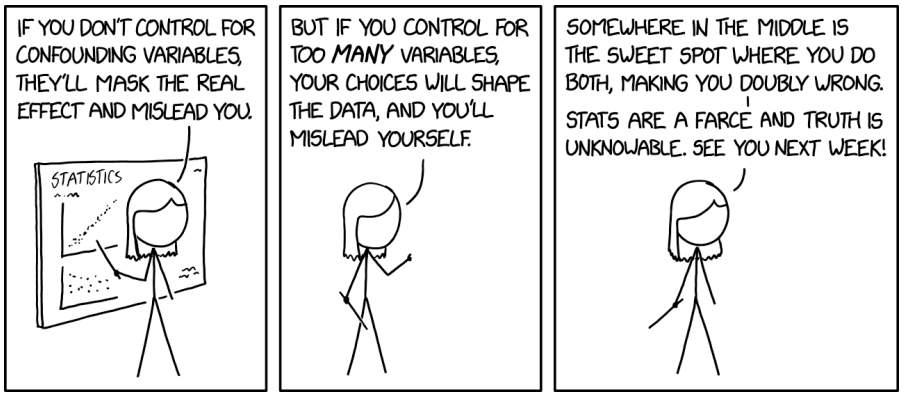
Corrado Caudek

Data Science per psicologi

Psicometria – AA 2021/2022



Indice

Elenco delle figure

Elenco delle tabelle

Prefazione

Data Science per psicologi contiene il materiale delle lezioni dell'insegnamento di *Psicometria B000286* (A.A. 2021/2022) rivolto agli studenti del primo anno del Corso di Laurea in Scienze e Tecniche Psicologiche dell'Università degli Studi di Firenze. *Psicometria* si propone di fornire agli studenti un'introduzione all'analisi dei dati in psicologia. Le conoscenze/competenze che verranno sviluppate in questo insegnamento sono quelle della Data science, ovvero un insieme di conoscenze/competenze che si pongono all'intersezione tra statistica (ovvero, richiedono la capacità di comprendere teoremi statistici) e informatica (ovvero, richiedono la capacità di sapere utilizzare un software).

La psicologia e la Data science

Sembra sensato spendere due parole su un tema che è importante per gli studenti: quello indicato dal titolo di questo Capitolo. È ovvio che agli studenti di psicologia la statistica non piace. Se piacesse, forse studierebbero Data science e non psicologia; ma non lo fanno. Di conseguenza, gli studenti di psicologia si chiedono: "perché dobbiamo perdere tanto tempo a studiare queste cose quando in realtà quello che ci interessa è tutt'altro?" Questa è una bella domanda.

C'è una ragione molto semplice che dovrebbe farci capire perché la Data science è così importante per la psicologia. Infatti, a ben pensarci, la psicologia è una disciplina intrinsecamente statistica, se per statistica intendiamo quella disciplina che studia la variazione delle caratteristiche degli individui nella popolazione. La psicologia studia *gli individui* ed è proprio la variabilità inter- e intra-individuale ciò che vogliamo descrivere e, in certi casi, predire. In questo senso, la psicologia è molto diversa dall'ingegneria, per esempio. Le proprietà di un determinato ponte sotto certe condizioni, ad esempio, sono molto simili a quelle di un altro pon-

te, sotto le medesime condizioni. Quindi, per un ingegnere la statistica è poco importante: le proprietà dei materiali sono unicamente dipendenti dalla loro composizione e restano costanti. Ma lo stesso non può dirsi degli individui: ogni individuo è unico e cambia nel tempo. E le variazioni tra gli individui, e di un individuo nel tempo, sono l'oggetto di studio proprio della psicologia: è dunque chiaro che i problemi che la psicologia si pone sono molto diversi da quelli affrontati, per esempio, dagli ingegneri. Questa è la ragione per cui abbiamo tanto bisogno della Data science in psicologia: perché la Data science ci consente di descrivere la variazione e il cambiamento. E queste sono appunto le caratteristiche di base dei fenomeni psicologici.

Sono sicuro che, leggendo queste righe, a molti studenti sarà venuta in mente la seguente domanda: perché non chiediamo a qualche esperto di fare il "lavoro sporco" (ovvero le analisi statistiche) per noi, mentre noi (gli psicologi) ci occupiamo solo di ciò che ci interessa, ovvero dei problemi psicologici slegati dai dettagli "tecnici" della Data science? La risposta a questa domanda è che non è possibile progettare uno studio psicologico sensato senza avere almeno una comprensione rudimentale della Data science. Le tematiche della Data science non possono essere ignorate né dai ricercatori in psicologia né da coloro che svolgono la professione di psicologo al di fuori dell'Università. Infatti, anche i professionisti al di fuori dall'università non possono fare a meno di leggere la letteratura psicologica più recente: il continuo aggiornamento delle conoscenze è infatti richiesto dalla deontologia della professione. Ma per potere fare questo è necessario conoscere un bel po' di Data science! Basta aprire a caso una rivista specialistica di psicologia per rendersi conto di quanto ciò sia vero: gli articoli che riportano i risultati delle ricerche psicologiche sono zeppi di analisi statistiche e di modelli formali. E la comprensione della letteratura psicologica rappresenta un requisito minimo nel bagaglio professionale dello psicologo.

Le considerazioni precedenti cercano di chiarire il seguente punto: la Data science non è qualcosa da studiare a malincuore, in un singolo insegnamento universitario, per poi poterla tranquillamente dimenticare. Nel bene e nel male, gli psicologi usano gli strumenti della Data science in tantissimi ambiti della loro attività professionale: in particolare quando costruiscono, somministrano e interpretano i test psicometrici. È dunque chiaro che possedere delle solide basi di Data science è un tassello imprescindibile del bagaglio professionale dello psicologo. In questo insegnamento verranno trattati i temi base della Data science e verrà

adottato un punto di vista bayesiano, che corrisponde all'approccio più recente e sempre più diffuso in psicologia.

Come studiare

Il giusto metodo di studio per prepararsi all'esame di Psicometria è quello di seguire attivamente le lezioni, assimilare i concetti via via che essi vengono presentati e verificare in autonomia le procedure presentate a lezione. Incoraggio gli studenti a farmi domande per chiarire ciò che non è stato capito appieno. Incoraggio gli studenti a utilizzare i forum attivi su Moodle e, soprattutto, a svolgere gli esercizi proposti su Moodle. I problemi forniti su Moodle rappresentano il livello di difficoltà richiesto per superare l'esame e consentono allo studente di comprendere se le competenze sviluppate fino a quel punto sono sufficienti rispetto alle richieste dell'esame.

La prima fase dello studio, che è sicuramente individuale, è quella in cui è necessario acquisire le conoscenze teoriche relative ai problemi che saranno presentati all'esame. La seconda fase di studio, che può essere facilitata da scambi con altri e da incontri di gruppo, porta ad acquisire la capacità di applicare le conoscenze: è necessario capire come usare un software (R) per applicare i concetti statistici alla specifica situazione del problema che si vuole risolvere. Le due fasi non sono però separate: il saper fare molto spesso ci aiuta a capire meglio.

Sviluppare un metodo di studio efficace

Avendo insegnato molte volte in passato un corso introduttivo di analisi dei dati ho notato nel corso degli anni che gli studenti con l'atteggiamento mentale che descriverò qui sotto generalmente ottengono ottimi risultati. Alcuni studenti sviluppano naturalmente questo approccio allo studio, ma altri hanno bisogno di fare uno sforzo per maturarlo. Fornisco qui sotto una breve descrizione del "metodo di studio" che, nella mia esperienza, è il più efficace per affrontare le richieste di questo insegnamento.

- Dedicate un tempo sufficiente al materiale di base, apparentemente facile; assicuratevi di averlo capito bene. Cercate le lacune nella vostra comprensione. Leggere presentazioni diverse dello stesso materiale (in libri o articoli diversi) può fornire nuove intuizioni.
- Gli errori che facciamo sono i nostri migliori maestri. Istintivamente cerchiamo di dimenticare subito i nostri errori. Ma il miglior modo di imparare è apprendere dagli errori che commettiamo. In questo senso, una soluzione corretta è meno utile di una soluzione sbagliata. Quando commettiamo un errore questo ci fornisce un'informazione importante: ci fa capire qual è il materiale di studio sul quale dobbiamo ritornare e che dobbiamo capire meglio.
- C'è ovviamente un aspetto "psicologico" nello studio. Quando un esercizio o problema ci sembra incomprensibile, la cosa migliore da fare è dire: "mi arrendo", "non ho idea di cosa fare!". Questo ci rilassa: ci siamo già arresi, quindi non abbiamo niente da perdere, non dobbiamo più preoccuparci. Ma non dobbiamo fermarci qui. Le cose "migliori" che faccio (se ci sono) le faccio quando non ho voglia di lavorare. Alle volte, quando c'è qualcosa che non so fare e non ho idea di come affontare, mi dico: "oggi non ho proprio voglia di fare fatica", non ho voglia di mettermi nello stato mentale per cui "in 10 minuti devo risolvere il problema perché dopo devo fare altre cose". Però ho voglia di *divertirmi* con quel problema e allora mi dedico a qualche aspetto "marginale" del problema, che so come affrontare, oppure considero l'aspetto più difficile del problema, quello che non so come risolvere, ma invece di cercare di risolverlo, guardo come altre persone hanno affrontato problemi simili, opppure lo stesso problema in un altro contesto. Non mi pongo l'obiettivo "risolvi il problema in 10 minuti", ma invece quello di farmi un'idea "generale" del problema, o quello di capire un caso più specifico e più semplice del problema. Senza nessuna pressione. Infatti, in quel momento ho deciso di non lavorare (ovvero, di non fare fatica). Va benissimo se "parto per la tangente", ovvero se mi metto a leggere del materiale che sembra avere poco a che fare con il problema centrale (le nostre intuizioni e la nostra curiosità solitamente ci indirizzano sulla strada giusta). Quando faccio così, molto spesso trovo la soluzione del problema che mi ero posto e, paradossalmente, la trovo in un tempo minore di quello che, in precedenza, avevo dedicato a "lavorare" al problema. Allora perché non faccio sempre così? C'è ovviamente l'aspetto dei "10 minuti" che non è sempre facile da dimenticare. Sotto pressione, possiamo solo agire in maniera automatica, ovvero possia-

mo solo applicare qualcosa che già sappiamo fare. Ma se dobbiamo imparare qualcosa di nuovo, la pressione è un impedimento.

