \pi_{si})$. Per simmetria col primo modello si riporta in Figura 3.9 il confronto fra struttura formale ed un esempio della matrice. La struttura dei blocchi è dovuta alle ipotesi

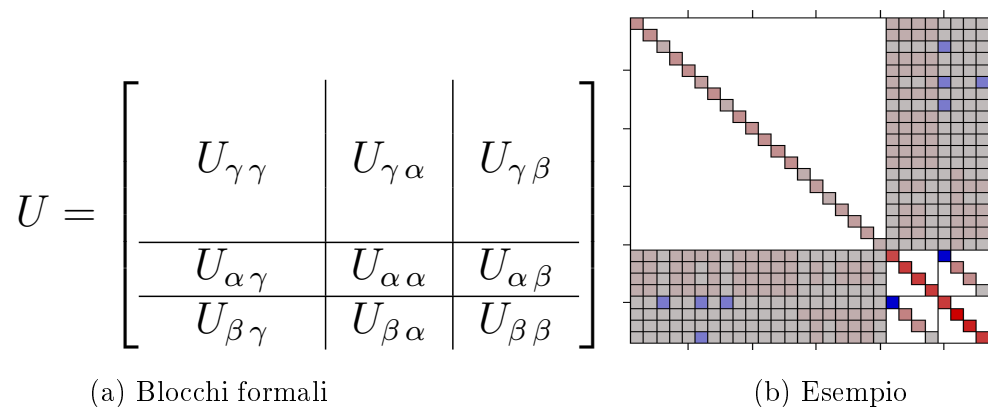


Figura 3.9: Divisione a blocchi della matrice di derivate seconde U .

già presentate nel primo modello, si sottolinea solamente che anche il blocco $U_{\alpha\beta}$ è diagonale perchè $i_{\alpha_i,\beta_j} \neq 0$ solo se $i = j$, in quanto l'indipendenza fra item si traduce in ortogonalità dei parametri. Al crescere dei dati disponibili, l'informazione relativa ai parametri cresce a velocità diverse; infatti $i_{\gamma\gamma}$ è di ordine $O(I)$, $i_{\alpha\alpha}$, $i_{\beta\beta}$ e $i_{\alpha\beta}$ sono di ordine $O(S)$, infine $i_{\gamma\alpha}$ e $i_{\gamma\beta}$ sono di ordine $O(1)$ perchè c'è solo un'osservazione per la coppia soggetto-item, indipendentemente dal numero di soggetti o item considerati.

Il tensore delle derivate terze U_{abc} è interamente non stocastico, proprietà che semplifica notevolmente i calcoli dei momenti nulli $\nu_{a,b,c}$. È suddivisibile in 27 blocchi ($3 \times 3 \times 3$) che per simmetria si riducono a 10 (così come nella matrice a 9 blocchi è sufficiente calcolarne 6) che sono presentati nelle formule di seguito, in particolare U_{abc} eredita la sparsità dalla matrice U_{ab} e quindi a loro volta i blocchi presentano molti elementi nulli, si dimostra che $U_{\gamma_s\gamma_t\gamma_t} = U_{\alpha_i\alpha_i\alpha_j} = U_{\beta_i\beta_i\beta_j} = 0$, $U_{\alpha_i\alpha_i\beta_j} = U_{\beta_i\beta_i\alpha_j} = 0$ e $U_{\gamma_s\alpha_i\beta_j} = 0$.

La derivata terza ha quindi elementi:

$$\begin{aligned}
U_{\gamma_s \gamma_s \gamma_s} &= -\sum_{i=1}^I \beta_i^3 \pi_{si} (1 - \pi_{si}) (1 - 2\pi_{si}), \\
U_{\alpha_i \alpha_i \alpha_i} &= -\sum_{s=1}^S \pi_{si} (1 - \pi_{si}) (1 - 2\pi_{si}), \\
U_{\beta_i \beta_i \beta_i} &= -\sum_{s=1}^S \gamma_s^3 \pi_{si} (1 - \pi_{si}) (1 - 2\pi_{si}), \\
\\
U_{\gamma_s \gamma_s \alpha_i} &= -\beta_i^2 \pi_{si} (1 - \pi_{si}) (1 - 2\pi_{si}), \\
U_{\gamma_s \gamma_s \beta_i} &= -\beta_i \pi_{si} (1 - \pi_{si}) (2 + \beta_i \gamma_s (1 - 2\pi_{si})), \\
U_{\alpha_i \alpha_i \gamma_s} &= -\beta_i \pi_{si} (1 - \pi_{si}) (1 - 2\pi_{si}), \\
U_{\alpha_i \alpha_i \beta_i} &= -\sum_{s=1}^S \gamma_s \pi_{si} (1 - \pi_{si}) (1 - 2\pi_{si}), \\
U_{\beta_i \beta_i \gamma_s} &= -\gamma_s \pi_{si} (1 - \pi_{si}) (2 + \beta_i \gamma_s (1 - 2\pi_{si})), \\
U_{\beta_i \beta_i \alpha_i} &= -\sum_{s=1}^S \gamma_s^2 \pi_{si} (1 - \pi_{si}) (1 - 2\pi_{si}), \\
U_{\gamma_s \alpha_i \beta_i} &= -\pi_{si} (1 - \pi_{si}) (1 + \beta_i \gamma_s (1 - 2\pi_{si})).
\end{aligned}$$

Il calcolo dei momenti nulli è basato sulle relazioni di Bartlett (1.6). Per quanto presentato prima vale che:

$$\begin{aligned}
\nu_{abc} &= \mathbb{E}[U_{abc}] = U_{abc}, \\
\nu_{a,bc} &= \mathbb{E}[U_a U_{bc}] = 0,
\end{aligned}$$

con l'unica eccezione del blocco stocastico $U_{\gamma\beta}$ che porta alle relazioni:

$$\begin{aligned}
\nu_{\gamma_s, \gamma_s \beta_i} &= \beta_i \pi_{si} (1 - \pi_{si}), \\
\nu_{\alpha_i, \gamma_s \beta_i} &= \pi_{si} (1 - \pi_{si}), \\
\nu_{\beta_i, \gamma_s \beta_i} &= \gamma_s \pi_{si} (1 - \pi_{si}).
\end{aligned}$$

Si ottengono, infine, gli elementi non nulli dei dieci blocchi del tensore $\nu_{a,b,c}$

$$\begin{aligned}
\nu_{\gamma_s, \gamma_s, \gamma_s} &= -\nu_{\gamma_s \gamma_s \gamma_s} = \sum_{i=1}^I \beta_i^3 \pi_{si} (1 - \pi_{si}) (1 - 2\pi_{si}), \\
\nu_{\alpha_i, \alpha_i, \alpha_i} &= -\nu_{\alpha_i \alpha_i \alpha_i} = \sum_{s=1}^S \pi_{si} (1 - \pi_{si}) (1 - 2\pi_{si}), \\
\nu_{\beta_i, \beta_i, \beta_i} &= -\nu_{\beta_i \beta_i \beta_i} = \sum_{s=1}^S \gamma_s^3 \pi_{si} (1 - \pi_{si}) (1 - 2\pi_{si}),
\end{aligned}$$

$$\begin{aligned}
\nu_{\gamma_s, \gamma_s, \alpha_i} &= -\nu_{\gamma_s \gamma_s \alpha_i} &= \beta_i^2 \pi_{si} (1 - \pi_{si}) (1 - 2\pi_{si}), \\
\nu_{\gamma_s, \gamma_s, \beta_i} &= -\nu_{\gamma_s \gamma_s \beta_i} - 2\nu_{\gamma_s, \gamma_s \beta_i} &= \beta_i^2 \gamma_s \pi_{si} (1 - \pi_{si}) (1 - 2\pi_{si}), \\
\nu_{\alpha_i, \alpha_i, \gamma_s} &= -\nu_{\alpha_i \alpha_i \gamma_s} &= \beta_i \pi_{si} (1 - \pi_{si}) (1 - 2\pi_{si}), \\
\nu_{\alpha_i, \alpha_i, \beta_i} &= -\nu_{\alpha_i \alpha_i \beta_i} &= \sum_{s=1}^S \gamma_s \pi_{si} (1 - \pi_{si}) (1 - 2\pi_{si}), \\
\nu_{\beta_i, \beta_i, \gamma_s} &= -\nu_{\beta_i \beta_i \gamma_s} - 2\nu_{\beta_i, \beta_i \gamma_s} &= \beta_i \gamma_s^2 \pi_{si} (1 - \pi_{si}) (1 - 2\pi_{si}), \\
\nu_{\beta_i, \beta_i, \alpha_i} &= -\nu_{\beta_i \beta_i \alpha_i} &= \sum_{s=1}^S \gamma_s^2 \pi_{si} (1 - \pi_{si}) (1 - 2\pi_{si}), \\
\nu_{\gamma_s, \alpha_i, \beta_i} &= -\nu_{\gamma_s \alpha_i \beta_i} - \nu_{\alpha_i, \gamma_s \beta_i} &= \beta_i \gamma_s \pi_{si} (1 - \pi_{si}) (1 - 2\pi_{si}).
\end{aligned}$$

3.2.3 Calcolo dei cumulanti k_1 , k_2 e k_3

Seguendo la linea proposta nel primo modello si riesce a sfruttare la sparsità di $\nu_{a,b,c}$ per ottimizzare il calcolo dei cumulanti, fissando il primo indice a di volta in volta e studiando gli elementi della matrice ottenuta.

- Se a si riferisce a γ_s
allora le matrici di riferimento sono della stessa forma della Figura 3.10

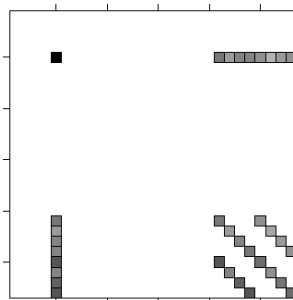


Figura 3.10: Particolare del tensore $\nu_{a,b,c}$ con a corrispondente a γ_5 .

e gli unici elementi non nulli della matrice $\nu_{a,b,c}$ sono:

$$\begin{aligned}
\nu_{\gamma_s, \gamma_s, \gamma_s}, \quad 2\nu_{\gamma_s, \gamma_s, \alpha_i}, \quad 2\nu_{\gamma_s, \gamma_s, \beta_i}, \quad \nu_{\gamma_s, \alpha_i, \alpha_i}, \\
2\nu_{\gamma_s, \alpha_i, \beta_i}, \quad \nu_{\gamma_s, \beta_i, \beta_i}, \quad i = 2, \dots, I.
\end{aligned}$$

- Se a si riferisce a α_i
allora le matrici sono quelle riportate in Figura 3.11 e gli unici elementi non nulli di $\nu_{a,b,c}$ sono:

$$\begin{aligned}
\nu_{\alpha_i, \gamma_s, \gamma_s}, \quad 2\nu_{\alpha_i, \gamma_s, \alpha_i}, \quad 2\nu_{\alpha_i, \gamma_s, \beta_i}, \quad \nu_{\alpha_i, \alpha_i, \alpha_i}, \\
2\nu_{\alpha_i, \alpha_i, \beta_i}, \quad \nu_{\alpha_i, \beta_i, \beta_i}, \quad s = 1, \dots, S.
\end{aligned}$$

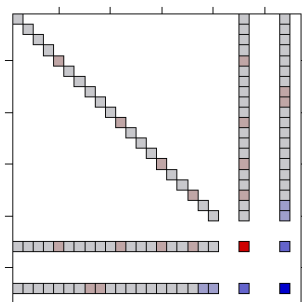


Figura 3.11: Particolare del tensore $\nu_{a,b,c}$ con a corrispondente a α_4 .

- Se a si riferisce a β_i allora le matrici sono quelle riportate in Figura 3.12, qualitativamente

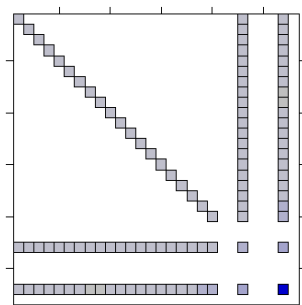


Figura 3.12: Particolare del tensore $\nu_{a,b,c}$ con a corrispondente a β_4 .

identiche alla 3.11. Gli unici elementi non nulli di $\nu_{a,b,c}$ sono:

$$\begin{aligned} &\nu_{\beta_i, \gamma_s, \gamma_s}, \quad 2\nu_{\beta_i, \gamma_s, \alpha_i}, \quad 2\nu_{\beta_i, \gamma_s, \beta_i}, \nu_{\beta_i, \alpha_i, \alpha_i}, \\ &2\nu_{\beta_i, \alpha_i, \beta_i}, \quad \nu_{\beta_i, \beta_i, \beta_i}, \quad s = 1, \dots, S. \end{aligned}$$

Il tensore $\nu_{a,b,c}$ è tutto nullo tranne i blocchi di tipo $\nu_{a,\gamma\beta}$ (nelle figure in alto a destra e in basso a sinistra) per cui seguendo la stessa logica utilizzata fin ora si distinguono tre casi:

- Se a si riferisce a γ_s allora gli unici elementi non nulli della matrice $\nu_{a,b,c}$ in Figura 3.13 sono

$$2\nu_{\gamma_s, \gamma_s, \beta_i}, \quad i = 2, \dots, I.$$

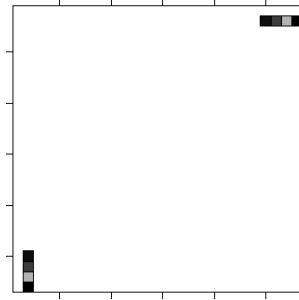


Figura 3.13: Particolare del tensore $\nu_{a,bc}$ con a corrispondente a γ_5 .

- Se a si riferisce a α_i allora gli unici elementi non nulli della matrice $\nu_{a,bc}$ in Figura 3.14 sono

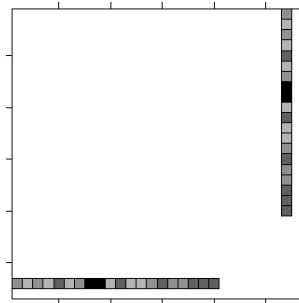


Figura 3.14: Particolare del tensore $\nu_{a,bc}$ con a corrispondente a α_4 .

$$2\nu_{\alpha_i, \gamma_s \beta_i}, \quad s = 1, \dots, S.$$

- Se a si riferisce a β_i allora gli unici elementi non nulli della matrice $\nu_{a,bc}$ in Figura 3.15 sono

$$2\nu_{\beta_i, \gamma_s \beta_i}, \quad s = 1, \dots, S.$$

L'analisi dei tensori porta alla semplificazione delle formule dei cumulanti k_1 e k_2 in quanto si riesce a considerare solo gli elementi non nulli per ogni casistica dell'indice a , eliminando due cicli di indici interni (quelli in b e c nella formula originaria). Di seguito i risultati, che generalizzano quelli riportati

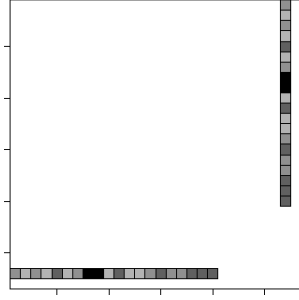


Figura 3.15: Particolare del tensore $\nu_{a,bc}$ con a corrispondente a β_4 .

in (3.2).

$$\begin{aligned}
 k_{1\theta_r} &= -\frac{1}{2}\bar{A}_{\theta_r a}\nu^{bc}(\nu_{a,b,c} + \nu_{a,bc}) \\
 &= \sum_{a=1}^p -\frac{1}{2}\bar{A}_{\theta_r a} \sum_{b=1}^p \sum_{c=1}^p \nu^{bc}(\nu_{a,b,c} + \nu_{a,bc}) \\
 &= \sum_{a=1}^p \delta_1(a),
 \end{aligned}$$

$$k_{2\theta_r} = (i^{rr})^{-1},$$

$$\begin{aligned}
 k_{3\theta_r} &= \bar{A}_{\theta_r a}\bar{A}_{\theta_r b}\bar{A}_{\theta_r c}\nu_{a,b,c} \\
 &= \sum_{a=1}^p \bar{A}_{\theta_r a} \sum_{b=1}^p \bar{A}_{\theta_r b} \sum_{c=1}^p \bar{A}_{\theta_r c}\nu_{a,b,c} \\
 &= \sum_{a=1}^p \delta_3(a),
 \end{aligned}$$

dove $\delta_1(a)$ e $\delta_3(a)$ assumono valore in funzione di a :

Se a si riferisce a γ_s

$$\begin{aligned}
 \delta_1(s) &= -\frac{1}{2}\bar{A}_{\theta_r \gamma_s} \left[\nu^{\gamma_s \gamma_s} \nu_{\gamma_s, \gamma_s, \gamma_s} + 2 \sum_{i=2}^I \nu^{\gamma_s \alpha_i} \nu_{\gamma_s, \gamma_s, \alpha_i} + \right. \\
 &\quad \left. + \sum_{i=2}^I \nu^{\gamma_s \beta_i} (\nu_{\gamma_s, \gamma_s, \beta_i} + \nu_{\gamma_s, \gamma_s \beta_i}) + \sum_{i=2}^I \nu^{\alpha_i \alpha_i} \nu_{\gamma_s, \alpha_i, \alpha_i} + \right. \\
 &\quad \left. + 2 \sum_{i=2}^I \nu^{\alpha_i \beta_i} \nu_{\gamma_s, \alpha_i, \beta_i} + \sum_{i=2}^I \nu^{\beta_i \beta_i} \nu_{\gamma_s, \beta_i, \beta_i} \right],
 \end{aligned}$$

$$\delta_3(s) = \bar{A}_{\theta_r \gamma_s} \left[\bar{A}_{\theta_r \gamma_s}^2 \nu_{\gamma_s, \gamma_s, \gamma_s} + 2 \sum_{i=2}^I \bar{A}_{\theta_r \gamma_s} \bar{A}_{\theta_r \alpha_i} \nu_{\gamma_s, \gamma_s, \alpha_i} + \right.$$

$$+ 2 \sum_{i=2}^I \bar{A}_{\theta_r \gamma_s} \bar{A}_{\theta_r \beta_i} \nu_{\gamma_s, \gamma_s, \beta_i} + \sum_{i=2}^I \bar{A}_{\theta_r \alpha_i}^2 \nu_{\gamma_s, \alpha_i, \alpha_i} + \\ + 2 \sum_{i=2}^I \bar{A}_{\theta_r \alpha_i} \bar{A}_{\theta_r \beta_i} \nu_{\gamma_s, \alpha_i, \beta_i} + \sum_{i=2}^I \bar{A}_{\theta_r \beta_i}^2 \nu_{\gamma_s, \beta_i, \beta_i} \Big].$$

Se a si riferisce a α_i

$$\delta_1(i) = -\frac{1}{2} \bar{A}_{\theta_r \alpha_i} \left[\sum_{s=1}^S \nu^{\gamma_s \gamma_s} \nu_{\beta_i, \gamma_s, \gamma_s} + \sum_{s=1}^S \nu^{\gamma_s \alpha_i} \nu_{\alpha_i, \gamma_s, \alpha_i} + \right. \\ \left. + 2 \sum_{s=1}^S \nu^{\gamma_s \beta_i} (\nu_{\alpha_i, \gamma_s, \beta_i} + \nu_{\alpha_i, \gamma_s \beta_i}) + \nu^{\alpha_i \alpha_i} \nu_{\alpha_i, \alpha_i, \alpha_i} + \right. \\ \left. + 2 \nu^{\alpha_i \beta_i} \nu_{\alpha_i, \alpha_i, \beta_i} + \nu^{\beta_i \beta_i} \nu_{\alpha_i, \beta_i, \beta_i} \right],$$

$$\delta_3(i) = \bar{A}_{\theta_r \alpha_i} \left[\sum_{s=1}^S \bar{A}_{\theta_r \gamma_s}^2 \nu_{\alpha_i, \gamma_s, \gamma_s} + 2 \sum_{s=1}^S \bar{A}_{\theta_r \gamma_s} \bar{A}_{\theta_r \alpha_i} \nu_{\alpha_i, \gamma_s, \alpha_i} + \right. \\ \left. + 2 \sum_{s=1}^S \bar{A}_{\theta_r \gamma_s} \bar{A}_{\theta_r \beta_i} \nu_{\alpha_i, \gamma_s, \beta_i} + \bar{A}_{\theta_r \alpha_i}^2 \nu_{\alpha_i, \alpha_i, \alpha_i} + \right. \\ \left. + 2 \bar{A}_{\theta_r \alpha_i} \bar{A}_{\theta_r \beta_i} \nu_{\alpha_i, \alpha_i, \beta_i} + \bar{A}_{\theta_r \beta_i}^2 \nu_{\alpha_i, \beta_i, \beta_i} \right].$$

Se a si riferisce a β_i

$$\delta_1(i) = -\frac{1}{2} \bar{A}_{\theta_r \beta_i} \left[\sum_{s=1}^S \nu^{\gamma_s \gamma_s} \nu_{\beta_i, \gamma_s, \gamma_s} + \sum_{s=1}^S \nu^{\gamma_s \alpha_i} \nu_{\beta_i, \gamma_s, \alpha_i} + \right. \\ \left. + 2 \sum_{s=1}^S \nu^{\gamma_s \beta_i} (\nu_{\beta_i, \gamma_s, \beta_i} + \nu_{\beta_i, \gamma_s \beta_i}) + \nu^{\alpha_i \alpha_i} \nu_{\beta_i, \alpha_i, \alpha_i} + \right. \\ \left. + 2 \nu^{\alpha_i \beta_i} \nu_{\beta_i, \alpha_i, \beta_i} + \nu^{\beta_i \beta_i} \nu_{\beta_i, \beta_i, \beta_i} \right],$$

$$\delta_3(i) = \bar{A}_{\theta_r \beta_i} \left[\sum_{s=1}^S \bar{A}_{\theta_r \gamma_s}^2 \nu_{\beta_i, \gamma_s, \gamma_s} + 2 \sum_{s=1}^S \bar{A}_{\theta_r \gamma_s} \bar{A}_{\theta_r \alpha_i} \nu_{\beta_i, \gamma_s, \alpha_i} + \right. \\ \left. + 2 \sum_{s=1}^S \bar{A}_{\theta_r \gamma_s} \bar{A}_{\theta_r \beta_i} \nu_{\beta_i, \gamma_s, \beta_i} + \bar{A}_{\theta_r \alpha_i}^2 \nu_{\beta_i, \alpha_i, \alpha_i} + \right. \\ \left. + 2 \bar{A}_{\theta_r \alpha_i} \bar{A}_{\theta_r \beta_i} \nu_{\beta_i, \alpha_i, \beta_i} + \bar{A}_{\theta_r \beta_i}^2 \nu_{\beta_i, \beta_i, \beta_i} \right].$$

La funzione per calcolare la modifica $m(\theta)$ dati i valori di θ è stata scritta in C++ secondo le formule appena riportate. Visti i risultati ottenuti nella versione del modello di Rasch ad un parametro non si è ritenuto necessario implementare la stessa funzione in R per confrontare le velocità di esecuzioni. Si rimanda al Capitolo 4 per maggiori studi simulativi di confronto.

3.2.4 Algoritmo di stima

In generale nell'ambito dei modelli non lineari generalizzati (GNLM) si possono riottenere le quantità classiche dell'inferenza statistica operando opportunamente sulle relazioni note per i modelli lineari generalizzati (GLM), si

veda per una trattazione completa Davison (2003). Per rimanere coerenti con la notazione precedente, si usa la scrittura y_h con $h = 1, \dots, n$ per intendere l' h -esimo elemento del vettore di y ($n \times 1$) che è in corrispondenza biunivoca con l'elemento (s, i) -esimo della matrice Y ($S \times I$).

Siano date le usuali ipotesi di indipendenza di y_1, \dots, y_n e la relazione funzionale $g(\mathbb{E}[Y_h]) = \eta_h$ dove $\eta_h = \eta_h(\theta)$ rappresenti una qualche funzione parametrica, non necessariamente lineare, in cui entrano le variabili esplicative. Allora vale la scrittura

$$l(\theta) = \sum_{h=1}^n \log f(y_h; \eta_h(\theta)),$$

o più semplicemente

$$l(\theta) = \sum_{h=1}^n l_h\{\eta_h(\theta)\}, \quad (3.12)$$

dove $l_h\{\eta_h(\theta)\}$ è il contributo della h -esima osservazione. Le stime di massima verosimiglianza $\hat{\theta}$ soddisfano il ben noto sistema di equazioni di stima

$$U_r(\theta) = \frac{\partial l(\hat{\theta})}{\partial \theta_r} = 0, \quad r = 1, \dots, p, \quad (3.13)$$

che si può scrivere in forma matriciale

$$U(\theta) = \frac{\partial l(\hat{\theta})}{\partial \theta} = \frac{\partial \eta^\top(\hat{\theta})}{\partial \theta} \frac{\partial l(\hat{\theta})}{\partial \eta(\hat{\theta})} = 0, \quad (3.14)$$

dove $n(\theta) = \partial l(\theta)/\partial \eta(\theta)$ è un vettore $n \times 1$. Per trovare le stime di massima verosimiglianza dato un vettore iniziale si può sviluppare in serie di Taylor la formula precedente in un intorno di θ fino al primo ordine.

$$\frac{\partial \eta^\top(\hat{\theta})}{\partial \theta} n(\theta) + \left[\sum_{h=1}^n \frac{\partial \eta_h(\theta)}{\partial \theta} \frac{\partial^2 l_h(\theta)}{\partial \eta_h^2} \frac{\partial \eta_h(\theta)}{\partial \theta^\top} + \sum_{h=1}^n \frac{\partial^2 \eta_h(\theta)}{\partial \theta \partial \theta^\top} n_h(\theta) \right] (\hat{\theta} - \theta) \doteq 0,$$

dalla cui inversione si ottiene la formula di *Newton-Raphson* nel nuovo contesto. Se si chiama la matrice $p \times p$ in parentesi quadrate $-J(\theta)$, che si assume invertibile, si ottiene la usuale formula ricorsiva

$$\hat{\theta} \doteq \theta + J^{-1}(\theta) \frac{\partial \eta^\top(\theta)}{\partial \theta} n(\theta). \quad (3.15)$$

In pratica, è più conveniente sostituire a $J(\theta)$ il suo valore atteso

$$i(\theta) = \sum_{h=1}^n \frac{\partial \eta_h(\theta)}{\partial \theta} \mathbb{E} \left[-\frac{\partial^2 l_h}{\partial \eta_h^2} \right] \frac{\partial \eta_h(\theta)}{\partial \theta^\top},$$

poichè il secondo termine dentro le parentesi quadre si annulla dato che $\mathbb{E}[n_h(\theta)] = 0$. Se si costruisce X tale che $X(\theta) = \partial \eta(\theta) / \partial \theta^\top$, come presentato nel Paragrafo 3.3.1, e W tale che sia una matrice diagonale di dimensione $n \times n$ il cui h -esimo elemento è $\mathbb{E}[-\partial^2 l_h / \partial \eta_h^2]$, allora vale che

$$i(\theta) = X^\top(\theta) W(\theta) X(\theta), \quad (3.16)$$

che rappresenta la naturale estensione alla scrittura dell'informazione nel caso dei modelli non lineari, rispetto al contesto dei modelli lineari generalizzati in cui $i(\theta) = X^\top W(\theta) X$.

Seguendo la nuova notazione è immediato aggiornare le formule (3.8)

$$\begin{aligned} U(\theta) &= X(\theta)^\top (y - \pi(\theta)) = X(\theta)^\top W(\theta) v(\theta), \\ W(\theta) &= \text{diag}[\pi(\theta)(1 - \pi(\theta))], \\ v(\theta) &= W^{-1}(\theta)(y - \pi(\theta)), \\ i(\theta) &= X^\top(\theta) W(\theta) X(\theta), \end{aligned} \quad (3.17)$$

e, data l'equazione di stima (3.4), la formula di *Newton-Raphson* (3.15) rimane formalmente invariata rispetto alla (3.7)

$$\tilde{\theta}^{(k+1)} = \tilde{\theta}^{(k)} + i^{-1}(\tilde{\theta}^{(k)}) \left[U(\tilde{\theta}^{(k)}) + i(\tilde{\theta}^{(k)}) m_1(\tilde{\theta}^{(k)}) \right]. \quad (3.18)$$

Si riesce, infine, ad ottenere con semplici passaggi algebrici la formula analoga alla (3.10) dell'algoritmo IWLS per la stima ricorsiva

$$\tilde{\theta}^{(k+1)} = \left(X^\top(\tilde{\theta}^{(k)}) W(\tilde{\theta}^{(k)}) X(\tilde{\theta}^{(k)}) \right)^{-1} X^\top(\tilde{\theta}^{(k)}) W(\tilde{\theta}^{(k)}) \tilde{z}(\tilde{\theta}^{(k)}), \quad (3.19)$$

dove $\tilde{z}(\theta) = X(\theta)(\theta + m_1(\theta)) + v(\theta)$.

Un problema tipico del modello di Rasch nella formulazione a due parametri è l'instabilità della convergenza e di pari passo la velocità della stessa. L'algoritmo di *Fisher-Scoring* presentato di fatto non converge mai se lasciato libero e senza aver fornito un vettore iniziale non banale. C'è quindi bisogno di modificare il metodo di stima in due direzioni. Per quanto concerne la velocità di ogni iterazione di stima si sono ottenuti risultati soddisfacenti portando in C++ la parte di codice relativa al calcolo della modifica $m(\theta)$. Il rapporto fra i tempi computazionali del codice scritto puramente in R e quello ibrido fra R e C++ sono simili a quelli già presentati in Tabella 3.1 e non vengono riportati ulteriori studi simulativi. Per il problema della stabilità invece si è operato su due fronti. Da una parte si modifica la formula (3.18) per introdurre un passo di *step-halving*, dall'altra si sceglie come punto iniziale $\theta^{(0)}$ la stima del modello Rasch 1PL per γ e α e dei valori perturbati intorno ad 1 per β .

Lo *step-halving* è una tecnica adottata per aver maggiore controllo dell'algoritmo ed evitare che compia un passo di ottimizzazione "troppo grande". Se l'algoritmo stesse convergendo allora è lecito aspettarsi che i vari passi di stima siano ogni volta inferiori ai precedenti rispetto ad una norma prefissata, in formule $|\theta_{k+1} - \theta_k| < |\theta_k - \theta_{k-1}|$. Invece se l'algoritmo stesse divergendo allora prima o poi si osserverebbe un passo di stima superiore al precedente, anche se non è sempre vero il viceversa. Con questa logica si può intervenire per rifiutare in modo automatico un passo troppo ampio e proporre uno inferiore che rispetti il criterio adottato. Nella funzione `mbrRasch.2pl` si è inserita una semplice quanto efficace tecnica di penalizzazione esponenziale del passo: se il passo proposto è già minore del precedente nulla è modificato, in caso contrario viene premoltiplicato per $1/2, 1/4, \dots$ fino alla verifica della condizione. Dopo la penalizzazione $1/2^{10}$ si interrompe la sequenza e si accetta questo ultimo valore in qualsiasi caso, procedendo al prossimo *iter* di Newton-Raphson. Si riporta la parte di codice in cui questa tecnica è implementata

```
1 #step halving alla fine dell'iter di Newton-Raphson
  stepPrev <- step
  stepFactor <- 0
  testhalf <- TRUE
```

```

6   while(testhalf & stepFactor < 11){
      step <- 2^(-stepFactor) * (Infoinv %**% (score + Info %**% m1))
      testhalf <- drop(crossprod(stepPrev) < crossprod(step))
      stepFactor <- stepFactor + 1
    }
11  par <- par + step
    iter = iter +1

```

Infine, è possibile inserire lo step-halving anche nella formula (3.19), con semplici passaggi si ottiene la nuova versione dell'algoritmo IWLS, in cui si sottolinea che la penalizzazione entra solo nella formula dell'adjusted response $\tilde{\tilde{z}}$.

$$\begin{aligned}
\tilde{\theta}^{(k+1)} &= \tilde{\theta}^{(k)} + 2^{-stepfactor} i^{-1}(\tilde{\theta}^{(k)}) \left[U(\tilde{\theta}^{(k)}) + i(\tilde{\theta}^{(k)}) m_1(\tilde{\theta}^{(k)}) \right] \\
&= \left(X^T(\tilde{\theta}^{(k)}) W(\tilde{\theta}^{(k)}) X(\tilde{\theta}^{(k)}) \right)^{-1} X^T(\tilde{\theta}^{(k)}) W(\tilde{\theta}^{(k)}) \tilde{\tilde{z}}(\tilde{\theta}^{(k)}), \quad (3.20)
\end{aligned}$$

dove $\tilde{\tilde{z}}(\theta) = X(\theta)(\theta + 2^{-stepfactor} m_1(\theta)) + 2^{-stepfactor} v(\theta)$.