- È utile farsi da soli delle domande sugli argomenti trattati, senza limitarsi a cercare di risolvere gli esercizi che vengono assegnati. Quando studio qualcosa mi viene in mente: "se questo è vero, allora deve succedere quest'altra cosa". Allora verifico se questo è vero, di solito con una simulazione. Se i risultati della simulazione sono quelli che mi aspetto, allora vuol dire che ho capito. Se i risultati sono diversi da quelli che mi aspettavo, allora mi rendo conto di non avere capito e ritorno indietro a studiare con più attenzione la teoria che pensavo di avere capito – e ovviamente mi rendo conto che c'era un aspetto che avevo frainteso. Questo tipo di verifica è qualcosa che dobbiamo fare da soli, in prima persona: nessun altro può fare questo al posto nostro.
- Non aspettatevi di capire tutto la prima volta che incontrate un argomento nuovo.[1] È utile farsi una nota mentalmente delle lacune nella vostra comprensione e tornare su di esse in seguito per carcare di colmarle. L'atteggiamento naturale, quando non capiamo i dettagli di qualcosa, è quello di pensare: "non importa, ho capito in maniera approssimativa questo punto, non devo preoccuparmi del resto". Ma in realtà non è vero: se la nostra comprensione è superficiale, quando il problema verrà presentato in una nuova forma, non riusciremo a risolverlo. Per cui i dubbi che ci vengono quando studiamo qualcosa sono il nostro alleato più prezioso: ci dicono esattamente quali sono gli aspetti che dobbiamo approfondire per potere migliorare la nostra preparazione.
- È utile sviluppare una visione d'insieme degli argomenti trattati, capire l'obiettivo generale che si vuole raggiungere e avere chiaro il contributo che i vari pezzi di informazione forniscono al raggiungimento di tale obiettivo. Questa organizzazione mentale del materiale di studio facilita la comprensione. È estremamente utile creare degli schemi di ciò che si sta studiando. Non aspettate che sia io a fornirvi un riepilogo di ciò che dovete imparare: sviluppate da soli tali schemi e tali riassunti.
- Tutti noi dobbiamo imparare l'arte di trovare le informazioni, non solo nel caso di questo insegnamento. Quando vi trovate di fronte a qualcosa che non capite, o ottenete un oscuro messaggio di errore da

[1] Ricordatevi inoltre che gli individui tendono a sottostimare la propria capacità di apprendere (Horn and Loewenstein, 2021).

un software, ricordatevi: “Google is your friend”!

Corrado Caudek
Marzo 2022

Inferenza statistica bayesiana

1

Inferenza bayesiana

La moderna statistica bayesiana viene per lo più eseguita utilizzando un linguaggio di programmazione probabilistico implementato su computer. Ciò ha cambiato radicalmente il modo in cui venivano eseguite le statistiche bayesiane anche fin pochi decenni fa. La complessità dei modelli che possiamo costruire è aumentata e la barriera delle competenze matematiche e computazionali che sono richieste è diminuita. Inoltre, il processo di modellazione iterativa è diventato, sotto molti aspetti, molto più facile da eseguire. Anche se formulare modelli statistici complessi è diventato più facile che mai, la statistica è un campo pieno di sottigliezze che non scompaiono magicamente utilizzando potenti metodi computazionali. Pertanto, avere una buona preparazione sugli aspetti teorici, specialmente quelli rilevanti nella pratica, è estremamente utile per applicare efficacemente i metodi statistici.

1.1 Modellizzazione bayesiana

Seguendo (Martin et al., 2022), possiamo descrivere il processo della modellazione bayesiana distinguendo 3 passaggi.

1. Dati alcuni dati e alcune ipotesi su come questi dati potrebbero essere stati generati, progettiamo un modello combinando e trasformando variabili casuali.
2. Usiamo il teorema di Bayes per condizionare i nostri modelli ai dati disponibili. Chiamiamo questo processo "inferenza" e come risultato otteniamo una distribuzione a posteriori. Ci auguriamo che i dati riducano l'incertezza per i possibili valori dei parametri, sebbene questo non sia garantito per nessun modello bayesiano.

3. Critichiamo il modello verificando se il modello abbia senso utilizzando criteri diversi, inclusi i dati e la nostra conoscenza del dominio. Poiché generalmente siamo incerti sui modelli stessi, a volte confrontiamo diversi modelli.

Questi 3 passaggi vengono eseguiti in modo iterativo e danno luogo a quello che si chiama un "flusso di lavoro bayesiano" (*bayesian workflow*).

Osservazione. Un modello è uno strumento concettuale che viene utilizzato per risolvere uno specifico problema. In quanto tale, è generalmente più conveniente parlare dell'adeguatezza del modello a un dato problema che di determinare la sua intrinseca correttezza. I modelli esistono esclusivamente come l'ausilio per il raggiungimento di un qualche ulteriore obiettivo. Il problema che i modelli bayesiani cercano di risolvere è quello dell'inferenza[1].

I modelli bayesiani, computazionali o meno, hanno due caratteristiche distintive:

- Le quantità incognite sono descritte utilizzando le distribuzioni di probabilità. Queste quantità incognite sono chiatame parametri.
- Il teorema di Bayes viene utilizzato per aggiornare i valori dei parametri condizionati ai dati. Possiamo anche concepire questo processo come una riallocazione delle probabilità.

1.2 Inferenza bayesiana come un problema inverso

In questo capitolo ci focalizzeremo sul passaggio 2 descritto sopra. Nello specifico, descrivemo in dettaglio il significato dei tre i termini a destra del segno di uguale nella formula di Bayes: la distribuzione a priori e la funzione di verosimiglianza al numeratore, e la verosimiglianza marginale al denominatore.

[1] In termini colloquiali, l'inferenza può essere descritta come la capacità di giungere a conclusioni basate su evidenze e ragionamenti. L'inferenza bayesiana è una particolare forma di inferenza statistica basata sulla combinazione di distribuzioni di probabilità che ha il fine di ottenere altre distribuzioni di probabilità. Nello specifico, la regola di Bayes ci fornisce un metodo per giungere alla quantificazione della plausibilità di una teoria alla luce dei dati osservati.

1.2.1 Notazione

Per fissare la notazione, nel seguito y rappresenterà i dati e θ rappresenterà i parametri incogniti di un modello statistico. Sia y che θ saranno concepiti come delle variabili casuali.[2] Con x verranno invece denotate le quantità note, come ad esempio i predittori del modello lineare. Per rappresentare in un modo conciso i modelli probabilistici viene usata una notazione particolare. Ad esempio, invece di scrivere $p(\theta) = \text{Beta}(1, 1)$ scriviamo $\theta \sim \text{Beta}(1, 1)$. Il simbolo "$\sim$" viene spesso letto "è distribuito come". Possiamo anche pensare che significhi che θ costituisce un campione casuale estratto dalla distribuzione Beta(1, 1). Allo stesso modo, ad esempio, la verosimiglianza del modello binomiale può essere scritta come $y \sim \text{Bin}(n, \theta)$.

1.2.2 Funzioni di probabilità

Una caratteristica attraente della statistica bayesiana è che la nostra credenza "a posteriori" viene sempre descritta mediante una distribuzione. Questo fatto ci consente di fare affermazioni probabilistiche sui parametri, come ad esempio: "la probabilità che un parametro sia positivo è 0.35"; oppure, "il valore più probabile di θ è 12 e abbiamo probabilità del 50% che θ sia compreso tra 10 e 15". Inoltre, possiamo pensare alla distribuzione a posteriori come alla logica conseguenza della combinazione di un modello con i dati; quindi, abbiamo la garanzia che le affermazioni probabilistiche associate alla distribuzione a posteriori siano matematicamente coerenti. Dobbiamo solo ricordare che tutte queste belle proprietà matematiche sono valide solo nel mondo platonico delle idee dove esistono oggetti matematici come sfere, distribuzioni gaussiane e catene di Markov. Quando passiamo dalla purezza della matematica al disordine della matematica applicata al mondo reale, dobbiamo sempre tenere a mente che i nostri risultati sono condizionati, non solo dai dati, ma anche dai modelli. Di conseguenza, dati errati e/o modelli errati conducono facilmente a conclusioni prive di senso, anche se matematica-

[2] Nell'approccio bayesiano si fa riferimento ad un modello probabilistico $f(y \mid \theta)$ rappresentativo del fenomeno d'interesse noto a meno del valore assunto dal parametro (o dei parametri) che lo caratterizza. Si fa inoltre riferimento ad una distribuzione congiunta (di massa o di densità di probabilità) $f(y, \theta)$. Entrambi gli argomenti della funzione y e θ hanno natura di variabili casuali, laddove la nostra incertezza relativa a y è dovuta alla naturale variabilità del fenomeno indagato (*variabilità aleatoria*), mentre la nostra incertezza relativa a θ è dovuta alla mancata conoscenza del suo valore numerico (*variabilità epistemica*).

mente coerenti. È dunque necessario conservare sempre una sana quota di scetticismo relativamente ai nostri dati, modelli e risultati (Martin et al., 2022).