Dato che la stima di massima verosimiglianza (sia condizionata che non) non è praticabile per il modello logistico a due parametri ad oggi l'unico metodo di stima attuabile per gli effetti fissi è la *bias reduction*. Quest'insieme di accorgimenti appena descritti rendono il metodo di stima non distorto in mediana stabile e veloce, ponendolo come unica alternativa alla proposta del pacchetto di Kosmidis `brRasch` (2016).

Capitolo 4

Studi di simulazione

*“How many roads must a man walk down
before you call him a man?”*

B. Dylan

4.1 Simulazioni per il modello logistico ad un parametro

Si presentano alcuni studi simulativi per la verifica delle proprietà distributive degli stimatori presentati nei Paragrafi 1.2, 1.3 e 2.3: stimatore di massima verosimiglianza (ML), stimatore non distorto in media (BR), stimatore non distorto in mediana (MBR), stimatore di massima verosimiglianza condizionata (CL) e stimatore con effetti casuali per i soggetti (RE). Tutti i modelli sono valutati in funzione di quattro indici: *Root Mean Square Error* (RMSE), *Median Absolute Error* (MAE), percentuale di stime finite (Stime finite (%)) e copertura empirica degli intervalli alla Wald con livello di confidenza al 95% (CI 0.95); questi ultimi sono da interpretare con cautela dato che sono calcolati sulle sole stime finite. Tutti i risultati sono stati ottenuti da $R = 5000$ simulazioni.

I primi risultati sono relativi ad uno studio incentrato sulla stima dei soli item. In Tabella 4.1 si mostra come variano gli indici descrittivi per la stima di $I = 5$ item sia fra i cinque modelli sia all'aumentare del numero di soggetti

S , rappresentativo della quantità di informazione a disposizione per la stima del vettore α . In particolare i valore da cui sono stati generati i dati sono $\alpha = (0, -1.3, -0.5, 0.2, 2.1)$ e $\gamma \sim T_4$.

| S | Stime finite (%) | RMSE | | | | MAE | | | | CI 0.95 | | | | |
|-----|------------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------|
| | | α_2 | α_3 | α_4 | α_5 | α_2 | α_3 | α_4 | α_5 | α_2 | α_3 | α_4 | α_5 | |
| 20 | MLE | 96.7 | 1.69 | 1.00 | 0.66 | 1.77 | 1.00 | 0.54 | 0.49 | 1.17 | 88.5 | 98.4 | 99.9 | 87.8 |
| | BR | 100 | 1.02 | 0.64 | 0.42 | 0.95 | 0.61 | 0.50 | 0.27 | 0.63 | 94.8 | 99.6 | 100 | 95.2 |
| | MBR | 100 | 1.12 | 0.71 | 0.47 | 1.00 | 0.66 | 0.50 | 0.32 | 0.66 | 95.0 | 99.6 | 100 | 95.7 |
| | CL | 96.8 | 5.08 | 2.49 | 2.08 | 5.91 | 0.67 | 0.50 | 0.36 | 0.67 | 53.1 | 74.1 | 99.0 | 98.6 |
| | RE | 98.3 | 1.08 | 0.66 | 0.43 | 1.28 | 0.66 | 0.50 | 0.28 | 0.85 | 93.4 | 97.9 | 98.2 | 91.2 |
| 40 | MLE | 99.7 | 1.36 | 0.77 | 0.38 | 1.60 | 0.82 | 0.50 | 0.20 | 0.99 | 79.0 | 94.2 | 100 | 74.0 |
| | BR | 100 | 0.69 | 0.50 | 0.25 | 0.69 | 0.43 | 0.34 | 0.20 | 0.46 | 94.3 | 98.3 | 100 | 94.4 |
| | MBR | 100 | 0.76 | 0.53 | 0.27 | 0.70 | 0.46 | 0.35 | 0.20 | 0.47 | 93.6 | 98.5 | 100 | 95.1 |
| | CL | 99.8 | 0.76 | 0.54 | 0.29 | 0.70 | 0.46 | 0.35 | 0.20 | 0.45 | 58.0 | 83.7 | 100 | 100 |
| | RE | 99.9 | 0.73 | 0.51 | 0.25 | 1.15 | 0.45 | 0.34 | 0.20 | 0.68 | 92.9 | 97.5 | 99.2 | 83.3 |
| 75 | MLE | 100 | 0.95 | 0.53 | 0.28 | 1.21 | 0.63 | 0.33 | 0.20 | 0.80 | 72.6 | 92.3 | 99.8 | 68.5 |
| | BR | 100 | 0.48 | 0.36 | 0.20 | 0.51 | 0.30 | 0.23 | 0.13 | 0.34 | 93.0 | 97.5 | 100 | 94.0 |
| | MBR | 100 | 0.52 | 0.37 | 0.21 | 0.52 | 0.33 | 0.24 | 0.13 | 0.35 | 92.4 | 97.7 | 100 | 94.4 |
| | CL | 100 | 0.51 | 0.37 | 0.22 | 0.51 | 0.32 | 0.24 | 0.13 | 0.33 | 68.2 | 93.9 | 100 | 100 |
| | RE | 100 | 0.49 | 0.35 | 0.20 | 0.73 | 0.31 | 0.23 | 0.13 | 0.45 | 90.3 | 95.1 | 97.7 | 84.7 |
| 150 | MLE | 100 | 0.71 | 0.38 | 0.21 | 1.00 | 0.52 | 0.26 | 0.14 | 0.79 | 63.6 | 90.4 | 99.6 | 49.7 |
| | BR | 100 | 0.33 | 0.25 | 0.15 | 0.35 | 0.22 | 0.17 | 0.09 | 0.24 | 92.8 | 97.7 | 100 | 93.7 |
| | MBR | 100 | 0.36 | 0.25 | 0.15 | 0.36 | 0.23 | 0.17 | 0.10 | 0.24 | 92.1 | 97.7 | 100 | 94.4 |
| | CL | 100 | 0.33 | 0.25 | 0.16 | 0.35 | 0.22 | 0.17 | 0.10 | 0.23 | 81.2 | 99.1 | 100 | 100 |
| | RE | 100 | 0.35 | 0.24 | 0.14 | 0.41 | 0.23 | 0.16 | 0.09 | 0.26 | 88.6 | 94.5 | 97.3 | 87.9 |

Tabella 4.1: Risultati delle simulazioni per $I = 5$ e $S = 20, 40, 75, 150$, confronto tramite gli indici RMSE, MAE e $CI_{0.95}$.

Un secondo studio prende in esame un modello con $I = 20$ item, dove $\alpha \sim N(0, 2)$. Si distinguono tre casi: (i) $S = 20$ e $\gamma_s \in (-5, 5)$; (ii) $S = 20$ e $\gamma_s \sim \text{Mistura}$; (iii) $S = 100$ e $\gamma_s \sim T_4$. La mistura è l'unione equipesata di $\gamma \sim N(0, 2)$ e $\gamma = \pm 100$ allo scopo di rappresetare un campione comprendente sia soggetti nella norma sia soggetti che eccellono da una parte e dall'altra, contesto che non sarebbe stimabile con modelli ad effetti casuali in cui l'assunzione alla base è la Normalità dei parametri γ . I risultati di questo studio si presentano in Figura 4.1, dove sono riportate le stime per le abilità dei soggetti nei tre modelli solo per il caso (i) con $S = I$, ed in Figura 4.2, dove sono riportate le distribuzioni empiriche simulate degli errori di stima per gli stimatori di α nei tre casi.

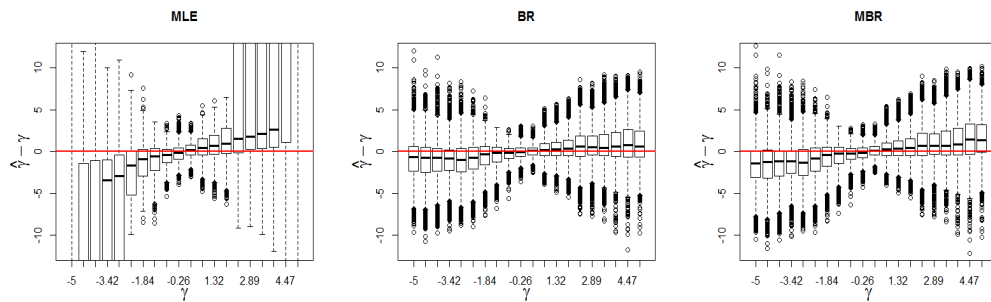


Figura 4.1: Errori di stima di γ per i tre stimatori MLE, BR, MBR con $I = 20$ e $S = 20$.

In generale la Tabella 4.1 e le Figure 4.1 e 4.2 mostrano che le stime di massima verosimiglianza hanno proprietà distributive peggiori fra i cinque modelli proposti; lo stimatore non distorto in media ha indicatori in linea con il modello condizionato e con quello ad effetti casuali ma ha il vantaggio di stimare contemporaneamente anche i parametri γ . Infine lo stimatore non distorto in mediana ha proprietà di poco peggiori rispetto allo stimatore non distorto in media. Le relative stime prodotte sono sempre finite e si posizionano fra le stime dello stimatore di Firth (1993) e quelle di massima verosimiglianza, questo risultato è dovuto ad un minor effetto di *shrinkage*, come spiegato nel Capitolo 1.

A conclusione si confrontano i cinque metodi di stima presentati in termini di tempi computazionali necessari alla stima per le relative funzioni R: la Tabella 4.2 mostra i risultati, in secondi, ottenuti mediando i tempi di ogni modello simulato. Si nota che la funzione `glm` è la più efficiente, la funzione `mbrRasch.lpl` risulta migliore di `brglm` (Kosmidis, 2013) perchè quest'ultima non è nella versione ottimizzata per il modello di Rasch. Per la stima di massima verosimiglianza condizionata e la stima con effetti casuali si sono utilizzate rispettivamente la funzione `clogit` del pacchetto `survival` di Therneau (2016) ed la funzione `glmer` del pacchetto `lme4` di Bates (2015). Le due funzioni risultano le più onerose in termini di tempo.

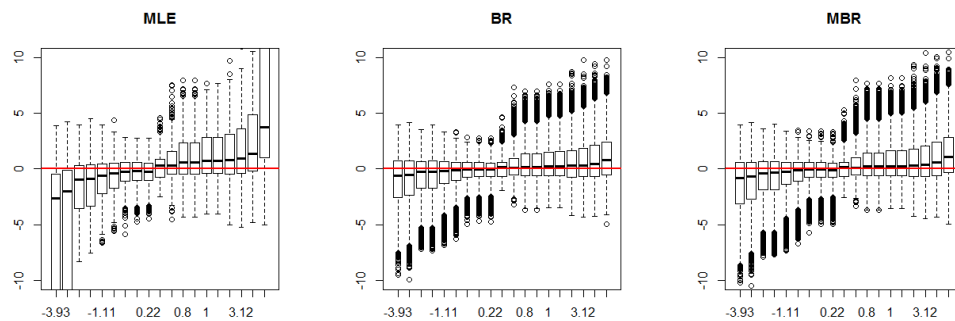
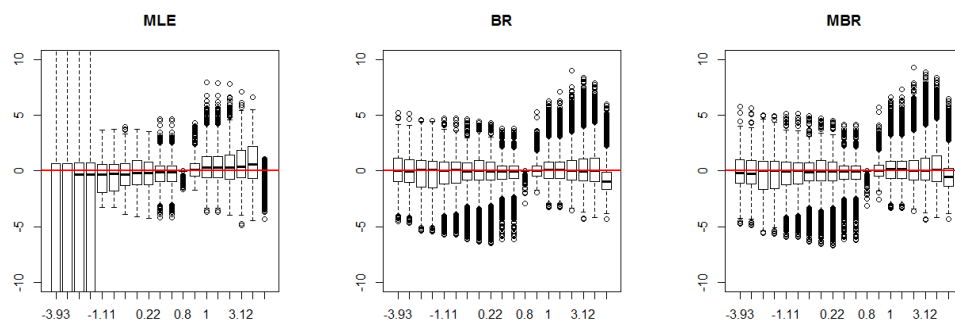
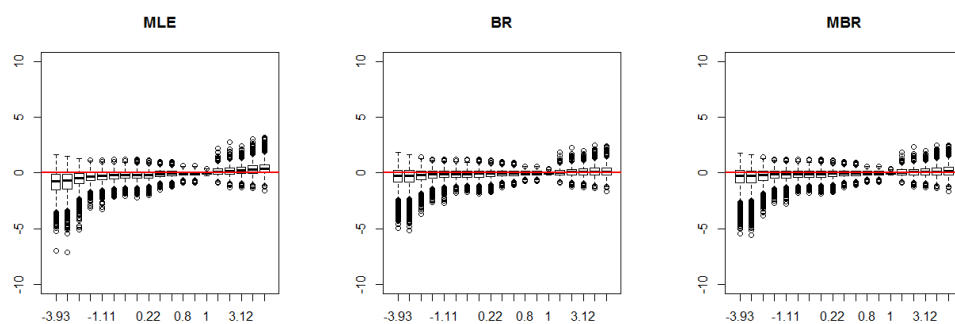
(a) Errori di stima di α nel caso (i)(b) Errori di stima di α nel caso (ii)(c) Errori di stima di α nel caso (iii)

Figura 4.2: Errori di stima di α per i tre stimatori MLE, BR, MBR nei tre casi con $I = 20$ e $S = 20, 20, 100$.

| I | S | dim(θ) | MLE | BR | MBR | CL | RE |
|----|-----|-----------------|------|------|------|--------|-------|
| 5 | 20 | 24 | 0.01 | 0.04 | 0.01 | 0.13 | 0.47 |
| | 50 | 54 | 0.02 | 0.12 | 0.04 | 5.57 | 0.75 |
| | 100 | 104 | 0.12 | 0.57 | 0.22 | 89.15 | 1.10 |
| 20 | 20 | 39 | 0.03 | 0.14 | 0.06 | 0.53 | 12.64 |
| | 50 | 69 | 0.12 | 0.54 | 0.26 | 19.30 | 31.70 |
| | 100 | 119 | 0.57 | 2.63 | 1.28 | 401.25 | 46.76 |

Tabella 4.2: Tempi computazionali in secondi delle cinque funzioni.

4.2 Simulazioni per il modello logistico a due parametri

Il modello di Rasch a due parametri ad effetti fissi per i soggetti è stimabile solamente tramite lo stimatore non distorto in media e lo stimatore non distorto in mediana. Per la stima del modello ad effetti casuali si possono consultare i pacchetti `mirt` di Chalmers (2016) o anche `nlme` di Pinhero (2016), ma in questo studio ci si limita al confronto su modelli ad effetti fissi. Ci si riferisce agli indici d'errore presentati nel Paragrafo 4.1, con l'aggiunta della probabilità di sottostima del vero valore del parametro (PU %). Non è possibile, invece, calcolare la copertura empirica degli intervalli di confidenza perchè la funzione `brRasch` di Kosmidis (2016), a differenza della funzione `mbrRasch.2pl`, non fornisce nella versione attuale gli *Standard Error*. In questo paragrafo le simulazioni sono svolte su un campione di $R = 2000$ prove.