Avendo detto questo, nell'aggiornamento bayesiano (dai dati ai parametri) vengono utilizzate le seguenti distribuzioni di probabilità (o di massa di probabilità):

- la *distribuzione a priori* $p(\theta)$ — la credenza iniziale (prima di avere osservato i dati $Y = y$) riguardo a θ;
- la *funzione di verosimiglianza* $p(y \mid \theta)$ — quanto sono compatibili i dati osservati $Y = y$ con i diversi valori possibili di θ?
- la *verosimiglianza marginale* $p(y)$ — costante di normalizzazione: qual è la probabilità complessiva di osservare i dati $Y = y$? In termini formali:

$$p(y) = \int_\theta p(y, \theta) \, \mathrm{d}\theta = \int_\theta p(y \mid \theta) p(\theta) \, \mathrm{d}\theta.$$

- la *distribuzione a posteriori* $p(\theta \mid y)$ — la nuova credenza relativa alla credibilità di ciascun valore θ dopo avere osservato i dati $Y = y$.

1.3 La regola di Bayes

Assumendo un modello statistico, la formula di Bayes consente di giungere alla distribuzione a posteriori $p(\theta \mid y)$ per il parametro di interesse θ, come indicato dalla seguente catena di equazioni[3]:

[3] In realtà, avremmo dovuto scrivere $p(\theta \mid y, \mathcal{M})$, in quanto non condizioniamo la stima di θ solo rispetto ai dati y ma anche ad un modello probabilistico $\mathcal{M}$ che viene assunto quale meccanismo generatore dei dati. Per semplicità di notazione, omettiamo il riferimento a $\mathcal{M}$.

$$
\begin{aligned}
p(\theta \mid y) &= \frac{p(\theta, y)}{p(y)} && \text{[def. prob. condizionata]} && (1.1)\\
&= \frac{p(y \mid \theta)\, p(\theta)}{p(y)} && \text{[legge prob. composta]} && (1.2)\\
&= \frac{p(y \mid \theta)\, p(\theta)}{\int_\Theta p(y, \theta)\ \mathrm{d}\theta} && \text{[legge prob. totale]} && (1.3)\\
&= \frac{p(y \mid \theta)\, p(\theta)}{\int_\Theta p(y \mid \theta)\, p(\theta)\ \mathrm{d}\theta} && \text{[legge prob. composta]} && (1.4)\\
&\propto p(y \mid \theta)\, p(\theta) && && (1.5)
\end{aligned}
$$

La regola di Bayes "inverte" la probabilità della distribuzione a posteriori $p(\theta \mid y)$, esprimendola nei termini della funzione di verosimiglianza $p(y \mid \theta)$ e della distribuzione a priori $p(\theta)$. L'ultimo passo è importante per la stima della distribuzione a posteriori mediante i metodi Monte Carlo a catena di Markov, in quanto per questi metodi richiedono soltanto che le funzioni di probabilità siano definite a meno di una costante di proporzionalità. In altri termini, per la maggior parte degli scopi dell'inferenza inversa, è sufficiente calcolare la densità a posteriori non normalizzata, ovvero è possibile ignorare il denominatore bayesiano $p(y)$. La distribuzione a posteriori non normalizzata, dunque, si riduce al prodotto della varosimiglianza e della distribuzione a priori.

Possiamo dire che la regola di Bayes viene usata per aggiornare le credenze a priori su θ (ovvero, la distribuzione a priori) in modo tale da produrre le nuove credenze a posteriori $p(\theta \mid y)$ che combinano le informazioni fornite dai dati y con le credenze precedenti. La distribuzione a posteriori riflette dunque l'aggiornamento delle credenze del ricercatore alla luce dei dati. La distribuzione a posteriori $p(\theta \mid y)$ contiene tutta l'informazione riguardante il parametro θ e viene utilizzata per produrre indicatori sintetici, per la determinazione di stime puntuali o intervallari, e per la verifica d'ipotesi.

La (1.5) rende evidente che, in ottica bayesiana, la quantità di interesse θ non è fissata (come nell'impostazione frequentista), ma è una variabile casuale la cui distribuzione di probabilità è influenzata sia dalle informazioni a priori sia dai dati a disposizione. In altre parole, nell'approccio bayesiano non esiste un valore vero di θ, ma invece lo scopo è quello di fornire invece un giudizio di probabilità (o di formulare una "previsione", nel linguaggio di de Finetti). Prima delle osservazioni, sulla base delle

nostre conoscenze assegnamo a θ una distribuzione a priori di probabilità. Dopo le osservazioni, correggiamo il nostro giudizio e assegniamo a θ una distribuzione a posteriori di probabilità.

1.3.1 Un esempio di aggiornamento bayesiano

Per descrivere l'aggiornamento bayesiano, in questo Capitolo (così come nei successivi) considereremo i dati di Zetsche et al. (2019). Questi ricercatori si sono chiesti se gli individui depressi manifestino delle aspettative accurate circa il loro umore futuro, oppure se tali aspettative siano distorte negativamente. Esamineremo qui i 30 partecipanti dello studio di Zetsche et al. (2019) che hanno riportato la presenza di un episodio di depressione maggiore in atto. All'inizio della settimana di test, a questi pazienti è stato chiesto di valutare l'umore che si aspettavano di esperire nei giorni seguenti della settimana. Mediante una app, i partecipanti dovevano poi valutare il proprio umore in cinque momenti diversi di ciascuno dei cinque giorni successivi. Lo studio considera diverse emozioni, ma qui ci concentriamo solo sulla tristezza.

Sulla base dei dati forniti dagli autori, abbiamo calcolato la media dei giudizi relativi al livello di tristezza raccolti da ciascun partecipante tramite la app. Tale media è stata poi sottratta dall'aspettativa del livello di tristezza fornita all'inizio della settimana. La discrepanza tra aspettative e realtà è stata considerata come un evento dicotomico: valori positivi di tale differenza indicano che le aspettative circa il livello di tristezza erano maggiori del livello di tristezza effettivamente esperito — ciò significa che le aspettative future risultano negativamente distorte (evento codificato con "1"). Viceversa, si ha che le aspettative risultano positivamente distorte se la differenza descritta in precedenza assume un valore negativo (evento codificato con "0").

Nel campione dei 30 partecipanti clinici di Zetsche et al. (2019), le aspettative future di 23 partecipanti risultano distorte negativamente e quelle di 7 partecipanti risultano distorte positivamente. Chiameremo θ la probabilità dell'evento "le aspettative del partecipante sono distorte negativamente". Ci poniamo il problema di ottenere una stima a posteriori di θ avendo osservato 23 "successi" in 30 prove.[4]

[4]Si noti un punto importante: dire semplicemente che la stima di θ è uguale a $23/30 = 0.77$ ci porta ad ignorare il livello di incertezza associato a tale stima. Infatti, lo stesso valore (0.77) si può ottenere come 23/30, o 230/300, o 2300/3000, o 23000/30000, ma l'incertezza di una stima pari a 0.77 è molto diversa nei quattro

1.4 Modello probabilistico

Nel caso dello studio di Zetsche et al. (2019), i dati qui considerati possono essere considerati la manifestazione di una variabile casuale Bernoulliana – 23 "successi" in 30 prove. Se i dati rappresentano una proporzione, allora possiamo adottare un modello probabilistico binomiale quale meccanismo generatore dei dati:

$$y \sim \text{Bin}(n, \theta), \tag{1.6}$$

laddove θ è la probabiltà che una prova Bernoulliana assuma il valore 1 e n corrisponde al numero di prove Bernoulliane. Questo modello assume che le prove Bernoulliane y_i che costituiscono il campione y siano tra loro indipendenti e che ciascuna abbia la stessa probabilità $\theta \in [0, 1]$ di essere un "successo" (valore 1). In altre parole, il modello generatore dei dati avrà una funzione di massa di probabilità

$$p(y \mid \theta) \;=\; \text{Bin}(y \mid n, \theta).$$

Nei capitoli precedenti è stato mostrato come, sulla base del modello binomiale, sia possibile assegnare una probabilità a ciascun possibile valore $y \in \{0, 1, \dots, n\}$ *assumendo noto il valore del parametro* θ. Ma ora abbiamo il problema inverso, ovvero quello di fare inferenza su θ alla luce dei dati campionari y. In altre parole, riteniamo di conoscere il modello probabilistico che ha generato i dati, ma di tale modello non conosciamo i parametri: vogliamo dunque ottenere informazioni su θ avendo osservato i dati y.

Nel modello probabilistico che stiamo esaminando, il termine n viene trattato come una costante nota e θ come una *variabile casuale*. Dato che θ è incognito, ma abbiamo a disposione i dati y, svolgeremo l'inferenza su θ mediante la regola di Bayes per determinare la distribuzione a posteriori $p(\theta \mid y)$.

casi. Quando si traggono conclusioni dai dati è invece necessario quantificare il livello della nostra incertezza relativamente alla stima del parametro di interesse (nel caso presente, θ). Lo strumento ci consente di quantificare tale incertezza è la distribizione a posteriori $p(\theta \mid y)$. Ovviamente, $p(\theta \mid y)$ assume forme molto diverse nei quattro casi descritti sopra.

Osservazione. Si noti che il modello probabilistico (1.6) non spiega perché, in ciascuna realizzazione, Y assuma un particolare valore. Questo modello deve piuttosto essere inteso come un costrutto matematico che ha lo scopo di riflettere alcune proprietà del processo corrispondente ad una sequenza di prove Bernoulliane. Una parte del lavoro della ricerca in tutte le scienze consiste nel verificare le assunzioni dei modelli e, se necessario, nel migliorare i modelli dei fenomeni considerati. Un modello viene giudicato in relazione al suo obiettivo. Se l'obiettivo del modello molto semplice che stiamo discutendo è quello di prevedere la proporzione di casi nei quali $y_i = 1$, $i = 1, \dots, n$, allora un modello con un solo parametro come quello che abbiamo introdotto sopra può essere sufficiente. Ma l'evento $y_i = 1$ (supponiamo: superare l'esame di Psicometria, oppure risultare positivi al COVID-19) dipende da molti fattori e se vogliamo rendere conto di una tale complessità, un modello come quello che stiamo discutendo qui certamente non sarà sufficiente. In altre parole, modelli sempre migliori vengono proposti, laddove ogni successivo modello è migliore di quello precedente in quanto ne migliora le capacità di previsione, è più generale, o è più elegante. Per concludere, un modello è un costrutto matematico il cui scopo è quello di rappresentare un qualche aspetto della realtà. Il valore di un tale strumento dipende dalla sua capacità di ottenere lo scopo per cui è stato costruito.

1.5 Distribuzioni a priori

Quando adottiamo un approccio bayesiano, i parametri della distribuzione di riferimento non venono considerati come delle costanti incognite ma bensì vengono trattati come variabili casuali e, di conseguenza, i parametri assumono una particolare distribuzione che nelle statistica bayesiana viene definita come "a priori". I parametri (o il parametro), che possiamo indicare con θ, possono assumere delle distribuzioni a priori differenti; a seconda delle informazioni disponibili bisogna cercare di assegnare una distribuzione di θ in modo tale che venga assegnata una probabilità maggiore a quei valori che si ritengono più plausibili per θ.

La distribuzione a priori sui valori dei parametri $p(\theta)$ è parte integrante del modello statistico. Ciò implica che due modelli bayesiani possono condividere la stessa funzione di verosimiglianza, ma tuttavia devono essere considerati come modelli diversi se specificano diverse distribuzioni

a priori. Ciò significa che, quando diciamo "Modello binomiale", intendiamo in realtà un'intera classe di modelli, ovvero tutti i possibili modelli che hanno la stessa verosimiglianza ma diverse distribuzioni a priori su θ.

Nell'analisi dei dati bayesiana, la distribuzione a priori $p(\theta)$ codifica le credenze del ricercatore a proposito dei valori dei parametri, prima di avere osservato i dati. Idealmente, le credenze a priori che supportano la specificazione di una distribuzione a priori dovrebbero essere supportate da una qualche motivazione, come ad esempio i risultati di ricerche precedenti, o altre motivazioni giustificabili.

Quando una nuova osservazione (p. es., vedo un cigno bianco) corrisponde alle mie credenze precedenti (p. es., la maggior parte dei cigni sono bianchi) la nuova osservazione rafforza le mie credenze precedenti: più nuove osservazioni raccolgo (p. es., più cigni bianchi vedo), più forti diventano le mie credenze precedenti. Tuttavia, quando una nuova osservazione (p. es., vedo un cigno nero) non corrisponde alle mie credenze precedenti, ciò contribuisce a diminuire la certezza che attribuisco alle mie credenze: tanto maggiori diventano le osservazioni non corrispondenti alle mie credenze (p. es., più cigni neri vedo), tanto più si indeboliscono le mie credenze. Fondamentalmente, tanto più forti sono le mie credenze precedenti, di tante più osservazioni incompatibili (ad esempio, cigni neri) ho bisogno per cambiare idea.

Pertanto, da una prospettiva bayesiana, l'incertezza intorno ai parametri di un modello *dopo* aver visto i dati (ovvero le distribuzioni a posteriori) deve includere anche le credenze precedenti. Se questo modo di ragionare vi sembra molto intuitivo, non è una coincidenza: vi sono infatti diverse teorie psicologiche che prendono l'aggiornamento bayesiano come modello di funzionamento di diversi processi cognitivi.

1.5.1 Tipologie di distribuzioni a priori

Possiamo distinguere tra diverse distribuzioni a priori in base a quanto fortemente impegnano il ricercatore a ritenere come plausibile un particolare intervallo di valori dei parametri. Il caso più estremo è quello che rivela una totale assenza di conoscenze a priori, il che conduce alle *distribuzioni a priori non informative*, ovvero quelle che assegnano lo stesso livello di credibilità a tutti i valori dei parametri. Le distribuzioni a priori informative, d'altra parte, possono essere *debolmente informa-*

tive o *fortemente informative*, a seconda della forza della credenza che esprimono. Il caso più estremo di credenza a priori è quello che riassume il punto di vista del ricercatore nei termini di un *unico valore* del parametro, il che assegna tutta la probabilità (massa o densità) su di un singolo valore di un parametro. Poiché questa non è più una distribuzione di probabilità, sebbene ne soddisfi la definizione, in questo caso si parla di una *distribuzione a priori degenerata*.

La figura seguente mostra esempi di distribuzioni a priori non informative, debolmente o fortemente informative, così come una distribuzione a priori espressa nei termini di un valore puntuale per il modello Binomiale. Le distribuzione a priori illustrate di seguito sono le seguenti:

- *non informativa* : $\theta_c \sim \text{Beta}(1, 1)$;
- *debolmente informativa* : $\theta_c \sim \text{Beta}(5, 2)$;
- *fortemente informativa* : $\theta_c \sim \text{Beta}(50, 20)$;
- *valore puntuale* : $\theta_c \sim \text{Beta}(\alpha, \beta)$ con $\alpha, \beta \to \infty$ e $\frac{\alpha}{\beta} = \frac{5}{2}$.

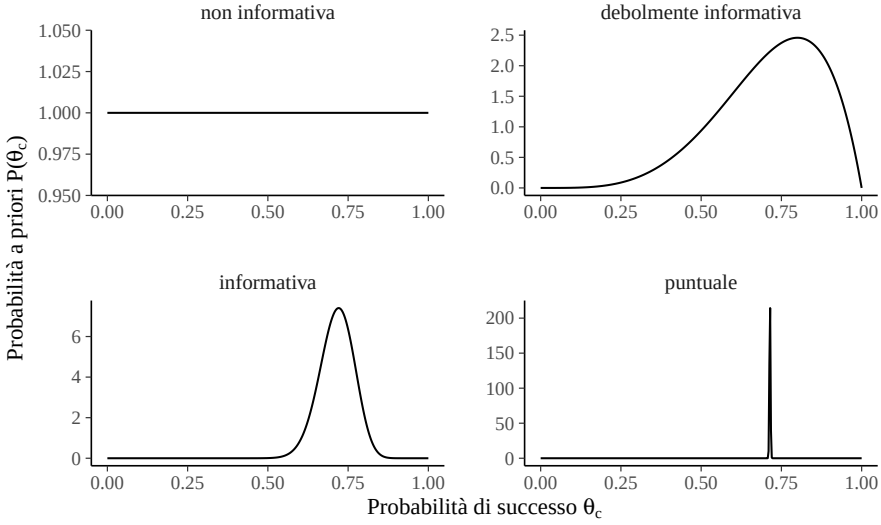


Figura 1.1: *Esempi di distribuzioni a priori per il parametro θ_c nel Modello Binomiale.*

1.5.2 Selezione della distribuzione a priori

La selezione delle distribuzioni a priori è stata spesso vista come una delle scelte più importanti che un ricercatore fa quando implementa un

modello bayesiano in quanto può avere un impatto sostanziale sui risultati finali. La soggettività delle distribuzioni a priori è evidenziata dai critici come un potenziale svantaggio dei metodi bayesiani. A questa critica, van de Schoot et al. (2021) rispondono dicendo che, al di là della scelta delle distribuzioni a priori, ci sono molti elementi del processo di inferenza statistica che sono soggettivi, ovvero la scelta del modello statistico e le ipotesi sulla distribuzione degli errori. In secondo luogo, van de Schoot et al. (2021) notano come le distribuzioni a priori svolgono due importanti ruoli statistici: quello della "regolarizzazione della stima", ovvero, il processo che porta ad indebolire l'influenza indebita di osservazioni estreme, e quello del miglioramento dell'efficienza della stima, ovvero, la facilitazione dei processi di calcolo numerico di stima della distribuzione a posteriori. L'effetto della distribuzione a priori sulla distribuzione a posteriori verrà discusso nel Capitolo **??**.