Nelle Tabelle 4.3 e 4.4 si riportano le statistiche d'errore per gli stimatori di α e β nei casi: (i) $S = 20$ e $\gamma \sim T_4$; (ii) $S = 60$ e $\gamma \sim \text{Mistura}$ e (iii) $S = 100$ e $\gamma \sim \text{Mistura}$, dove la mistura è definita nel Paragrafo 4.1. Nelle simulazioni in cui il modello non distorto in media non ha raggiunto la convergenza le rispettive stime sono state escluse, quindi gli indici sono da interpretare con cautela; il modello non distorto in mediana non ha invece presentato problemi di stima evidenti. Analogamente al caso precedente il nuovo stimatore presenta maggior dispersione rispetto allo stimatore di riferimento, inoltre entrambi i modelli incontrano difficoltà nella stima dei parametri β , probabilmente dovute alla non linearità associata. Il risultato conferma l'influenza che ha l'alto numero di parametri incidentali γ sulla

stima dei parametri α e β relativi item, infatti all'aumentare di S aumenta anche la variabilità delle stime dell'intero vettore dei parametri.

| I | S | | Stime finite (%) | RMSE | | | | MAE | | | | PU% | | | |
|---|-----|-----|------------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
| | | | | α_2 | α_3 | α_4 | α_5 | α_2 | α_3 | α_4 | α_5 | α_2 | α_3 | α_4 | α_5 |
| 5 | 20 | BR | 100 | 0.71 | 0.70 | 0.72 | 0.91 | 0.50 | 0.46 | 0.50 | 0.67 | 0.44 | 0.46 | 0.61 | 0.78 |
| | | MBR | 100 | 1.26 | 1.04 | 1.09 | 1.30 | 0.61 | 0.56 | 0.60 | 0.82 | 0.56 | 0.49 | 0.53 | 0.57 |
| 5 | 60 | BR | 99.6 | 0.77 | 0.57 | 0.58 | 0.83 | 0.50 | 0.36 | 0.38 | 0.66 | 0.39 | 0.42 | 0.65 | 0.82 |
| | | MBR | 100 | 1.07 | 0.70 | 0.71 | 0.87 | 0.53 | 0.47 | 0.47 | 0.60 | 0.54 | 0.48 | 0.55 | 0.66 |
| 5 | 100 | BR | 100 | 0.77 | 0.47 | 0.50 | 0.92 | 0.43 | 0.31 | 0.34 | 0.78 | 0.56 | 0.40 | 0.66 | 0.92 |
| | | MBR | 100 | 0.97 | 0.55 | 0.60 | 0.85 | 0.51 | 0.36 | 0.40 | 0.63 | 0.69 | 0.47 | 0.57 | 0.80 |

Tabella 4.3: Risultati delle simulazioni della stima di α per $I = 5$ e $S = 20, 60, 100$, confronto tramite gli indici RMSE, MAE e PU%.

| I | S | | Stime finite (%) | RMSE | | | | MAE | | | | PU% | | | |
|---|-----|-----|------------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| | | | | β_2 | β_3 | β_4 | β_5 | β_2 | β_3 | β_4 | β_5 | β_2 | β_3 | β_4 | β_5 |
| 5 | 20 | BR | 100 | 0.42 | 0.30 | 0.40 | 0.61 | 0.28 | 0.19 | 0.34 | 0.58 | 0.39 | 0.53 | 0.88 | 0.97 |
| | | MBR | 100 | 0.81 | 0.72 | 0.67 | 0.74 | 0.36 | 0.31 | 0.36 | 0.47 | 0.32 | 0.42 | 0.59 | 0.68 |
| 5 | 60 | BR | 99.6 | 0.52 | 0.18 | 0.23 | 0.45 | 0.47 | 0.14 | 0.22 | 0.45 | 0.00 | 0.08 | 0.97 | 1.00 |
| | | MBR | 100 | 0.56 | 0.24 | 0.21 | 0.40 | 0.48 | 0.17 | 0.17 | 0.38 | 0.00 | 0.10 | 0.81 | 0.97 |
| 5 | 100 | BR | 100 | 0.57 | 0.19 | 0.77 | 1.37 | 0.51 | 0.17 | 0.76 | 1.38 | 0.00 | 0.96 | 1.00 | 1.00 |
| | | MBR | 100 | 0.55 | 0.19 | 0.70 | 1.31 | 0.48 | 0.15 | 0.70 | 1.31 | 0.00 | 0.86 | 1.00 | 1.00 |

Tabella 4.4: Risultati delle simulazioni della stima di β per $I = 5$ e $S = 20, 60, 100$, confronto tramite gli indici RMSE, MAE e PU%.

Relativamente al caso (i) in cui $I = 5$ e $S = 20$ si riportano in Figura 4.3 anche i grafici delle distribuzioni empiriche simulate degli errori di stima per θ , dove i dati sono generati dai parametri $\gamma \sim T_4$, $\alpha = (0, -1.5, -0.5, 0.5, 1.5)$, $\beta = (1, 0.5, 0.8, 1.2, 1.5)$ riportati in ascissa. Dalla figura emerge che la centratura in mediana è raggiunta componente per componente dallo stimatore proposto già per campioni piccoli.

Infine, in Figura 4.4 si riportano i risultati di simulazione relativi alla stima dei parametri degli item (α, β) nel caso in cui $I = 20$ e $S = 60$ e $\gamma \sim \text{Mistura } \alpha \in (-2, 2)$, $\beta \in (0.5, 2.5)$. I grafici confermano che i due stimatori hanno indici di errore con ordini di grandezza comparabili, anche per campioni di dimensioni maggiori. Dalle analisi si sono esclusi tre campioni per i quali la funzione `mbrRasch.2pl` non ha raggiunto la convergenza.

Per completezza rispetto alla Tabella 4.2, si riporta in Tabella 4.5 un confronto dei tempi computazionali fra le funzioni `brRasch` e `mbrRasch.2pl`. In questo caso entrambe le funzioni sono ottimizzate per il modello logistico

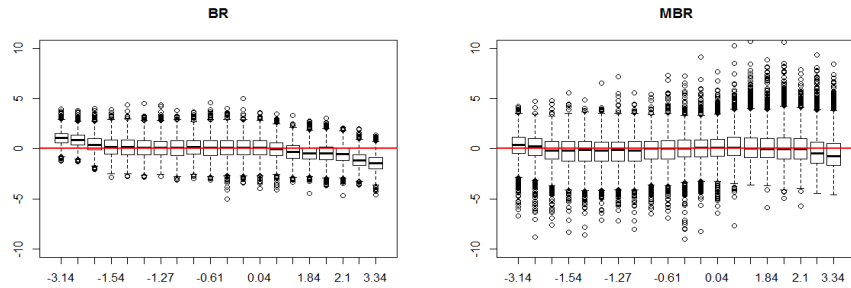
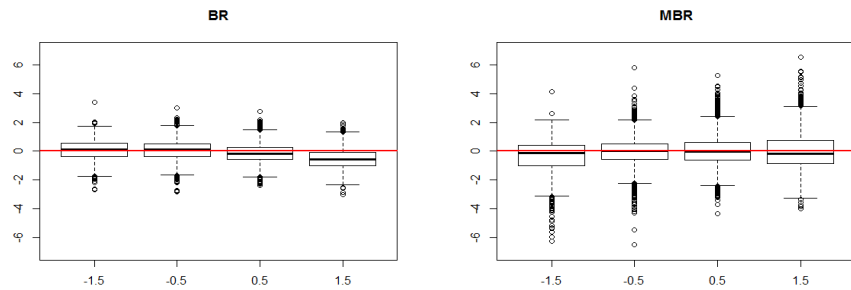
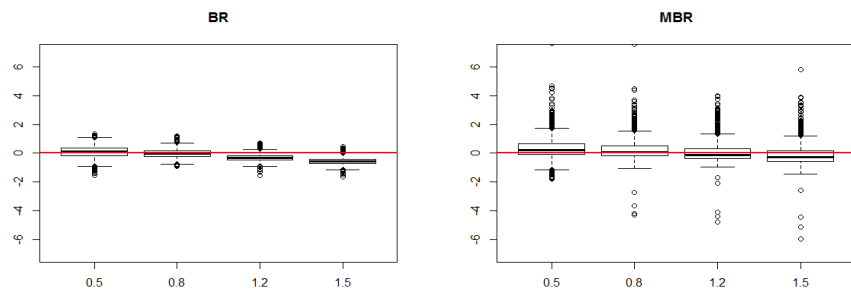
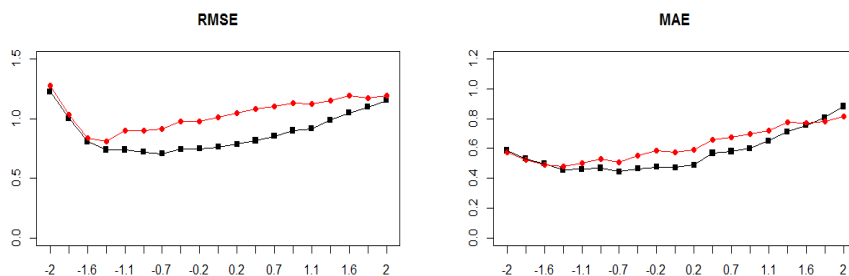
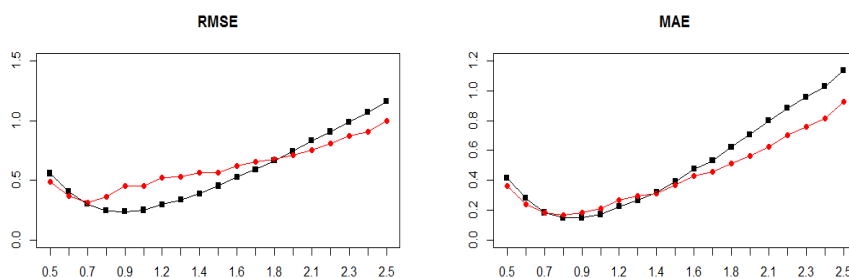
(a) Errore di stima di γ (b) Errore di stima di α (c) Errore di stima di β 

Figura 4.3: Errori di stima di (γ, α, β) per i due stimatori BR e MBR con $I = 5$ e $S = 20$.

a due parametri, ma la seconda risulta comunque più efficiente grazie alla parte implementata in C++.

(a) Stima di α (b) Stima di β 

—■— BR —●— MBR

Figura 4.4: Stimatori BR e MBR di (α, β) con $I = 20$ e $S = 60$, confronto con indici RMSE e MAE.

| I | S | dim(θ) | BR | MBR | Rapporto |
|----|-----|-----------------|-------|-------|----------|
| 5 | 20 | 28 | 5.48 | 0.23 | 23.8 |
| | 50 | 58 | 8.04 | 0.73 | 11.0 |
| | 100 | 108 | 9.09 | 3.88 | 2.3 |
| 20 | 20 | 58 | 28.51 | 1.09 | 26.2 |
| | 50 | 88 | 32.78 | 4.38 | 7.5 |
| | 100 | 138 | 44.98 | 17.45 | 2.6 |

Tabella 4.5: Tempi computazionali in secondi delle due funzioni.

Capitolo 5

Applicazione ai dati dei test d'ingresso ai corsi di laurea triennale del Dipartimento di Scienze Statistiche

“The ability to quote is a serviceable substitute for wit”

S. Maugham

Si analizzano i dati relativi ad un questionario di comprensione linguistica all'interno del test d'ingresso per il Dipartimento di Scienze Statistiche. Sono state considerate quattro sessioni del test, dove, a fronte di una serie di dieci domande uguali, il testo scritto di riferimento varia per ognuna delle sessioni. Si vuole quindi procedere all'analisi dei questionari con lo scopo di verificare se le quattro sessioni del test d'ingresso presentano testi di difficoltà equiparabile. I dati sono costituiti da quattro tabelle, una per sessione, a due entrate, una per i candidati ed una per le domande. Si analizzano le quattro sessioni sia separatamente, sia unendole a formare un unico campione rappresentativo del test d'ingresso.

La Tabella 5.1 mostra in forma sintetica le principali caratteristiche relative ai dieci item. Si nota che vi sono differenze fra le probabilità di rispondere in modo corretto ai quesiti relativi ai diversi testi, si veda per esempio l'item 2. Nell'item 9 vi è un caso di perfetta separazione e ci sono problemi di quasi perfetta separazione sia nell'item 1 che nel 3. Nella tabella 5.2 sono

riportate le analoghe statistiche descrittive per i soggetti, dove emerge che ci son diversi soggetti che hanno risposto correttamente a tutti gli item o, al contrario, a nessun item. In tutti questi casi evidenziati non esiste la stima di massima verosimiglianza per i rispettivi parametri.

| | Num. Sogg. | Percentuale di risposte giuste per domanda | | | | | | | | | |
|---------|---------------|--|------|------|------|------|------|------|------|-------|------|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Testo 1 | 63 | 77.8 | 47.6 | 88.9 | 84.1 | 57.1 | 76.2 | 58.7 | 42.9 | 84.1 | 87.3 |
| Testo 2 | 54 | 90.7 | 81.5 | 98.1 | 87.0 | 96.3 | 70.4 | 96.3 | 61.1 | 46.3 | 33.3 |
| Testo 3 | 60 | 98.3 | 73.3 | 96.7 | 33.3 | 95.0 | 88.3 | 71.7 | 85.0 | 100.0 | 88.3 |
| Testo 4 | 61 | 67.2 | 1.6 | 83.6 | 95.1 | 90.2 | 91.8 | 78.7 | 95.1 | 86.9 | 93.4 |
| Totale | 238 | 83.2 | 50.0 | 91.6 | 74.8 | 84.0 | 81.9 | 75.6 | 71.0 | 80.3 | 76.9 |

Tabella 5.1: Statistiche descrittive relative agli item.

| Soggetti che hanno risposto | Num. Item | Percentuale di soggetti per testo | | | | Totale |
|--------------------------------|--------------|-----------------------------------|---------|---------|---------|--------|
| | | Testo 1 | Testo 2 | Testo 3 | Testo 4 | |
| sempre corretto | 10 | 12.7 | 11.1 | 16.7 | 0.0 | 10.3 |
| sempre sbagliato | 10 | 3.2 | 0.0 | 0.0 | 0.0 | 0.8 |

Tabella 5.2: Statistiche descrittive relative ai soggetti.

Si stimano cinque diversi modelli di Rasch nella versione a due parametri, quattro per le sessioni ed uno per il campione unificato composto dai $S = 238$ studenti. Si prende come riferimento per il vincolo di identificabilità l'item 10 perché la scelta dell'item 1 porta a problemi di stabilità delle stime, in particolare per il Testo 3, probabilmente dovuti alla quasi perfetta separabilità delle risposte; si pone quindi $\alpha_{10} = 0$ e $\beta_{10} = 1$. La stima dei quattro modelli ha raggiunto la convergenza, con una tolleranza pari a 10^{-3} , fra le 20 e le 100 iterazioni in meno di 5 secondi per ogni modello; per il quinto modello invece sono occorse più di 200 iterazioni per un tempo totale pari a 2 minuti circa. Tutte le stime sono finite e presentano *Standard Error* con un ordine di grandezza coerente.

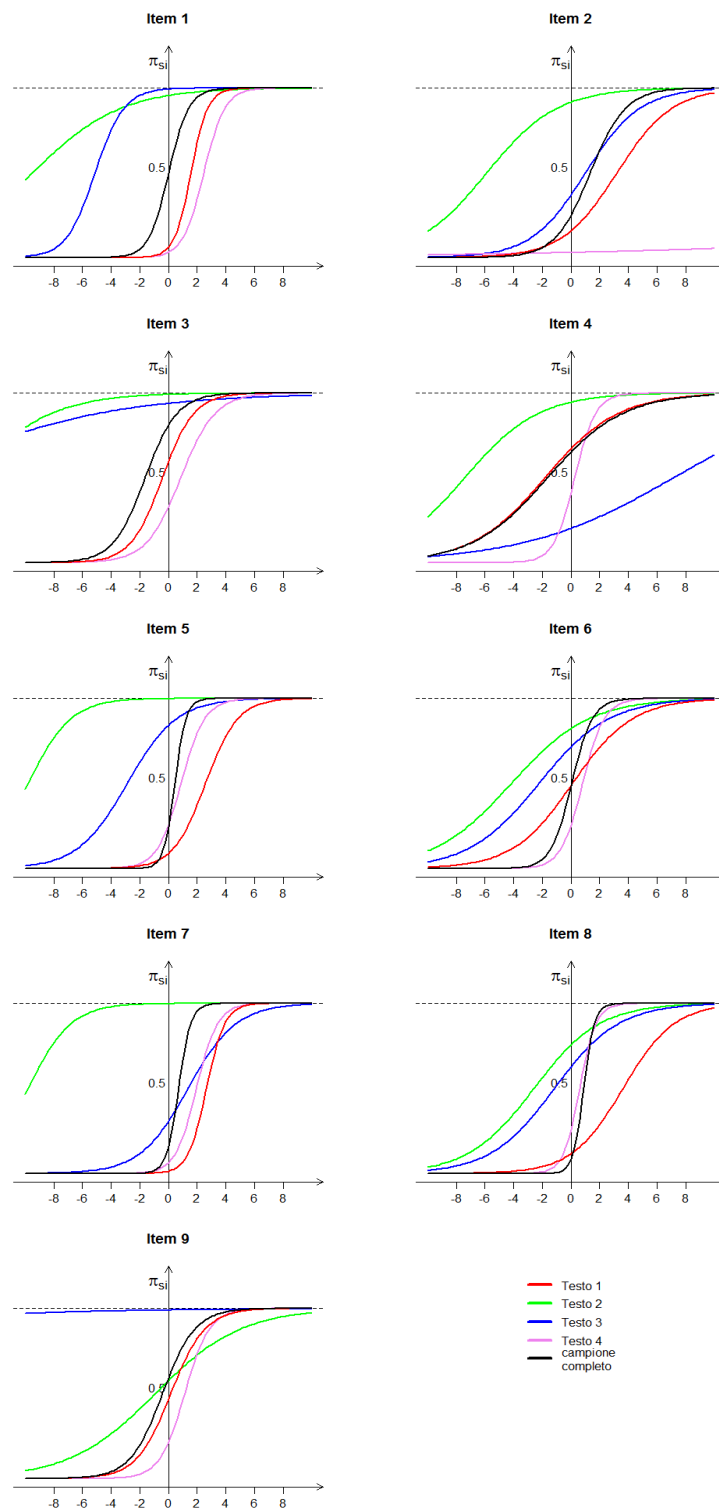


Figura 5.1: Curve caratteristiche dei dieci item divisi per le quattro sessioni ed il campione unificato.

Si presenta in Figura 5.1 la stima delle curve caratteristiche dei dieci item relative ai cinque modelli. Dall'analisi grafica si possono già trarre conclusioni circa la difficoltà dei testi proposti. Si nota ad esempio che i Testi 1 e 4 sono considerabili in generale più difficili del Testo 2, in quanto le relative curve sono sempre più "a destra" delle curve verdi. L'item 4 del testo 3 ha una bassa percentuale di risposte giuste (Tabella 5.1) e la rispettiva curva associa, a parità di bravura del candidato, probabilità di successo minore rispetto agli altri tre testi. Infine l'item 5 è quello che meglio discrimina la bravura dei candidati nel campione unificato degli studenti, in quanto presenta la curva con maggior inclinazione.

Si presenta anche il calcolo degli *Standard Error* degli stimatori $\tilde{\alpha}_i$ e $\tilde{\beta}_i$ attraverso la procedura *jackknife* sul campione unificato per fare una verifica di robustezza sul modello. Per ogni item si sono calcolate 238 coppie di parametri $(\tilde{\alpha}_i^{(s)}, \tilde{\beta}_i^{(s)})$ ed il risultato è riportato in Figura 5.2. Tutti i grafici sono stati centrati sui valori nulli $(\alpha, \beta) = (0, 1)$ per migliorare il confronto diretto. Gli *Standard Error* appena calcolati portano a conclusioni sulla significatività dei parametri concordi con quelle ottenute a partire dagli valori teorici e fornite dalla funzione `mbrRasch.2pl`; inoltre emerge una netta correlazione negativa fra gli stimatori $\tilde{\alpha}_i$ e $\tilde{\beta}_i$ sia dai grafici sia dalle corrispettive quantità teoriche $i(\tilde{\alpha}, \tilde{\beta})^{-1}$; i valori negativi di correlazione sono in parte dovuti alla stessa dinamica che porta in un modello lineare univariato gli stimatori dei parametri di posizione ed inclinazione ad essere correlati negativamente. Questo studio nasce con scopo esplorativo dei dati per mostrare il funzionamento della procedura di stima, tuttavia si potrebbe procedere con dei test formali per verificare l'ipotesi di uguaglianza fra i parametri $\alpha_1, \dots, \alpha_9$ o fra i β_1, \dots, β_9 con un test di Wald $W = (\tilde{\theta} - \theta)^\top i(\theta)(\tilde{\theta} - \theta)$ la cui distribuzione nulla asintotica è χ_{df}^2 , dove i gradi di libertà (*df*) dipendono dal numero di vincoli imposti. Si sottolinea che un test di uguaglianza fra i β equivale ad un test per la scelta fra il modello logistico ad un parametro, ipotesi nulla, o a due, ipotesi alternativa. Non è invece possibile eseguire un test per la verifica di uguaglianza fra tutti i parametri α o β in quanto sono relativi a campioni diversi.

A titolo esemplificativo si presenta anche l'analisi sulla abilità degli studenti che hanno partecipato al test. Il sistema che gestisce i risultati del test

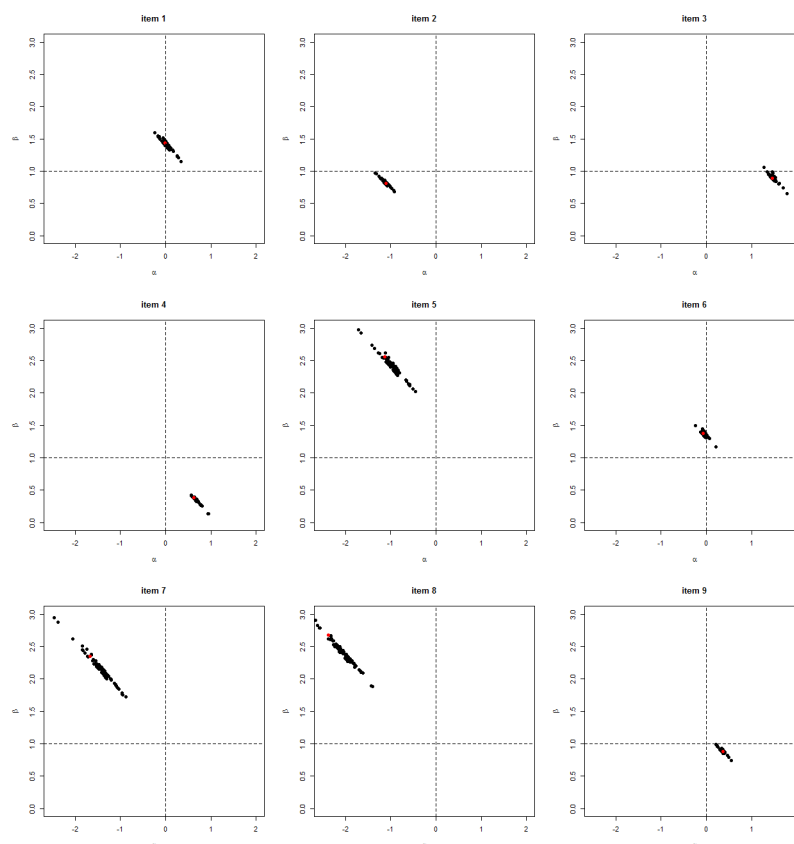


Figura 5.2: Stime jackknife di (α, β) relative ai dieci item.

di ingresso è progettato per non fare discriminazioni in base alla sessione, ma per considerare tutti i soggetti in un unico campione. Seguendo questa logica si è considerato il campione dei 238 studenti e si è stimato un modello logistico a due parametri, dal quale poi sono stati considerati solo le stime $\tilde{\gamma}$. Nonostante vi siano solo dieci domande, i parametri γ_s vengono ben discriminati pesando le risposte del candidato per le caratteristiche delle domande affrontate; in altri termini a parità di percentuale di risposte esatte è migliore il candidato che ha risposto giusto a domande più difficili. Sul totale dei 238 parametri stimati per i soggetti, 137 sono significativi e, fra questi, 28 hanno il punteggio massimo stimato e sono quindi classificati come i migliori del campione considerato. In fase interpretativa dei risultati si può verificare a quali sessioni hanno partecipato gli studenti più meritevoli. In Figura 5.3 si riporta l'istogramma delle stime in cui ogni barra è colorata secondo la ses-

sione d'appartenenza della maggioranza degli studenti che hanno associato un valore di abilità dentro il corrispettivo range. Ad esempio fra i migliori 28 studenti, 23 hanno partecipato alla sessione 4, mentre fra i valori più bassi di γ si trova una maggioranza di studenti sottoposti alla sessione 3.

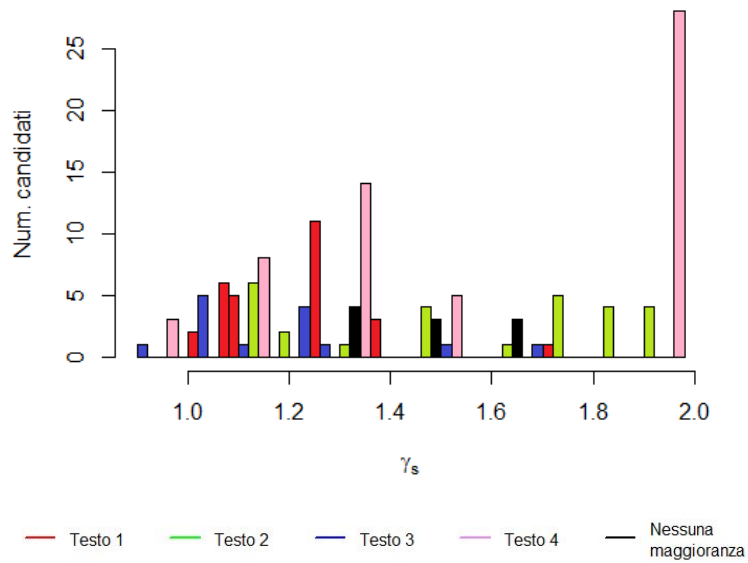


Figura 5.3: Istogramma dei parametri γ_s evidenziato per sessione d'appartenenza degli studenti.

Conclusioni

Questo lavoro si è posto come obiettivo primario la stima non distorta in mediana del modello di Rasch in due specifiche diverse, che costituisce l'elemento di novità della tesi.

Il modello di Rasch è riconducibile al caso particolare di una regressione logistica per l'analisi di dati generati da test di valutazione. La probabilità di rispondere in modo corretto ad una domanda, o item, da parte di un candidato è in funzione sia della difficoltà della domanda sia della bravura del rispondente. Per introdurre il problema di ricerca si è presentato inizialmente il modello di Rasch nelle specifiche ad uno o due parametri e si sono esaminati i principali metodi di stima del modello presenti in letteratura, portando alla luce i limiti di ogni approccio. Si è concluso che la soluzione migliore sotto diversi aspetti sia lo stimatore non distorto in media di Firth (1993), unico ad essere estendibile alla seconda specifica; esso, quindi, assume il ruolo di *benchmark* nelle analisi. Si è inoltre introdotto lo stimatore non distorto in mediana proposto da Kenne Pagui *et al.* (2016), che raggiunge le proprietà desiderate tramite una modificazione della *score function* e rispetta l'invarianza per trasformazione che operano sugli elementi di θ in modo separato.

L'obiettivo è stato raggiunto in due fasi fortemente legate fra loro: da una parte si sono sviluppate le formule per il calcolo esplicito dei cumulanti necessari per la modificazione e si è proposta l'approssimazione dell'algoritmo di Newton-Raphson per la stima in via numerica del modello; dall'altra si sono scritte due funzioni nel linguaggio R capaci di ricevere in input i dati e di fornire in output le stime e relativi *standard error* in tempi computazionali ridotti, sfruttando le potenzialità dell'interazione fra R e C++. Particolare

attenzione è stata posta sulla specifica a due parametri, dove si è dovuto modificare l'algoritmo di stima per garantire maggior stabilità nella funzione in R. Il risultato di questa parte dell'elaborato è racchiuso nelle due funzioni `mbrRasch.1pl` e `mbrRasch.2pl` riportate in Appendice B.

L'obiettivo secondario della tesi risiede nel confronto fra lo stimatore non distorto in mediana e gli altri metodi introdotti per la stima del modello di Rasch; nel Capitolo 4 si presentano gli studi di simulazione che sono stati proposti a tale fine. Per il modello logistico ad un parametro si sono considerate le stime: di massima verosimiglianza, non distorta in media, di massima verosimiglianza condizionata e con effetti casuali per i soggetti. Ne risulta che la stima di massima verosimiglianza porta a risultati inferenziali non affidabili a causa del problema di Neyman Scott sui modelli con un elevato numero di parametri incidentali e di casi in cui le stime non esistono finite; la stima di massima verosimiglianza condizionata e quella con effetti casuali risultano efficienti per la stima dei parametri relativi agli item, ma hanno come principale limite quello di non poter stimare congiuntamente anche i parametri ad effetti fissi per i soggetti. Infine lo stimatore non distorto in media è l'unico che stima congiuntamente l'intero vettore dei parametri del modello e garantisce l'esistenza delle stime; i vari indici di errore, quali il *root mean square error*, il *median absolute error* e la copertura empirica degli intervalli di confidenza, lo indicano come la miglior scelta in letteratura.

Il nuovo stimatore non distorto in mediana è, al pari di quello non distorto in media, in grado di stimare l'intero vettore dei parametri e ha sempre mostrato stime finite. Gli indici di errore indicano proprietà distributive simili allo stimatore di Firth, seppur con maggiore dispersione. Questo risultato è dovuto al fatto che nel caso della regressione logistica la modificazione per la non distorsione in mediana comporta un minor effetto di *shrinkage* dei parametri rispetto a quella per la non distorsione in media.

La seconda specifica del modello di Rasch porta maggiormente alla luce i vantaggi del nuovo metodo di stima. Esso si configura come alternativa alla stima non distorta in media, unico metodo fin ora implementabile; rispetto a quest'ultimo presenta indici di errore di uguale entità e riduce la percentuale di non convergenza dell'algoritmo sotto l'uno per mille. Infine, la funzione `mbrRasch.2pl` è, a seconda delle dimensioni del campione, dalle due alle venti

volte più veloce rispetto a quella ottimizzata per la stima non distorta in media.

Per concludere si è mostrato un caso di applicazione del metodo di stima a dei dati relativi ad un questionario di comprensione linguistica all'interno del test di ingresso per il Dipartimento di Scienze Statistiche dell'Università di Padova. Hanno partecipato in totale 238 studenti divisi in quattro sessioni del test, composto da dieci domande uguali fra loro ma riferite ad un testo diverso per ogni sessione. L'obiettivo dell'analisi è quello di verificare se i quattro test proposti sono di pari livello di difficoltà, con l'intento di garantire la massima uniformità. Sono stati stimati quattro modelli per le sessioni che hanno evidenziato la non omogeneità fra le prove, in particolare la prima sessione è risultata la più difficile, mentre la seconda la più facile.

Appendice A

Rcpp

A.1 Introduzione a Rcpp

R ha una doppia natura, da una parte è un ambiente per l'analisi, la visualizzazione e la modellizzazione dei dati, dall'altra è un linguaggio di programmazione costruito per soddisfare queste richieste. Questa sua caratteristica è unica nell'insieme dei linguaggi ed è ben descritta dal titolo della guida per il linguaggio S di Chambers (1998). La flessibilità che caratterizza R è data dall'essere un linguaggio interpretato ed è ottenuta a costo di un rallentamento nella fase di esecuzione di algoritmi. Con l'aumentare della complessità degli algoritmi ed il bisogno di elaborare moli di dati sempre crescenti si è guardato a soluzioni complementari ad R che integrassero linguaggi di tipo compilato.

Vi sono diversi punti di contatto fra R e C++ che configurano quest'ultimo come una naturale soluzione al problema. L'interprete di R e tutta la sua implementazione interna sono scritti in C, poichè è facile, veloce e supportato dalle principali piattaforme hardware. Un principale vantaggio di C è la possibilità di estenderlo tramite moduli e librerie esterne, che come si vedrà è un elemento chiave per lo sviluppo di Rcpp. A sua volta C è strettamente legato a C++ che si può vedere come un'estensione ed arricchimento. In secondo luogo sia R (nelle classi S3 e S4) sia C++ sono programmi orientati ad oggetti (Bates, 2001).