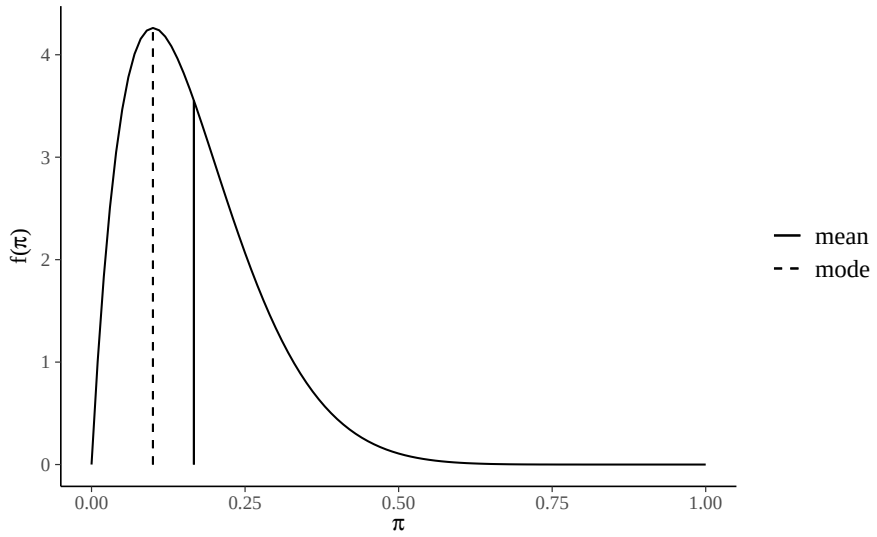
1.5.3 La distribuzione a priori per i dati di Zetsche et al. (2019)

In un problema concreto di analisi dei dati, la scelta della distribuzione a priori dipende dalle credenze a priori che vogliamo includere nell'analisi dei dati. Se non abbiamo alcuna informazione a priori, potremmo pensare di usare una distribuzione a priori uniforme, ovvero una Beta di parametri $\alpha = 1$ e $\beta = 1$. Questa, tuttavia, è una cattiva idea perché il risultato ottenuto non è invariante a seconda della trasformazione della scala dei dati (ad esempio, se esprimiamo l'altezza in cm piuttosto che in m). Il problema della *riparametrizzazione* verrà discusso nel Capitolo ?? **TODO**. È invece raccomandato usare una distribuzione a priori poco informativa, come ad esempio Beta$(2, 2)$.

Nella presente discussione, per fare un esempio, quale distribuzione a priori useremo una Beta$(2, 10)$, ovvero:

$$p(\theta) = \frac{\Gamma(12)}{\Gamma(2)\Gamma(10)}\theta^{2-1}(1-\theta)^{10-1}.$$

```
bayesrules::plot_beta(alpha = 2, beta = 10, mean = TRUE, mode = TRUE)
```



La Beta$(2, 10)$ esprime la credenza che θ assume valori < 0.5, con il valore più plausibile pari a circa 0.1. Questo è assolutamente implausibile, nel caso dell'esempio in discussione. Adotteremo una tale distribuzione a priori solo per scopi didattici, per esplorare le conseguenze di tale scelta (molto più sensato sarebbe stato usare Beta$(2, 2)$).

1.6 Verosimiglianza

Oltre alla distribuzione a priori di θ, nel numeratore della regola di Bayes troviamo la funzione di verosimigliana. Iniziamo dunque con una definizione.

Definizione 1.1. La *funzione di verosimiglianza* $\mathcal{L}(\theta \mid y) = f(y \mid \theta), \theta \in \Theta$, è la funzione di massa o di densità di probabilità dei dati y vista come una funzione del parametro sconosciuto (o dei parametri sconosciuti) θ.

Detto in altre parole, le funzioni di verosimiglianza e di (massa o densità di) probabilità sono formalmente identiche, ma è completamente diversa la loro interpretazione. Nel caso della funzione di massa o di densità di

probabilità la distribuzione del vettore casuale delle osservazioni campionarie y dipende dai valori assunti dal parametro (o dai parametri) θ; nel caso della la funzione di verosimiglianza la credibilità assegnata a ciascun possibile valore θ viene determinata avendo acquisita l'informazione campionaria y che rappresenta l'elemento condizionante. In altri termini, la funzione di verosimiglianza è lo strumento che consente di rispondere alla seguente domanda: avendo osservato i dati y, quanto risultano (relativamente) credibili i diversi valori del parametro θ?

Spesso per indicare la verosimiglianza si scrive $\mathcal{L}(\theta)$ se è chiaro a quali valori y ci si riferisce. La verosimiglianza $\mathcal{L}$ è una curva (in generale, una superficie) nello spazio Θ del parametro (in generale, dei parametri) che riflette la credibilità relativa dei valori θ alla luce dei dati osservati.

Notiamo un punto importante: la funzione $\mathcal{L}(\theta \mid y)$ non è una funzione di densità. Infatti, essa non racchiude un'area unitaria.

In conclusione, la funzione di verosimiglianza descrive in termini relativi il sostegno empirico che $\theta \in \Theta$ riceve da y. Infatti, la funzione di verosimiglianza assume forme diverse al variare di y (lasciamo come esercizio da svolgere la verifica di questa affermazione).

1.6.1 La stima di massima verosimiglianza

La funzione di verosimiglianza rappresenta la "credibilità relativa" dei valori del parametro di interesse. Ma qual è il valore più credibile? Se utilizziamo soltanto la funzione di verosimiglianza, allora la risposta è data dalla stima di massima verosimiglinza.

Definizione 1.2. Un valore di θ che massimizza $\mathcal{L}(\theta \mid y)$ sullo spazio parametrico Θ è detto *stima di massima verosimiglinza* (s.m.v.) di θ ed è indicato con $\hat{\theta}$:

$$\hat{\theta} = \arg\max_{\theta \in \Theta} \mathcal{L}(\theta). \tag{1.7}$$

Il paradigma frequentista utilizza la funzione di verosimiglianza quale unico strumento per giungere alla stima del valore più credibile del parametro sconosciuto θ. Tale stima corrisponde al punto di massimo della funzione di verosimiglianza. In base all'approccio bayesiano, invece, il valore più credibile del parametro sconosciuto θ, anziché alla s.m.v., corrisponde invece alla moda (o media, o mediana) della distribuzione a

posteriori $p(\theta \mid y)$ che si ottiene combinando la verosimiglianza $p(y \mid \theta)$ con la distribuzione a priori $p(\theta)$. Per un approfondimento della stima di massima verosimiglianza si veda l'Appendice **??**.

1.6.2 La log-verosimiglianza

Dal punto di vista pratico risulta più conveniente utilizzare, al posto della funzione di verosimiglianza, il suo logaritmo naturale, ovvero la funzione di log-verosimiglianza:

$$\ell(\theta) = \log \mathcal{L}(\theta). \tag{1.8}$$

Poiché il logaritmo è una funzione strettamente crescente (usualmente si considera il logaritmo naturale), allora $\mathcal{L}(\theta)$ e $\ell(\theta)$ assumono il massimo (o i punti di massimo) in corrispondenza degli stessi valori di θ:

$$\hat{\theta} = \arg\max_{\theta \in \Theta} \ell(\theta) = \arg\max_{\theta \in \Theta} \mathcal{L}(\theta).$$

Per le proprietà del logaritmo, si ha

$$\ell(\theta) = \log \left(\prod_{i=1}^{n} f(y \mid \theta) \right) = \sum_{i=1}^{n} \log f(y \mid \theta). \tag{1.9}$$

Si noti che non è necessario lavorare con i logaritmi, ma è fortemente consigliato. Il motivo è che i valori della verosimiglianza, in cui si moltiplicano valori di probabilità molto piccoli, possono diventare estremamente piccoli – qualcosa come 10^{-34}. In tali circostanze, non è sorprendente che i programmi dei computer mostrino problemi di arrotondamento numerico. Le trasformazioni logaritmiche risolvono questo problema.

Osservazione. Seguendo una pratica comune, in questa dispensa spesso useremo la notazione $p(\cdot)$ per rappresentare due quantità differenti, ovvero la funzione di verosimiglianza e la distribuzione a priori. Questo piccolo abuso di notazione riflette il seguente punto di vista: anche se la verosimiglianza non è una funzione di densità di probabilità, noi non vogliamo stressare questo aspetto, ma vogliamo piuttosto pensare alla verosimiglianza e alla distribuzione a priori come a due elementi che sono egualmente necessari per calcolare la distribuzione a posteriori. In altri termini, per così dire, questa notazione assegna lo stesso status epistemologico alle due diverse quantità che si trovano al numeratore della regola di Bayes.

Esercizio 1.1. Per i dati di Zetsche et al. (2019), ovvero 23 "successi" in 30 prove, si trovi e si interpreti la funzione di verosimiglianza.