Per questi motivi l'interazione di R con C++ è stato un problema posto

da molti autori e la prima pubblicazione sull'argomento risale al 2001 (Bates e DebRoy, 2001), il manuale del R Development Core Team per la scrittura di estensioni di R (R Development Core Team, 2016b) affronta lo stesso tema in quegli anni. Nel 2005 nasce il pacchetto **Rcpp** ad opera di Samperi che si fonda sul pacchetto di **RQuantLib** di Eddelbuettel del 2002 (Eddelbuettel, 2016b). Dopo un rapido avvio e un'altrettanta brusca frenata il pacchetto è rivitalizzato dallo stesso Eddelbuettel nel 2008 ed aggiornato nel 2009 per seguire l'evoluzione del linguaggio C++. Quest'ultima versione entra nel CRAN e viene periodicamente aggiornata dagli autori (Eddelbuettel, 2016a). Già nel 2012 più di novanta pacchetti fanno utilizzo di Rcpp, ad oggi (Aggiornamento sett 2016) sono più di settecento. L'attuale versione è la 12.7 mantenuta da Eddelbuettel (2016a).

Ci sono vari altri pacchetti che hanno finalità analoghe, il primo in ordine cronologico è **Rserve** nato nel 2003 ad opera di Urbanek (2003, 2012) che si configura come un server TCP/IP per facilitare l'uso di R ad altri linguaggi di programmazione (soprattutto per servizi web). Esistono anche i pacchetti **rcppbind** di Liang (2008), **RObjects** e **Rabstaction** di Armstrong (2009 e 2014), ed **rdyncall** di Adler (2012) non rilasciati ufficialmente dal CRAN. Si segnala inoltre il progetto **CXXR** di Runnals (2009) che vuole gradualmente ristrutturare l'interprete di R, attualmente scritto in C, in C++.

A.2 Un primo esempio

Per meglio comprendere l'utilizzo di Rcpp e le sue potenzialità si presenta un semplice esempio, tratto da Eddelbuettel, 2013. Si vuole scrivere una funzione per calcolare la sequenza di Fibonacci F_n definita dalla formula ricorsiva $F_n = F_{n-1} + F_{n-2}$ e dalle condizioni iniziali $F_0 = 0$ e $F_1 = 1$. Una prima formulazione in R può essere:

```
fibR <- function(n){  
  if(n==0) return(0)  
  if(n==1) return(1)  
4  return(fibR(n-1)+fibR(n-2))  
}
```

Questo script è breve e facile da comprendere, ma terribilmente inefficiente. Un'alternativa possibile è trasportare lo stesso script in C++ e richiamarlo in R attraverso un *wrapper*. Repp propone due funzioni che svolgono la doppia funzione di convertire gli oggetti da R in C++ e viceversa: `as` e `wrap`.

```
int fibonacci(const int x){
    if (x == 0) return(0);
    if (x == 1) return(1);
    return (fibonacci(x-1) + fibonacci(x-2));
5 }
```

```
extern "C" SEXP fibWrapper(SEXP xs) {
    int x = Rcpp::as<int>(xs);
    int fib = fibonacci(x);
    return (Rcpp::wrap(fib));
5 }
```

Il codice in C++ presentato risulta altrettanto facile da leggere almeno nella sua prima parte dove segue la stessa logica della funzione in R. Nella seconda metà si opera affinché il codice richiamabile da R tramite la funzione `.Call()`. Il comando `as` converte l'argomento in input `xs` dal tipo `SEXP`, un puntatore ad un'espressione in S, in un intero. Analogamente `wrap` converte l'output di tipo `integer` della variabile `fib` nel tipo `SEXP`. I successivi passaggi sono la compilazione di queste due funzioni, il collegamento in una libreria condivisa e il loro effettivo lancio. Per facilitare questi passaggi è consigliabile l'utilizzo del pacchetto `inline` (Sklyar *et al.*, 2012) che attraverso la funzione `cxxfunction` fornisce un wrapper che copre questi tre passaggi e le relative specifiche interne. Tramite questi strumenti si può giungere ad una soluzione scritta interamente in R dove il codice C++ entra come semplice elemento testuale.

```
incltxt <- '
int fibonacci(const int x) {
    if (x == 0) return(0);
    if (x == 1) return(1);
5   return fibonacci(x - 1) + fibonacci(x - 2);
}'

fibRcpp <- cxxfunction(signature(xs="int"),
                        plugin = "Rcpp",
```

```

10     incl = incltxt,
        body = '
        int x = Rcpp::as<int>(xs);
        return Rcpp::wrap( fibonacci(x) );
        ,
15     )

```

L'utilizzo di `cxxfunction` è standardizzato ed il programmatore deve solo preoccuparsi di aggiornare i parametri a seconda del caso. Il puro codice C++ entra nell'argomento `includes` che permette anche l'inserimento di altre direttive come la definizione di funzioni ausiliari o classi. Il corpo centrale della funzione (il *wrapper*) entra nell'argomento `body` che gestisce il passaggio delle variabili fra R e C++ come spiegato in precedenza. Si definisce l'argomento `signature` per la scrittura della stessa in C++, dove si dichiarano le variabili input ed, infine, si specifica che si utilizzano le funzioni di `Rcpp` attraverso l'argomento `plugin`. Una volta che il programma è stato lanciato si crea una funzione `fibRcpp` che è a tutti gli effetti una funzione in R.

Quanto appena descritto è la strada più corretta per l'inserimento di codice C++ in uno script R. Esiste tuttavia un'alternativa recente che evita l'uso del pacchetto `inline` e si serve della funzione `sourceCpp` di `Rcpp` per compilare direttamente codice C++ durante l'esecuzione del programma R. Il programmatore deve solamente dichiarare alcuni "attributi" di C++ per l'esportazione della funzione in R, in questo caso `[[Rcpp::export]]`. Questa modalità è sconsigliabile in quanto meno elegante e leggermente più lenta della precedente, ma a parere dell'autore è facilmente comunicabile data la sua stretta analogia con la funzione `source()` già nota ad un utilizzatore R. Ancora una volta il risultato del codice, lanciato interamente in R, è la creazione di una funzione R `fibonacci()` richiamabile in modo classico.

```

sourceCpp( code= '
    #include <Rcpp.h>
    using namespace Rcpp;
    // [[Rcpp::export]]
5    int fibonacci(const int x) {
        if (x == 0) return(0);
        if (x == 1) return(1);
        return (fibonacci(x-1) + fibonacci(x-2));
    }

```

```

10  }
    ')

```

Per concludere la Tabella A.1 si fornisce un esempio della differenza di tempi computazionali fra la funzione scritta puramente in R e quella che fa uso di C++. Si nota che il tempo necessario per la funzione R, scritta in modo molto inefficiente, cresce esponenzialmente. Mentre la stessa funzione tradotta in C++ e richiamata in R rimane competitiva al crescere di n e non presenta problemi di scala.

| n | fibR(n) | fibRcpp(n) | ratio |
|----|---------|------------|-------|
| 1 | 0.00 | 0.00 | - |
| 5 | 0.00 | 0.00 | - |
| 10 | 0.00 | 0.00 | - |
| 15 | 0.00 | 0.00 | - |
| 20 | 0.11 | 0.00 | - |
| 25 | 1.17 | 0.00 | - |
| 30 | 13.01 | 0.02 | 651 |
| 35 | 145.64 | 0.08 | 1821 |
| 40 | 1637.20 | 0.95 | 1723 |

Tabella A.1: Confronto dei tempi computazionali in secondi per funzioni ricorsive in R e Rcpp.

A.3 RcppArmadillo

RcppArmadillo ad opera di Francois *et al.* (2016) è un pacchetto creato dagli stessi autori per integrare in R la libreria **Armadillo** (si veda Sanderson, 2010, e Conrad & Curtin, 2016) di C++ appoggiandosi alle caratteristiche e all'interfaccia di **Rcpp**. Come dichiarato dall'autore, **Armadillo** è una libreria di C++ di algebra lineare che mira a bilanciare velocità di esecuzione e facilità d'utilizzo. La sintassi è deliberatamente ispirata a Matlab, ma non si discosta troppo da quella di R. Oltre ai semplici tipi di variabile quali `int`, `float` e `double` sono definite delle classi ad hoc per vettori, matrici ed array a tre dimensioni, chiamati cubi. Vi sono funzioni trigonometriche e statistiche, in particolare la generazione di numeri casuali e funzioni di analisi descrittiva

ed esplorativa, come ad esempio k-means, componenti principali e modelli di mistura Gaussiani. Di particolare rilevanza è ovviamente l'attenzione rivolta alle operazioni matriciali che di fatto ha reso Armadillo la libreria di riferimento, superando altre librerie precedenti quali BLAS, LAPACK, Newmat e IT++ che presentavano limitazioni sia dal punto di vista dei contenuti sia dal punto di vista legale circa la licenza concessa.

Gli obiettivi chiave cui Armadillo è chiamato a rispondere sono:

- una sintassi facile per l'algebra ispirata a Matlab;
- supporto di più tipi di elementi tipici dell'algebra;
- decomposizione e manipolazione di matrici (appoggiandosi a LAPACK);
- accesso rapido alle sotto-matrici;
- fornire una licenza appropriata sia per lo sviluppo open source sia per software (proprietary) closed source (è stata usata la licenza LGPL);
- possibilità di utilizzo su piattaforme Unix e MS Windows.

Nella Tabella A.2 presa da Conrad & Curtin (2016) si riportano, a titolo esemplificativo delle caratteristiche di Armadillo, le principali funzioni presenti nel codice di `modRcpp_1pl` e `modRcpp_2pl` riportato in Appendice B.

| Nome funzione | Descrizione |
|---|--|
| <code>A * k</code> | prodotto matrice per scalare |
| <code>A * B</code> | prodotto matriciale |
| <code>A % B</code> | prodotto fra matrici element-wise |
| <code>pow(A, p)</code> | elevamento a potenza A^p element-wise |
| <code>inv(A)</code> | inversione di matrice |
| <code>solve(A,B)</code> | soluzione del sistema di tipo $AX = B$ |
| <code>svd(A)</code> | scomposizione di A in valori singolari |
| <code>regspace<vect>(a, b)</code> | creazione di un vettore sequenza da a a b |
| <code>zeros<mat>(k,k)</code> | creazione di matrice di 0 |
| <code>diagmat(vect)</code> | creazione di una matrice diagonale da vettore |
| <code>A(p,q)</code> | accesso alla cella della matrice |
| <code>A.col(q)</code> | accesso alla colonna della matrice |
| <code>A.submat(p1,q1,p2,q2)</code> | accesso alla sotto matrice |
| <code>A.submat(vect_row, vect_col)</code> | accesso alla sotto matrice con vettore di indici |
| <code>A.diag(k)</code> | accesso alla k-esima diagonale |

Tabella A.2: Esempio di funzioni della libreria Armadillo.

Appendice B

Codice R per le funzioni di stima del modello di Rasch

Codice R della funzione `mbrRasch.1pl` per la stima del modello di Rasch ad un parametro

```
mbrRasch.1pl <- function(y, X, S, I, start=NULL,
                        max.iter=500, epsilon, trace=F)
{
  p <- S+I-1
  5  if(is.null(start)) par <- numeric(p)
  # X e' gia' una model frame

  iter=0
  is.far=T
10
  ## NEWTON RAPHSON
  while(is.far & iter<max.iter){
    eta <- as.vector(X%*%par)
    F_eta <- plogis(eta)
    15  dF_eta <- dlogis(eta)
    d2F_eta <- plogis(eta)*(1-plogis(eta))*(1-2*plogis(eta))

    Info <- t(X)%*%(dF_eta*X)  ##i(beta)
    Infoinv <- solve(Info)    ##Var(beta)
20

    W <- 1*dF_eta
    u <- (y - F_eta)/dF_eta
```

```

    m1 <- modRcpp_1pl(S=S, I=I, Imeno1=Infoinv, Fsecondo=d2F_eta) * diag
      (Infoinv)
    z <- X %*% (par + m1) + u
25  old.par <- par
    par <- Infoinv %*% t(X)%*% (W * z)

    iter = iter +1
    dist <- sum((par-old.par)^2)
30  if(dist<epsilon) is.far <- F

    if(trace){
      cat("\n")
      cat("Iteration:", iter, "\n")
35  cat("Shift:", dist)
    }
  }
  #risultato finale
  eta <- as.vector(X%*%par)
40  dF_eta <- dlogis(eta)
  Info <- t(X)%*%(dF_eta*X)
  Infoinv <- solve(Info)
  se <- sqrt(diag(Infoinv))

45  return(list(coefficients=par, se=se, iterations=iter))
}

```

Codice in C++ della funzione `modRcpp_1pl` per il calcolo della modificazione per il modell di Rasch ad un parametro

```

sourceCpp(code='
  // [[Rcpp::depends(RcppArmadillo)]]
  #include <RcppArmadillo.h>
4  using namespace Rcpp;
  // [[Rcpp::export]]
  arma::vec modRcpp_1pl (int S, int I, arma::mat Imeno1, arma::vec
    Fsecondo){

    int p = S+I-1;
9  arma::vec k1 =arma::zeros<arma::vec>(p); //px1
  arma::vec k3 =arma::zeros<arma::vec>(p); //px1
  arma::vec k2 = 1 / arma::diagvec(Imeno1); //px1

```



```

arma::mat barA = arma::diagmat(k2) * Imeno1;
14 arma::mat numenol; //pxp
double nusss; //scalare
arma::vec nusst, nustt; //px1
arma::vec vett1_s, vett3_s; //px1
int rr,ss; //indici dei for
19 arma::uvec tval, ind1, ind2, rind, sind; //indici dei vettori
double Ars; //scalare, migliora lettura
arma::vec result; //px1

for(rr=1; rr<p+1; rr++){ //parte da 1 non da 0!
24 vett1_s = arma::zeros<arma::vec>(p);
vett3_s = arma::zeros<arma::vec>(p);
arma::vec col = Imeno1.col(rr-1);
numenol = Imeno1 - (col * k2(rr-1) * col.t());
rind = rr; //indice per matrice, vett di 1x1
29

for(ss=1; ss<p+1; ss++){ //parte da 1 non da 0!

if(ss<=S) { //ss e' sogg, t e' item
tval = arma::regspace<arma::uvec>(S+1,p);
34 ind1 = ss+arma::regspace<arma::uvec>(0,I-1)*S;
ind2 = ss+arma::regspace<arma::uvec>(1,tval.n_elem)*S;
} else { //s e' item, t e' sogg
tval = arma::regspace<arma::uvec>(1,S);
ind1 = tval + (ss-S)*S;
39 ind2 = ind1;
}
nusss = sum(Fsecondo.elem(ind1-1)); //scalare
nusst = Fsecondo.elem(ind2-1); //length(tval)x1
nustt = nusst; //length(tval)x1
44 Ars = arma::as_scalar(barA(rr-1,ss-1)); //scalare
sind = ss; //indice

vett1_s(ss-1) = Ars * (
arma::as_scalar(numenol(ss-1,ss-1)*nusss)
49 + 2*arma::as_scalar(numenol.submat(sind-1,tval-1)*
nusst)
+ arma::as_scalar(arma::trans(diagvec(numenol.
submat(tval-1,tval-1))*nustt) );
vett3_s(ss-1) = pow(Ars,3) * nusss

```

```

+ 2*pow(Ars,2) *arma::as_scalar(barA.submat(rind
-1,tval-1) *nusst)
+ Ars *arma::as_scalar(pow(barA.submat(rind-1,tval
-1),2) *nustt);
54   }

k1(rr-1) = -0.5*sum(vett1_s);
k3(rr-1) = sum(vett3_s);
}
59 result = -k1 +(k3/k2)/6;
return result;
}
')
```

Codice R della funzione `mbrRasch.2pl` per la stima del modello di Rasch a due parametri

```

mbrRasch.2pl <- function(data,start=NULL,max.iter=50,epsilon=1e-2,trace=F)
{
3   #funzione che crea la matrice X
x.fun <- function(gamma,alpha,beta,S,I,n,p){
  out <- matrix(0,n,p)
  #gamma --> I blocchi, ognuno di SxS
  for(i in 1:I){
8     diag(out[(S*(i-1)+1):(S*i),1:S]) <- rep(beta[i],S) }
  #alpha --> I blocchi, ognuno di Sx(I-1)
  for(i in 2:I){
    out[(S*(i-1)+1):(S*i),S+i-1] <- rep(1,S) }
  #alpha --> I blocchi, ognuno di Sx(I-1)
13  for(i in 2:I){
    out[(S*(i-1)+1):(S*i),(S+I-1)+i-1] <- gamma }
  return(out)
}

18 #funzioni utili
f1 <- function(x) x*(1-x)
f2 <- function(x) x*(1-x)*(1-2*x)

#lettura dati
23 S <- nrow(data)
I <- ncol(data)
```

```

d <- data.frame(y=data[1:(S*I)],
               subject=factor(rep(1:S,I)),
               item=factor(rep(1:I,each=S)))
28 y <- d$y

#quantita' costanti
p <- S+2*(I-1)
n <- S*I
33 i_gamma <- 1:S
i_alpha <- (S+1):(S+I-1)
i_beta <- (S+I):p

if(is.null(start)){
38   if(trace) cat("Initialize parameters ...")
   Xs <- model.matrix(y~1+subject+item, data=d)
   m0 <- mbrRasch.lpl(y, Xs, S, I, start=NULL, epsilon=1e-2, trace=F)
   start <- c(m0$coef, jitter(rep(1,I-1)))
   rm(m0,Xs)
43 }

iter=0
is.far=T
step <- .Machine$integer.max
48 par <- start

## NEWTON RAPHSON
while(is.far & iter<max.iter){

53   gamma <- par[i_gamma] #Sx1
   alpha <- c(0,par[i_alpha]) #Ix1
   beta <- c(1,par[i_beta ]) #Ix1

   X <- x.fun(gamma, alpha, beta, S, I, n, p) #nxp
58   eta.mat <- t(apply(outer(gamma,beta),1,function(riga) riga + alpha))
   pi.mat <- plogis(eta.mat) #SxI
   pigreco <- c(pi.mat)      #nx1
   W <- diag(fl(pigreco))   #nxn diag
   score <- crossprod(X,(y-pigreco)) #px1
63   Info <- crossprod(X,(diag(W)*X)) #pxp
   Infoinv <- solve(Info)   #pxp

   mrcpp <- modRcpp_2pl(S,I,Infoinv,gamma,alpha,beta,pi.mat) #px1
   m1 <- diag(Infoinv)*mrcpp #px1

```

```

68
#step halving
stepPrev <- step
stepFactor <- 0
73
testhalf <- TRUE
while(testhalf & stepFactor < 11){
  step <- 2^(-stepFactor) * (Infoinv %*% (score + Info %*% m1))
  testhalf <- drop(crossprod(stepPrev) < crossprod(step))
  stepFactor <- stepFactor + 1
78
}

#nuova stima
par <- par + step

83
iter = iter +1
#dist <- sum((par-old.par)^2)
#if(dist<epsilon) is.far <- F
if(sqrt(crossprod(step)) < epsilon) is.far <- F

88
if(trace){
  cat("\n")
  cat("Iteration:", iter, "\n")
  cat("Step Factor:", stepFactor-1, " | Shift:",
      sqrt(crossprod(step)) )
93
}
}

#stima finale
gamma <- par[i_gamma] #Sx1
98
alpha <- c(0,par[i_alpha]) #Ix1
beta <- c(1,par[i_beta ]) #Ix1

X <- x.fun(gamma, alpha, beta, S, I, n, p) #nxp
eta.mat <- t(apply(outer(gamma,beta),1,function(riga) riga + alpha))
103
pi.mat <- plogis(eta.mat) #SxI
pigreco <- c(pi.mat) #nx1
W <- diag(f1(pigreco)) #nxn diag
Info <- t(X) %*% (diag(W)*X) #pxp
Infoinv <- solve(Info) #pxp
108
invisible(list(coefficients=c(par), se=sqrt(diag(Infoinv)), iteration=
  iter))

```

}

Codice in C++ della funzione `modRcpp_2pl` per il calcolo della modificazione per il modell di Rasch a due parametri

```

sourceCpp(code='
  // [[Rcpp::depends(RcppArmadillo)]]
  #include <RcppArmadillo.h>
  using namespace Rcpp;
5  // [[Rcpp::export]]
  arma::vec modRcpp_2pl (int S, int I, arma::mat Imeno1,
                        arma::vec gamma, arma::vec alpha,
                        arma::vec beta, arma::mat pimat){

10     const int p = S+2*(I-1);
        const int g1 = 1 -1;      //indice
        const int g2 = S -1;      //indice
        const int a1 = S+1 -1;    //indice
        const int a2 = S+I-1 -1;  //indice
15     const int b1 = S+I -1;     //indice
        const int b2 = p -1;      //indice

        int rr, aa, ss, ii; //indici for

20     arma::vec k1 = arma::zeros<arma::vec>(p); //px1
        arma::vec k3 = arma::zeros<arma::vec>(p); //px1
        arma::vec k2 = 1 / arma::diagvec(Imeno1); //px1

        arma::mat barA = arma::diagmat(k2) * Imeno1; //pxp
25     arma::mat numeno1; //pxp
        arma::mat pi1 = pimat % (1-pimat); //SxI
        arma::mat pi2 = pimat % (1-pimat) % (1-2*pimat); //SxI
        arma::vec piltemp, pi2temp;

30     arma::vec vett1_a, vett3_a; //px1
        arma::vec result; //px1

        double barAra; //scal

35     double nu_ggg;
        arma::vec nu_gga, nu_ggb, nu_gaa, nu_gab, nu_gbb, nu_g_gb;

```

```

double nu_aaa, nu_aab, nu_abb;
arma::vec nu_agg, nu_aga, nu_agb, nu_a_gb;
double nu_baa, nu_bab, nu_bbb;
40 arma::vec nu_bgg, nu_bga, nu_bgb, nu_b_gb;

for(rr=0; rr<p; rr++){
  vett1_a = arma::zeros<arma::vec>(p);
  vett3_a = arma::zeros<arma::vec>(p);
45 arma::vec col = Imeno1.col(rr);
  numenol = Imeno1 - (col * k2(rr) * col.t());

  for(aa=0; aa<p; aa++){

50     if((aa>=g1) && (aa<=g2)) { //aa e' un gamma_(ss)
        ss = aa;
        pi1temp = vectorise(pi1.row(ss)); //Ix1
        pi2temp = vectorise(pi2.row(ss));

55         nu_ggg = arma::as_scalar(arma::trans(pow(beta,3)) *
            pi2temp);
        nu_gga = pow(beta,2) % pi2temp;
        nu_ggb = gamma(ss) * (pow(beta,2) % pi2temp);
        nu_gaa = beta % pi2temp;
        nu_gab = gamma(ss) * (beta % pi2temp);
60         nu_gbb = pow(gamma(ss),2) * (beta % pi2temp);
        nu_g_gb = beta % pi1temp;

        nu_gga = nu_gga.subvec(1,I-1); // (I-1)x1
        nu_ggb = nu_ggb.subvec(1,I-1); // (I-1)x1
65         nu_gaa = nu_gaa.subvec(1,I-1); // (I-1)x1
        nu_gab = nu_gab.subvec(1,I-1); // (I-1)x1
        nu_gbb = nu_gbb.subvec(1,I-1); // (I-1)x1
        nu_g_gb = nu_g_gb.subvec(1,I-1); // (I-1)x1

70         barAra = arma::as_scalar(barA(rr,aa)); //scalare

        vett1_a(aa) = barAra * arma::as_scalar(
            numenol(aa,aa) * nu_ggg +
            2* numenol.submat(aa,a1,aa,a2) * nu_gga +
75         2* numenol.submat(aa,b1,aa,b2) * (nu_ggb + nu_g_gb)+
            arma::trans(diagvec(numenol.submat(a1,a1,a2,a2)))*nu_gaa
            +2*arma::trans(diagvec(numenol.submat(a1,b1,a2,b2)))
            *nu_gab+

```

```

        arma::trans(diagvec(numeno1.submat(b1,b1,b2,b2)))*nu_gbb
    );
80
    vett3_a(aa) = barAra * arma::as_scalar(
        pow(barAra,2) * nu_ggg +
        2* barAra* ( barA.submat(rr,a1,rr,a2) * nu_gga ) +
        2* barAra* ( barA.submat(rr,b1,rr,b2) * nu_ggb ) +
85
        pow(barA.submat(rr,a1,rr,a2),2) * nu_gaa +
        2* sum( vectorise(barA.submat(rr,a1,rr,a2)) % vectorise(
            barA.submat(rr,b1,rr,b2)) % nu_gab) +
        pow(barA.submat(rr,b1,rr,b2),2) * nu_gbb
    );
    }
90
    if((aa>=a1) && (aa<=a2)){ //aa e' un alpha_(ii)
        ii = aa - S +1;
        pi1temp = pi1.col(ii); //Sx1
        pi2temp = pi2.col(ii); //Sx1
95
        nu_agg = pow(beta(ii),2) * pi2temp; //Sx1
        nu_aga = beta(ii) * pi2temp; //Sx1
        nu_agb = beta(ii) * (gamma % pi2temp); //Sx1
        nu_aaa = arma::as_scalar(sum(pi2temp));
100
        nu_aab = arma::as_scalar(arma::trans(gamma) * pi2temp);
        nu_abb = arma::as_scalar(arma::trans(pow(gamma,2)) *
            pi2temp);
        nu_a_gb = pi1temp; //Sx1

        barAra = arma::as_scalar(barA(rr,aa)); //scalare
105

        vett1_a(aa) = barAra * arma::as_scalar(
            arma::trans(diagvec(numeno1.submat(g1,g1,g2,g2)))*nu_agg+
            2* arma::trans(numeno1.submat(g1,aa,g2,aa)) * nu_aga +
            2* arma::trans(numeno1.submat(g1,aa+I-1,g2,aa+I-1)) *
110
                (nu_agb + nu_a_gb) +
            numeno1(aa,aa) * nu_aaa +
            2* numeno1(aa,aa+I-1) * nu_aab +
            numeno1(aa+I-1,aa+I-1) * nu_abb
        );
115

        vett3_a(aa) = barAra * arma::as_scalar(
            pow(barA.submat(rr,g1,rr,g2),2) * nu_agg +
            2* barAra * (barA.submat(rr,g1,rr,g2) * nu_aga) +

```

```

120     2* barA(rr,aa+I-1) * (barA.submat(rr,g1,rr,g2) * nu_agb) +
        pow(barAra,2) * nu_aaa +
        2* barAra * barA(rr,aa+I-1) * nu_aab +
        pow(barA(rr,aa+I-1),2) * nu_abb
    );
}
125
if((aa>=b1) && (aa<=b2)){ //aa e' un beta_(ii)
    ii = aa - (S+I-1) +1;
    piltemp = pil.col(ii); //Sx1
    pi2temp = pi2.col(ii); //Sx1
130
    nu_bgg = pow(beta(ii),2) * (gamma % pi2temp); //Sx1
    nu_bga = beta(ii) * (gamma % pi2temp); //Sx1
    nu_bgb = beta(ii) * (pow(gamma,2) % pi2temp); //Sx1
    nu_baa = arma::as_scalar(arma::trans(gamma) * pi2temp);
        //scal
135
    nu_bab = arma::as_scalar(arma::trans(pow(gamma,2)) *
        pi2temp); //scal
    nu_bbb = arma::as_scalar(arma::trans(pow(gamma,3)) *
        pi2temp); //scal
    nu_b_gb = gamma % piltemp; //Sx1

    barAra = arma::as_scalar(barA(rr,aa)); //scalare
140
    vett1_a(aa) = barAra * arma::as_scalar(
        arma::trans(diagvec(numeno1.submat(g1,g1,g2,g2)))*nu_bgg+
        2*arma::trans(numeno1.submat(g1,aa-I+1,g2,aa-I+1))*nu_bga+
        2*arma::trans(numeno1.submat(g1,aa,g2,aa)) *
145
        (nu_bgb + nu_b_gb)+
        numeno1(aa-I+1,aa-I+1) * nu_baa +
        2* numeno1(aa-I+1,aa) * nu_bab +
        numeno1(aa,aa) * nu_bbb
    );
150
    vett3_a(aa) = barAra * arma::as_scalar(
        pow(barA.submat(rr,g1,rr,g2),2) * nu_bgg +
        2* barA(rr,aa-I+1) * (barA.submat(rr,g1,rr,g2) * nu_bga) +
        2* barAra * (barA.submat(rr,g1,rr,g2) * nu_bgb) +
155
        pow(barA(rr,aa-I+1),2) * nu_baa +
        2* barA(rr,aa-I+1) * barAra * nu_bab +
        pow(barAra,2) * nu_bbb
    );

```

```
    }  
160     } //chiudo for di aa  
  
    k1(rr) = -0.5* sum(vett1_a);  
    k3(rr) = sum(vett3_a);  
165 } //chiudo for di rr  
  
    result = -k1 +(k3/k2)/6;  
    return result;  
170 }  
,)
```

Appendice C

Codice delle funzioni per il calcolo della modificazione della score

In questa appendice si riportano i codici utilizzati per il confronto fra funzioni presentate nella Tabella 3.1.

Fun 1

```
fun1 <- function(theta, X){
  eta <- as.vector(X%%theta)
  F_eta <- plogis(eta)
4  dF_eta <- dlogis(eta)
  d2F_eta <- plogis(eta)*(1-plogis(eta))*(1-2*plogis(eta))
  A <- rep(1,S*I)
  Info <- t(X)%%(dF_eta*X)
  Infoinv <- solve(Info)
9
  k2 <- 1/diag(Infoinv)      ##k2
  barA <- Infoinv*k2

  nu_r_s_t <- array(0,dim=rep(p,3))
14  for (r in 1:p){
    nu_r_s_t[r,,] <- t(X)%% diag(A^3*d2F_eta*X[,r]) %% X
  }

  k1 <- rep(0,p)           ##k1
19  k3 <- rep(0,p)           ##k3
  for(r in 1:p){
```

```

vett1_s=vett1_t=vett1_u=rep(0,p)
vett3_s=vett3_t=vett3_u=rep(0,p)
for(s in 1:p){
24   for(t in 1:p){
       for(u in 1:p){
           nu__tu <- Infoinv[t,u]-Infoinv[t,r]*k2[r]*Infoinv[r,u]
           vett1_u[u] <- nu__tu*(nu_r_s_t[s,t,u])
           vett3_u[u] <- barA[r,u]*nu_r_s_t[s,t,u]
29       }
       vett1_t[t] <- sum(vett1_u) #integro u
       vett3_t[t] <- barA[r,t]*sum(vett3_u) #integro u
       }
34   vett1_s[s] <- barA[r,s]*sum(vett1_t) #integro t
       vett3_s[s] <- barA[r,s]*sum(vett3_t) #integro t
       }
       k1[r] <- -0.5*sum(vett1_s) #integro s
       k3[r] <- sum(vett3_s) #integro s
39   }
   m <- (-k1+k3/k2/6)
   return(m)
}

```

Fun 2

```

fun2 <- function(theta, X){
eta <- as.vector(X%%theta)
F_eta <- plogis(eta)
4 dF_eta <- dlogis(eta)
d2F_eta <- plogis(eta)*(1-plogis(eta))*(1-2*plogis(eta))
A <- rep(1,S*I)
Info <- t(X)%*(dF_eta*X)
Infoinv <- solve(Info)
9
k2 <- 1/diag(Infoinv)      ##k2
barA <- Infoinv*k2

nu_r_s_t <- array(0,dim=rep(p,3))
14
for (r in 1:p) {
   nu_r_s_t[r,,] <- t(X)%*((A^3*d2F_eta*X[,r])*X)
}

```

```

19 k1 <- rep(0,p)          ##k1
   k3 <- rep(0,p)          ##k3
   for (r in 1:p) {
     sum_s1 <- rep(0,p)
     sum_s3 <- rep(0,p)
24   nu.tu <- Infoinv-outer(Infoinv[,r]*k2[r],Infoinv[,r])
     for (s in 1:p){
       sum_s1[s] <- sum(diag(nu.tu%%(nu_r_s_t[s,,])))
     }
     k1[r] <- -0.5*sum(sum_s1*barA[r,])
29   k3[r] <- sum(sum_s3*barA[r,])
   }

   m <- (-k1+k3/k2/6)
   return(m)
34 }