Per i dati di Zetsche et al. (2019) la funzione di verosimiglianza corrisponde alla funzione binomiale di parametro $\theta \in [0, 1]$ sconosciuto. Abbiamo osservato un "successo" 23 volte in 30 "prove", dunque, $y = 23$ e $n = 30$. La funzione di verosimiglianza diventa

$$\mathcal{L}(\theta \mid y) = \frac{(23 + 7)!}{23!7!}\theta^{23} + (1 - \theta)^7. \tag{1.10}$$

Per costruire la funzione di verosimiglianza dobbiamo applicare la (1.10) tante volte, cambiando ogni volta il valore θ ma *tenendo sempre costante il valore dei dati*. Per esempio, se poniamo $\theta = 0.1$

$$\mathcal{L}(\theta \mid y) = \frac{(23 + 7)!}{23!7!}0.1^{23} + (1 - 0.1)^7$$

otteniamo

```
dbinom(23, 30, 0.1)
#> [1] 9.737e-18
```

Se poniamo $\theta = 0.2$

$$\mathcal{L}(\theta \mid y) = \frac{(23 + 7)!}{23!7!}0.2^{23} + (1 - 0.2)^7$$

otteniamo

```
dbinom(23, 30, 0.2)
#> [1] 3.581e-11
```

e così via. La figura 1.2 — costruita utilizzando 100 valori equispaziati $\theta \in [0, 1]$ — fornisce una rappresentazione grafica della funzione di verosimiglianza.

```
n <- 30
y <- 23
theta <- seq(0, 1, length.out = 100)
```

```
like <- choose(n, y) * theta^y * (1 - theta)^(n - y)
tibble(theta, like) %>%
  ggplot(aes(x = theta, y = like)) +
  geom_line() +
  labs(
    y = expression(L(theta)),
    x = expression("Valori possibili di" ~ theta)
  )
```

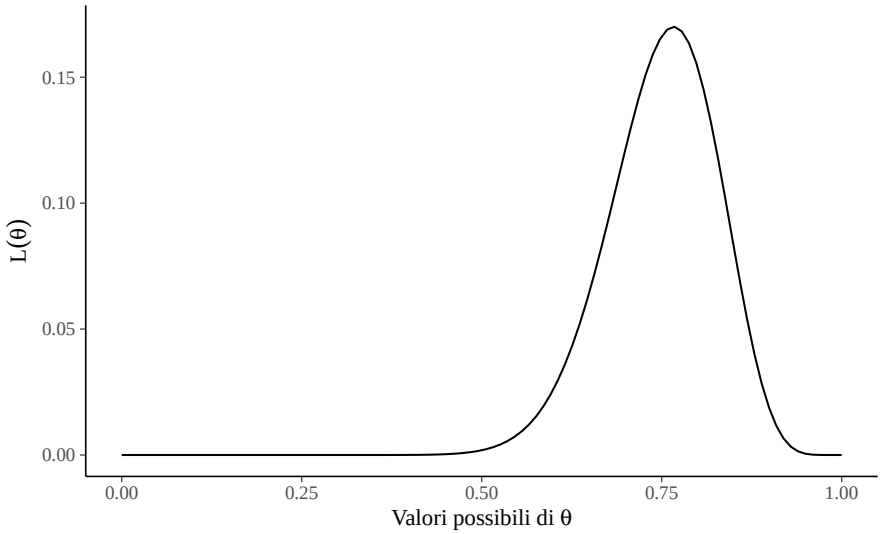


Figura 1.2: *Funzione di verosimiglianza nel caso di 23 successi in 30 prove.*

Come possiamo interpretare la curva che abbiamo ottenuto? Per alcuni valori θ la funzione di verosimiglianza assume valori piccoli; per altri valori θ la funzione di verosimiglianza assume valori più grandi. Questi ultimi sono i valori di θ "più credibili" e il valore 23/30 è il valore più credibile di tutti. La funzione di verosimiglianza di θ valuta la compatibilità dei dati osservati $Y = y$ con i diversi possibili valori θ. In termini più formali possiamo dire che la funzione di verosimiglianza ha la seguente interpretazione: sulla base dei dati, $\theta_1 \in \Theta$ è più credibile di $\theta_2 \in \Theta$ come indice del modello probabilistico generatore delle osservazioni se $\mathcal{L}(\theta_1) > \mathcal{L}(\theta_1)$.

1.7 La verosimiglianza marginale

Per il calcolo di $p(\theta \mid y)$ è necessario dividere il prodotto tra la distribuzione a priori e la verosimiglianza per una costante di normalizzazione. Tale costante di normalizzazione, detta *verosimiglianza marginale*, ha lo scopo di fare in modo che $p(\theta \mid y)$ abbia area unitaria.

Si noti che il denominatore della regola di Bayes (ovvero la verosimiglianza marginale) è sempre espresso nei termini di un integrale. Tranne in pochi casi particolari, tale integrale non ha una soluzione analitica. Per questa ragione, l'inferenza bayesiana procede calcolando una approssimazione della distribuzione a posteriori mediante metodi numerici.

Esercizio 1.2. Si trovi la verosimiglianza maginale per i dati di Zetsche et al. (2019).

Supponiamo che nel numeratore bayesiano la verosimiglianza sia moltiplicata per una distribuzione uniforme, Beta$(1, 1)$. In questo caso, il prodotto si riduce alla funzione di verosimiglianza. In riferimento ai dati di Zetsche et al. (2019), la costante di normalizzazione per si ottiene semplicemente marginalizzando la funzione di verosimiglianza $p(y = 23, n = 30 \mid \theta)$ sopra θ, ovvero risolvendo l'integrale:

$$p(y = 23, n = 30) = \int_0^1 \binom{30}{23} \theta^{23} (1 - \theta)^7 \, \mathrm{d}\theta. \tag{1.11}$$

Una soluzione numerica si trova facilmente usando R:

```
like_bin <- function(theta) {
  choose(30, 23) * theta^23 * (1 - theta)^7
}
integrate(like_bin, lower = 0, upper = 1)$value
#> [1] 0.03226
```

La derivazione analitica della costante di normalizzazione qui discussa è fornita nell'Appendice **??**.

1.8 Distribuzione a posteriori

La distribuzione a postreriori si trova applicando il teorema di Bayes:

$$\text{probabilità a posteriori} = \frac{\text{probabilità a priori} \cdot \text{verosimiglianza}}{\text{costante di normalizzazione}}$$

Ci sono due metodi principali per calcolare la distribuzione a posteriori $p(\theta \mid y)$:

- una precisa derivazione matematica formulata nei termini della distribuzione a priori coniugata alla distribuzione a posteriori (si veda il Capitolo **??**); tale procedura però ha un'applicabilità molto limitata;
- un metodo approssimato, molto facile da utilizzare in pratica, che dipende da metodi Monte Carlo basati su Catena di Markov (MCMC); questo problema verrà discusso nel Capitolo ??

Una volta trovata la distribuzione a posteriori, possiamo usarla per derivare altre quantità di interesse. Questo viene generalmente ottenuto calcolando il valore atteso:

$$J = \int f(\theta) p(\theta \mid y) \, \mathrm{d}y$$

Se $f(\cdot)$ è la funzione identità, ad esempio, J risulta essere la media di θ:

$$\bar{\theta} = \int_{\Theta} \theta p(\theta \mid y) \, \mathrm{d}\theta.$$

1.9 Distribuzione predittiva a priori

La distribuzione a posteriori è l'oggetto centrale nella statistica bayesiana, ma non è l'unico. Oltre a fare inferenze sui valori dei parametri, potremmo voler fare inferenze sui dati. Questo può essere fatto calcolando la *distribuzione predittiva a priori*:

$$p(y^*) = \int_{\Theta} p(y^* \mid \theta) p(\theta) \, \mathrm{d}\theta. \tag{1.12}$$

La (1.12) descrive la distribuzione prevista dei dati in base al modello (che include la distribuzione a priori e la verosimiglianza). Questi sono i dati y^* che ci aspettiamo, dato il modello, prima di avere osservato i dati del campione.

Possiamo utilizzare campioni dalla distribuzione predittiva a priori per valutare e calibrare i modelli utilizzando le nostre conoscenze dominio-specifiche. Ad esempio, ci potremmo chiedere: "È sensato che un modello dell'altezza umana preveda che un essere umano sia alto -1.5 metri?". Già prima di misurare una singola persona, possiamo renderci conto dell'assurdità di questa domanda. Se la distribuzione prevista dei dati consente domande di questo tipo, è chiaro che il modello deve essere riformulato.

Osservazione. Si dice comunemente che l'adozione di una prospettiva probabilistica per la modellazione conduce all'idea che i modelli generano dati. Se i modelli generano dati, possiamo creare modelli adatti per i nostri dati solo pensando a come i dati potrebbero essere stati generati. Inoltre, questa idea non è solo un concetto astratto. Assume una concreta nella forma della distribuzione predittiva a priori. Se la distribuzione predittiva a priori non ha senso, come abbiamo detto sopra, diventa necessario riformulare il modello.

1.10 Distribuzione predittiva a posteriori

Un'altra quantità utile da calcolare è la distribuzione predittiva a posteriori:

$$p(\tilde{y} \mid y) = \int_{\Theta} p(\tilde{y} \mid \theta) p(\theta \mid y) \, \mathrm{d}\theta. \tag{1.13}$$

Questa è la distribuzione dei dati attesi futuri $\tilde{y}$ alla luce della distribuzione a posteriori $p(\theta \mid y)$, che a sua volta è una conseguenza del modello (distribuzione a priori e verosimiglianza) e dei dati osservati. In altre parole, questi sono i dati che il modello si aspetta dopo aver osservato i dati y. Dalla (1.13) possiamo vedere che le previsioni sui dati attesi futuri sono calcolate integrando (o marginalizzando) sulla distribuzione a posteriori dei parametri. Di conseguenza, le previsioni calcolate in que-

sto modo incorporano l'incertezza relativa alla stima dei parametri del modello.