```

Fun 3

```

1 fun3 <- function(theta, X){
   eta <- as.vector(X%%theta)
   F_eta <- plogis(eta)
   dF_eta <- dlogis(eta)
   d2F_eta <- plogis(eta)*(1-plogis(eta))*(1-2*plogis(eta))
6   A <- rep(1,S*I)
   Info <- t(X)%%(dF_eta*X)
   Infoinv <- solve(Info)

   k2 <- 1/diag(Infoinv)      ##k2
11  barA <- Infoinv*k2

   k1 <- numeric(p)          ##k1
   k3 <- numeric(p)          ##k3
   for( r in 1:p){
16   vett1_s <- vett3_s <- numeric(p)
     nu.inv <- Infoinv-outer(Infoinv[,r]*k2[r],Infoinv[,r])
     for(s in 1:p){
       if(s<=S) { #s e' sogg, t e' item
         tval <- (S+1):p
21         nu_s_s_s <- sum(d2F_eta[s+(0:(I-1))*S])
           nu_s_s_t <- nu_s_t_t <- d2F_eta[s+(1:length(tval))*S]
       } else { #s e' item, t e' sogg

```

```

    tval <- 1:S
    nu_s_s_s <- sum(d2F_eta[tval+(s-S)*S])
26    nu_s_s_t <- nu_s_t_t <- d2F_eta[tval+(s-S)*S]
  }
  vett1_s[s] <- barA[r,s]*(nu.inv[s,s]*nu_s_s_s + 2*sum(nu.inv[s,
    tval]*nu_s_s_t) + sum(diag(nu.inv[tval,tval])*nu_s_t_t))
  vett3_s[s] <- barA[r,s]^3*nu_s_s_s + 2*barA[r,s]^2*sum(barA[r,
    tval]*nu_s_s_t) + barA[r,s]*sum(barA[r,tval]^2*nu_s_t_t)
  }
31  k1[r] <- -0.5*sum(vett1_s)
  k3[r] <- sum(vett3_s)
}
m <- (-k1+k3/k2/6)
  return(m)
36 }

```

Fun 4

```

library(foreach)
library(doParallel)
fun4 <- function(theta, X){
4  eta <- as.vector(X%%theta)
  F_eta <- plogis(eta)
  dF_eta <- dlogis(eta)
  d2F_eta <- plogis(eta)*(1-plogis(eta))*(1-2*plogis(eta))
  A <- rep(1,S*I)
9  Info <- t(X)%*(dF_eta*X)
  Infoinv <- solve(Info)

  k2 <- 1/diag(Infoinv)      ##k2
  barA <- Infoinv*k2

14  k1 <- numeric(p)        ##k1
  k3 <- numeric(p)        ##k3
  #getDoParWorkers() #quanti core ci sono?
  options(cores=2)
19  cl <- makeCluster(2)
  registerDoParallel(cl)

  foreach(r=1:p, .combine='cbind') %dopar% {
24    vett1_s <- vett3_s <- numeric(p)
    nu.inv <- Infoinv-outer(Infoinv[,r]*k2[r],Infoinv[,r])
  }
}

```

```

for(s in 1:p){
  if(s<=S) { #s e' sogg, t e' item
    tval <- (S+1):p
    nu_s_s_s <- sum(d2F_eta[s+(0:(I-1))*S])
    nu_s_s_t <- nu_s_t_t <- d2F_eta[(1:length(tval))*S+s]
  } else { #s e' item, t e' sogg
    tval <- 1:S
    nu_s_s_s <- sum(d2F_eta[(1:S)+(s-S)*S])
    nu_s_s_t <- nu_s_t_t <- d2F_eta[(1:length(tval))+(s-S)*S]
  }

  vett1_s[s] <- barA[r,s]*(nu.inv[s,s]*nu_s_s_s + 2*sum(nu.inv[s,
    tval]*nu_s_s_t) + sum(diag(nu.inv[tval,tval])*nu_s_t_t))
  vett3_s[s] <- barA[r,s]^3*nu_s_s_s + 2*barA[r,s]^2*sum(barA[r,
    tval]*nu_s_s_t) + barA[r,s]*sum(barA[r,tval]^2*nu_s_t_t)
}
k1[r] <- -0.5*sum(vett1_s)
k3[r] <- sum(vett3_s)
}
m <- (-k1+k3/k2/6)
return(m)
}

```

Fun 5

Per il codice C++ della funzione `modRcpp_1pl`, richiamata dalla Fun 5, si veda l'Appendice B

```

1 fun5 <- function(theta, X, S, I){
  eta <- as.vector(X**theta)
  F_eta <- plogis(eta)
  dF_eta <- dlogis(eta)
  d2F_eta <- plogis(eta)*(1-plogis(eta))*(1-2*plogis(eta))
  Info <- t(X)**(dF_eta*X)
  Infoinv <- solve(Info)

  m <- modRcpp_1pl(S=S, I=I, Imeno1=Infoinv, Fsecondo=d2F_eta)
  return(m)
11 }

```


Bibliografia

- ADLER, D. (2012). `rdyncall`: Improved Foreign Function Interface (FFI) and Dynamic Bindings to C Libraries. URL <http://CRAN.R-Project.org/package=rdyncall>, R package version 0.7.5.
- ANDERSEN, E. B. (1970). Asymptotic properties of conditional maximum likelihood estimators. *Journal of the Royal Statistical Society B* **32**, 283-301.
- ANDERSEN, E. B. (1980). *Discrete Statistical Models with Social Science Applications*. North-Holland, Amsterdam.
- ARMSTRONG, W. (2009). `RObjects`: C++ wrapper for R objects (a better implementation of `RAbstraction`). URL <http://github.com/armstrtw/RObjects>, code repository aggiornato al 28 nov 2009.
- ARMSTRONG, W. (2014). `RAbstraction`: C++ abstraction for R objects. URL <http://github.com/armstrtw/rabstraction>, code repository aggiornato al 14 lug 2014.
- BARTLETT, M.S. (1953). Approximate confidence intervals. II. More than one unknown parameter. *Biometrika* **40**, 306-317.
- BATES, D. & DEBROY, S. (2001). C++ classes for R objects. In: *Hornik K, Leisch F (eds) Proceedings of the 2nd International Workshop on Distributed Statistical Computing (DSC 2001)*, TU Vienna, Austria.

- BATES, D et al. (2015). Fitting Linear Mixed-Effects Models Using lme4. *Journal of Statistical Software* **67(1)**, 1-48.
- BIRNBAUM, A. (1957). *On the Estimation of Mental Ability*. Rep. No. **15**, Randolph Air Force Base, U.S. Air Force School of Aviation Medicine, Texas.
- CAI, T. T. & WANG, H. (2009). Tolerance intervals for discrete distributions in exponential families. *Statistica Sinica* **19**, 905–923.
- CALAWAY, R. (2015a). doParallel: Foreach Parallel Adaptor for the 'parallel' Package. URL <http://CRAN.R-Project.org/package=doParallel>. R package version 1.0.10.
- CALAWAY, R. (2015b). foreach: Provides Foreach Looping Construct for R. URL <http://CRAN.R-Project.org/package=foreach>. R package version 1.4.3.
- CHALMERS, P. (2016). mirt: Multidimensional Item Response Theory. URL <http://CRAN.R-Project.org/package=mirt>. R package version 1.20.1.
- CHAMBERS, J. M. (1998). *Programming with Data: A Guide to the S Language*. Springer, Heidelberg.
- CONRAD S. & CURTIN R. (2016). Armadillo: a template-based C++ library for linear algebra. *Journal of Open Source Software* **1**, 26-32.
- CRESSIE, N. E. & HOLLAND, P. W. (1983). Characterizing the manifest probabilities of latent trait models. *Psychometrika* **48**, 129–141.
- DAVISON, A. C. (2003). *Statistical Models*. Cambridge University Press, Cambridge.
- DE BOECK, P. (2008). Random item IRT models. *Psychometrika* **4**, 533–559.
- EDDELBUETTEL, D. (2013). *Seamless R and C++ integration with Rcpp*. Springer-Verlag, New York.

- EDDELBUETTEL, D. & FRANCOIS, S. (2016a). Rcpp: Seamless R and C++ Integration. URL <http://CRAN.R-Project.org/package=Rcpp>, R package version 0.12.7.
- EDDELBUETTEL, D. & NGUYEN, K. (2016b). RQuantLib: R interface to the QuantLib library. URL <http://CRAN.R-Project.org/package=RQuantLib>, R package version 0.4.3.
- FIENBERG S. E. (2006). The Rasch model *Encyclopedia of Statistical Sciences*, S. Kotz, N. Johnson & C. Read, eds. Wiley, New York.
- FIRTH, D. (1993). Bias reduction of maximum likelihood estimates. *Biometrika* **80**, 27–38.
- FRANCOIS, S., EDDELBUETTEL, D. & BATES, D. (2016). RcppArmadillo: Rcpp integration for Armadillo templated linear algebra library. URL <http://CRAN.R-Project.org/package=RcppArmadillo>, R package version 0.7.500.0.0.
- FUCHS, L. et al (2001). Oral Reading Fluency as an Indicator of Reading Competence: A Theoretical, Empirical, and Historical Analysis. *Scientific studies of reading* **5**, 239-256.
- HABERMAN, S. J. (1977). Maximum likelihood estimation in exponential response models. *Journal of Applied Statistics* **5**, 815-841.
- KENNE PAGUI, E. C., SALVAN, A. & SARTORI, N. (2016a). Median bias reduction of maximum likelihood estimates. <http://arxiv.org/abs/1604.04768>.
- KENNE PAGUI, E. C., SALVAN, A. & SARTORI, N. (2016b). mbrglm: Median bias reduction in Binomial-response GLMs. URL <https://github.com/eulogepagui/mbrglm>, code repository aggiornato al 8 set 2016.
- KOSMIDIS, I. (2013). brglm: Bias reduction in binomial-response generalized linear models. URL <http://CRAN.R-Project.org/package=brglm>, R package version 0.5-9.

- KOSMIDIS, I. (2014). Bias in parametric estimation: reduction and useful sideeffects. *Wiley Interdisciplinary Reviews: Computational Statistics* **6**, 185–196.
- KOSMIDIS, I. (2016). brRasch: Maximum likelihood and bias reduction for fixed-effects Rasch models. URL <https://github.com/ikosmidis/brRasch/tree/master/R>, code repository aggiornato al 3 mag 2016.
- KOSMIDIS, I. & FIRTH, D. (2009). Bias reduction in exponential family nonlinear models. *Biometrika* **96**, 793–804.
- KOSMIDIS, I. & FIRTH, D. (2010). A generic algorithm for reducing bias in parametric estimation. *Electronic Journal of Statistics* **4**, 1097–1112.
- LIANG, G. (2008). rcppbind: A template library for R/C++ developers. URL <http://R-Forge.R-Project.org/projects/rcppbind>, R package version 1.0.
- LINACRE, J. M. (2004). Rasch model estimation: further topics. *Journal of applied measurement* **5**, 95–110.
- LORD, F. M. & NOVICK, M. R. (1968). *Statistical Theories of Mental Test Scores*. Addison-Wesley, Reading, Mass.
- McCULLAGH, P. & TIBSHIRANI, R. (1990). A simple method for the adjustment of profile likelihoods. *Journal of the Royal Statistical Society Series B* **52**, 325–344.
- NEYMAN, J. & SCOTT, E. L. (1948), Consistent estimates based on partially consistent observations. *Econometrica* **16**, 1–32.
- OJA, H. (2013). Multivariate median. *Robustness and Complex Data Structures*, C. Becker, R. Fried & S. Kuhnt, eds. Springer, Berlin, 3–15.
- PACE, L. & SALVAN, A. (1992). A note on conditional cumulants in canonical exponential families. *Scandinavian Journal of Statistics* **19**, 185–191.

- PACE, L. & SALVAN, A. (1997). *Principles of Statistical Inference from a Neo-Fisherian Perspective*. World Scientific Pub Co Inc., Singapore.
- PERLINE, R. & WAINER, H. (1980). In *Indicators of Crime and Criminal Justice: Quantitative Studies*, S. E. Fienberg and A. J. Reiss, Jr., eds. U.S. Department of Justice, Washington, D.C., pp. 59–62.
- PINHEIRO, J. *et al* (2016). nlme: Linear and Nonlinear Mixed Effects Models. URL <http://CRAN.R-Project.org/package=nlme>, R package version 3.1-128.
- R DEVELOPMENT CORE TEAM (2016a). R: A language and environment for statistical computing. *R Foundation for Statistical Computing*, Vienna, Austria. URL <http://www.R-project.org/>.
- R DEVELOPMENT CORE TEAM (2016b). Writing R extensions. *R Foundation for Statistical Computing*, Vienna, Austria, URL <http://CRAN.R-Project.org/doc/manuals/R-exts.html>.
- RASCH, G. (1960). *Probabilistic models for some Intelligence and Attainment tests*. Studies in Mathematical Psychology I. Danish Inst. Educational Research, Copenhagen.
- READ, C. B. (1985). Median unbiased estimators. *Encyclopedia of Statistical Sciences*, S. Kotz, N. Johnson & C. Read, eds., vol. 5. Wiley, New York.
- RUNNALLS, A. (2009). *Aspects of CXXR internals*. Directions in Statistical Computing, University of Copenhagen, Denmark.
- SANDERSON, C. (2010) Armadillo: An open source C++ algebra library for fast prototyping and computationally intensive experiments. *Tech. rep.*, NICTA. URL <http://arma.sf.net>.
- SKLYAR, O *et al*. (2012). inline: Inline C, C++, Fortran function calls from R. URL <http://CRAN.R-Project.org/package=inline>, R package version 0.3.14.

- THERNEAU, T. (2016). survival: Survival Analysis. URL <http://CRAN.R-Project.org/package=survival>, R package version 2,40-1.
- URBANEK, S. (2003). Rserve: A fast way to provide R functionality to applications. In: *Hornik K, Leisch F, Zeileis A (eds) Proceedings of the 3rd International Workshop on Distributed Statistical Computing (DSC 2003)*, TU Vienna, Austria.
- URBANEK, S. (2012). Rserve: Binary R server. URL <http://CRAN.R-Project.org/package=Rserve>, R package version 1,7-3.
- WRIGHT, B . D . & MASTERS, G N. (1982) . *Rating scale analysis*. MESA Press, Chicago.

Ringraziamenti

Arrivati alla fine di un percorso come questo, i ringraziamenti, oltre che un fatto dovuto, diventano un piacere personale. Fra i primi, desidero ringraziare il mio relatore e correlatore per essere stati un esempio di competenza e disponibilità in tutti i colloqui che mi hanno generosamente concesso, è raro trovare un tale connubio.

Dicono che solo diventando genitori a propria volta si riesca a capire quanto sia difficile il ruolo del padre e della madre, ma ciò non mi impedisce di aver già apprezzato il loro impegno nell'educarmi e questo primo traguardo è dedicato a loro. Vorrei unire nei ringraziamenti anche la mia famiglia, mia sorella, i miei cugini, i miei zii e le mie nonne per essere stati un punto fermo in questo tempo di continui spostamenti.

Un ringraziamento ad Andrea e Mattia, che mi hanno fatto sentire a casa anche lontano da Milano; con loro, più di tutti, ho speso le giornate a studiare e le serate a ridere.

Se questi due anni a Padova saranno un bellissimo ricordo lo devo anche agli amici d'università dai quali ho imparato moltissimo, agli amici fuorisede con cui ho condiviso le gioie e le noie del vivere da soli ed ai miei colleghi tutor che hanno tramutato in piacere quello che, in teoria, doveva essere un dovere. Se, invece, nei weekend avevo voglia di tornare è perchè a Milano ritrovavo i miei amici di sempre.

Ora come allora, vorrei dedicare l'ultimo ringraziamento alla mia Francesca che mi è accanto da sei anni. Quando ho fatto fatica lei mi ha sostenuto e quando ero insicuro lei mi ha incoraggiato, ha reso questi i miei *anni felici*.