Considerazioni conclusive

Questo Capitolo ha brevemente passato in rassegna alcuni concetti di base dell'inferenza statistica bayesiana. In base all'approccio bayesiano, invece di dire che il parametro di interesse di un modello statistico ha un valore vero ma sconosciuto, diciamo che, prima di eseguire l'esperimento, è possibile assegnare una distribuzione di probabilità, che chiamano stato di credenza, a quello che è il vero valore del parametro. Questa distribuzione a priori può essere nota (per esempio, sappiamo che la distribuzione dei punteggi del QI è normale con media 100 e deviazione standard 15) o può essere del tutto arbitraria. L'inferenza bayesiana procede poi nel modo seguente: si raccolgono alcuni dati e si calcola la probabilità dei possibili valori del parametro alla luce dei dati osservati e delle credenze a priori. Questa nuova distribuzione di probabilità è chiamata "distribuzione a posteriori" e riassume l'incertezza dell'inferenza. I concetti importanti che abbiamo appreso in questo Capitolo sono quelli di distribuzione a priori, verosimiglianza, verosimiglianza marginale e distribuzione a posteriori. Questi sono i concetti fondamentali della statistica bayesiana.

2

Approssimazione della distribuzione a posteriori

In questo Capitolo ci occuperemo di metodi numerici per l'approssimazione della distribuzione a posteriori.

2.1 Stima della distribuzione a posteriori

In un problema bayesiano i dati y provengono da una densità $p(y \mid \theta)$ e al parametro θ viene assegnata una densità a priori $p(\theta)$. Dopo avere osservato un campione $Y = y$, la funzione di verosimiglianza è uguale a $\mathcal{L}(\theta) = p(y \mid \theta)$ e la densità a posteriori diventa

$$p(\theta \mid y) = \frac{p(\theta)\mathcal{L}(\theta)}{\int p(\theta)\mathcal{L}(\theta)d\theta}. \tag{2.1}$$

Si noti che, quando usiamo il teorema di Bayes per calcolare la distribuzione a posteriori del parametro di un modello statistico, al denominatore troviamo un integrale che, nella maggior parte dei casi, non si può risolvere analiticamente. In altre parole: è possibile ottenere analiticamente la distribuzione a posteriori solo per alcune specifiche combinazioni di distribuzioni a priori e verosimiglianza, il che limita considerevolmente la flessibilità della modellizzazione. Per questa ragione, la strada principale che viene seguita nella modellistica bayesiana è quella che porta a determinare la distribuzione a posteriori non per via analitica, ma bensì mediante metodi numerici. La simulazione fornisce dunque la strategia generale del calcolo bayesiano.

Ci sono molte librerie R o Python che consentono di stimare la distribuzione a posteriori con metodi numerici, quindi in generale è molto improbabile che un ricercatore abbia bisogno di codificare un proprio

algoritmo per risolvere questo problema. Ad oggi ci sono solo due buoni motivi per scrivere il codice che ha lo scopo di approssimare la distribuzione a posteriori per via numerica: o si sta progettando un nuovo metodo che sia in grado di migliorare quelli già esistenti (questo è un tipico problema da informatici o ingegneri) o si sta imparando come funzionano i metodi attuali. Dato che il nostro obiettivo è, appunto, quello di imparare, in questo capitolo vedremo come questo problema possa essere affrontato. Nel resto della dispensa useremo invece i metodi già disponibili nelle librerie R.

In questo Capitolo esaminando tre diverse tecniche che possono essere utilizzate per calcolare per via numerica la distribuzione a posteriori:

1. il metodo basato su griglia,
2. il metodo dell'approssimazione quadratica,
3. il metodo di Monte Carlo basato su Catena di Markov (MCMC).

2.2 Metodo basato su griglia

Il metodo basato su griglia (*grid-based*) è un metodo di approsimazione numerica basato su una griglia di punti uniformemente spaziati. Anche se la maggior parte dei parametri è continua (ovvero, in linea di principio ciascun parametro può assumere un numero infinito di valori), possiamo ottenere un'eccellente approssimazione della distribuzione a posteriori considerando solo una griglia finita di valori dei parametri. In un tale metodo, la densità di probabilità a posteriori può dunque essere approssimata tramite le densità di probabilità calcolate in ciascuna cella della griglia.

Il metodo basato su griglia si sviluppa in quattro fasi:

- fissare una griglia discreta di possibili valori θ;[1]
- valutare la distribuzione a priori $p(\theta)$ e la funzione di verosimiglianza $\mathcal{L}(y \mid \theta)$ in corrispondenza di ciascun valore θ della griglia;
- ottenere un'approssimazione discreta della densità a posteriori:

[1] È chiaro che, per ottenere buone approssimazioni, è necessaria una griglia molto densa.

  - per ciascun valore θ della griglia, calcolare il prodotto $p(\theta)\mathcal{L}(y \mid \theta)$;
  - normalizzare i prodotti così ottenuti in modo tale che la loro somma sia 1;
- selezionare N valori casuali della griglia in modo tale da ottenere un campione casuale delle densità a posteriori normalizzate.

Possiamo migliorare l'approssimazione aumentando il numero di punti della griglia. Infatti utilizzando un numero infinito di punti si otterrebbe la descrizione esatta della distribuzione a posteriori, dovendo però pagare il costo dell'utilizzo di infinite risorse di calcolo. Il limite maggiore dell'approccio basato su griglia è che al crescere della dimensionalità N dello spazio dei parametri i punti della griglia necessari per avere una buona stima crescerebbero esponenzialmente con N, rendendo questo metodo inattuabile.

2.2.1 Modello Beta-Binomiale

Per fare un esempio, consideriamo il modello Beta-Binomiale di cui conosciamo la soluzione esatta. Supponiamo di avere osservato 9 sucessi in 10 prove Bernoulliane indipendenti.[2] Imponiamo alla distribuzione a priori su θ (proabilità di successo in una singola prova) una Beta(2, 2) per descrivere la nostra incertezza sul parametro prima di avere osservato i dati. Dunque, il modello diventa:

$$\begin{aligned} Y \mid \theta &\sim \text{Bin}(10, \pi) \\ \theta &\sim \text{Beta}(2, 2). \end{aligned}$$

In queste circostanze, l'aggiornamento bayesiano produce una distribuzione a posteriori Beta di parametri 11 ($y + \alpha = 9 + 2$) e 3 ($n - y + \beta = 10 - 9 + 2$):

$$\theta \mid (y = 9) \sim \text{Beta}(11, 3).$$

Per approssimare la distribuzione a posteriori, fissiamo una griglia di $n = 6$ valori equispaziati: $\theta \in \{0, 0.2, 0.4, 0.6, 0.8, 1\}$ (in seguito aumenteremo n):

[2] La discussione del modello Beta-Binomiale segue molto da vicino la presentazione di Johnson et al. (2022) utilizzando anche lo stesso codice R.

```
grid_data <- tibble(
  theta_grid = seq(from = 0, to = 1, length = 6)
)
```

In corrispondenza di ciascun valore della griglia, valutiamo la distribuzione a priori Beta$(2, 2)$ e la verosimiglianza Bin$(10, \theta)$ con $y = 9$.

```
grid_data <- grid_data %>%
  mutate(
    prior = dbeta(theta_grid, 2, 2),
    likelihood = dbinom(9, 10, theta_grid)
  )
```

In ciascuna cella della griglia calcoliamo poi il prodotto della verosimiglianza e della distribuzione a priori. Troviamo così un'approssimazione discreta e non normalizzata della distribuzione a posteriori (`unnormalized`). Normalizziamo infine questa approssimazione dividendo ciascun valore del vettore `unnormalized` per la somma di tutti i valori del vettore:

```
grid_data <- grid_data %>%
  mutate(
    unnormalized = likelihood * prior,
    posterior = unnormalized / sum(unnormalized)
  )
```

La somma dei valori così trovati sarà uguale a 1:

```
grid_data %>%
  summarize(
    sum(unnormalized),
    sum(posterior)
  )
#> # A tibble: 1 x 2
#>   `sum(unnormalized)` `sum(posterior)`
#>                 <dbl>            <dbl>
#> 1               0.318                1
```

Abbiamo dunque ottenuto la seguente distribuzione a posteriori discretizzata $p(\theta \mid y)$:

```
round(grid_data, 2)
#> # A tibble: 6 x 5
#>   theta_grid prior likelihood unnormalized posterior
#>        <dbl> <dbl>      <dbl>        <dbl>     <dbl>
#> 1        0    0          0            0         0
#> 2        0.2  0.96       0            0         0
#> 3        0.4  1.44       0            0         0.01
#> 4        0.6  1.44       0.04         0.06      0.18
#> 5        0.8  0.96       0.27         0.26      0.81
#> 6        1    0          0            0         0
```

La figura 2.1 mostra un grafico della distribuzione a posteriori discretizzata che è stata ottenuta:

```
grid_data %>%
  ggplot(
    aes(x = theta_grid, y = posterior)
  ) +
  geom_point() +
  geom_segment(
    aes(
      x = theta_grid,
      xend = theta_grid,
      y = 0,
      yend = posterior)
  )
```

L'ultimo passo della simulazione è il campionamento dalla distribuzione a posteriori discretizzata:

```
set.seed(84735)
post_sample <- sample_n(
  grid_data,
  size = 1e5,
  weight = posterior,
```

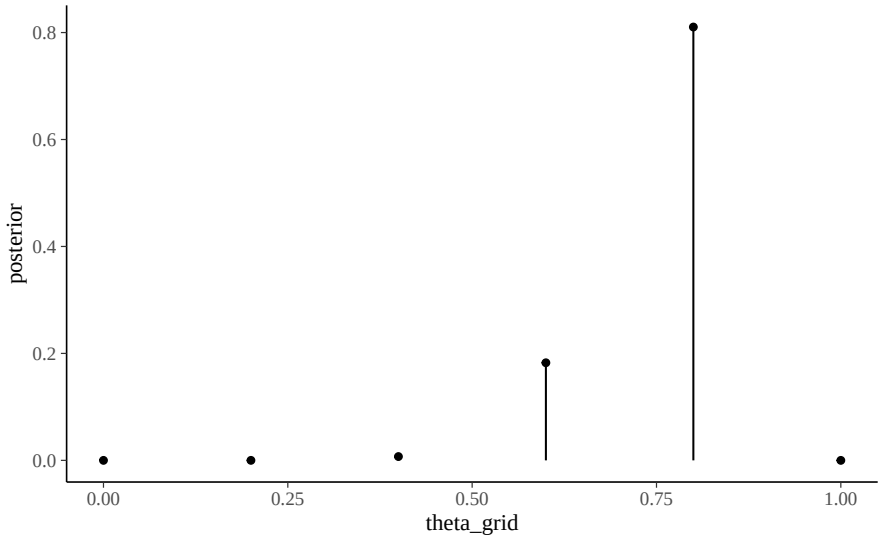


Figura 2.1: *Distribuzione a posteriori discretizzata ottenuta con il metodo grid-based per $y = 9$ successi in 10 prove Bernoulliane, con distribuzione a priori Beta$(2, 2)$. È stata utilizzata una griglia di solo $n = 6$ punti.*

```
  replace = TRUE
)
```

È facile intuire che i valori estratti con rimessa dalla distribuzione a posteriori discretizzata saranno quasi sempre uguali a 0.6 o 0.8. Questa intuizione è confermata dal grafico 2.2 a cui è stata sovrapposta la vera distribuzione a posteriori Beta$(11, 3)$:

```
ggplot(post_sample, aes(x = theta_grid)) +
  geom_histogram(aes(y = ..density..), color = "white") +
  stat_function(fun = dbeta, args = list(11, 3)) +
  lims(x = c(0, 1))
```

La figura 2.2 mostra che, con una griglia così sparsa abbiamo ottenuto una versione estremamente approssimata della vera distribuzione a posteriori. Possiamo però ottenere un risultato migliore con una griglia più fine, come indicato dalla figura 2.3:

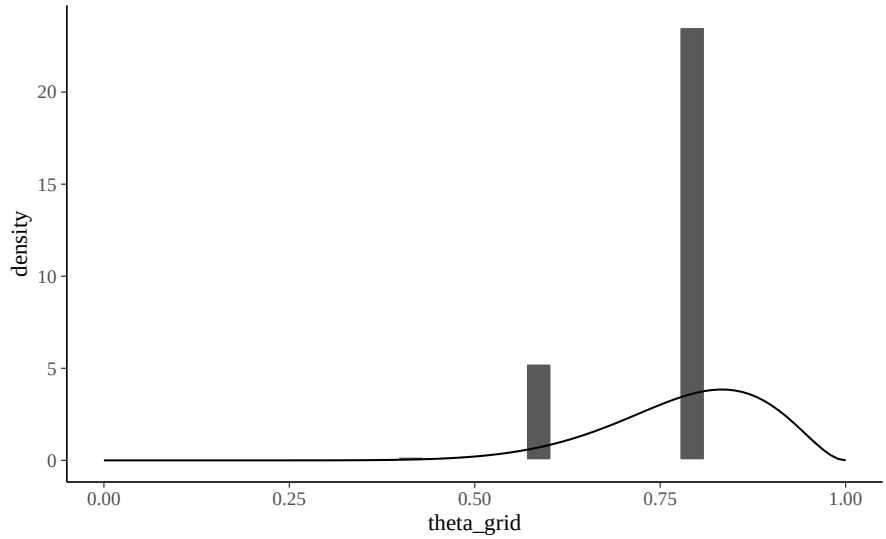


Figura 2.2: *Campionamento dalla distribuzione a posteriori discretizzata ottenuta con il metodo grid-based per $y = 9$ successi in 10 prove Bernoulliane, con distribuzione a priori Beta$(2, 2)$. È stata utilizzata una griglia di solo $n = 6$ punti.*

```
grid_data <- tibble(
  theta_grid = seq(from = 0, to = 1, length.out = 100)
)
grid_data <- grid_data %>%
  mutate(
    prior = dbeta(theta_grid, 2, 2),
    likelihood = dbinom(9, 10, theta_grid)
  )
grid_data <- grid_data %>%
  mutate(
    unnormalized = likelihood * prior,
    posterior = unnormalized / sum(unnormalized)
  )
grid_data %>%
  ggplot(
    aes(x = theta_grid, y = posterior)
  ) +
```

```
  geom_point() +
  geom_segment(
    aes(
      x = theta_grid,
      xend = theta_grid,
      y = 0,
      yend = posterior
    )
  )
```

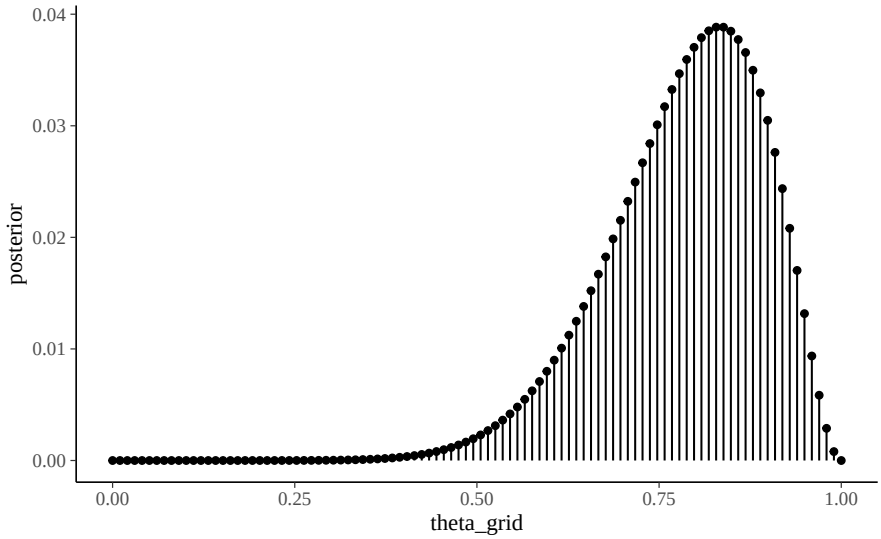


Figura 2.3: *Distribuzione a posteriori discretizzata ottenuta con il metodo grid-based per $y = 9$ successi in 10 prove Bernoulliane, con distribuzione a priori Beta$(2, 2)$. È stata utilizzata una griglia di $n = 100$ punti.*

Campioniamo ora 10000 punti:

```
# Set the seed
set.seed(84735)
post_sample <- sample_n(
  grid_data,
  size = 1e4,
  weight = posterior,
```

```
  replace = TRUE
)
```

Con il campionamento dalla distribuzione a posteriori discretizzata costruita mediante una griglia più densa ($n = 100$) otteniamo un risultato soddisfacente (figura 2.4): ora la distribuzione dei valori prodotti dalla simulazione approssima molto bene la corretta distribuzione a posteriori $p(\theta \mid y) = \text{Beta}(11, 3)$.

```
post_sample %>%
  ggplot(aes(x = theta_grid)) +
  geom_histogram(
    aes(y = ..density..),
    color = "white",
    binwidth = 0.05
  ) +
  stat_function(fun = dbeta, args = list(11, 3)) +
  lims(x = c(0, 1))
```

In conclusione, il metodo basato su griglia è molto intuitivo e non richiede particolari competenze di programmazione per essere implementato. Inoltre, fornisce un risultato che, per tutti gli scopi pratici, può essere considerato come un campione casuale estratto da $p(\theta \mid y)$. Tuttavia, anche se tale metodo fornisce risultati accuratissimi, esso ha un uso limitato. A causa della *maledizione della dimensionalità*[3], tale metodo può solo essere solo nel caso di semplici modelli statistici, con non più di due parametri. Nella pratica concreta tale metodo viene dunque sostituito da altre tecniche più efficienti in quanto, anche nei più comuni modelli utilizzati in psicologia, vengono solitamente stimati centinaia se non migliaia di parametri.

[3]Per capire cosa sia la maledizione della dimensionalità, supponiamo di utilizzare una griglia di 100 punti equispaziati. Nel caso di un solo parametro, è necessario calcolare 100 valori. Per due parametri devono essere calcolari 100^2 valori. Ma già per 10 parametri è necessario calcolare 10^{10} valori – è facile capire che una tale quantità di calcoli è troppo grande anche per un computer molto potente. Per modelli che richiedono la stima di un numero non piccolo di parametri è dunque necessario procedere in un altro modo.

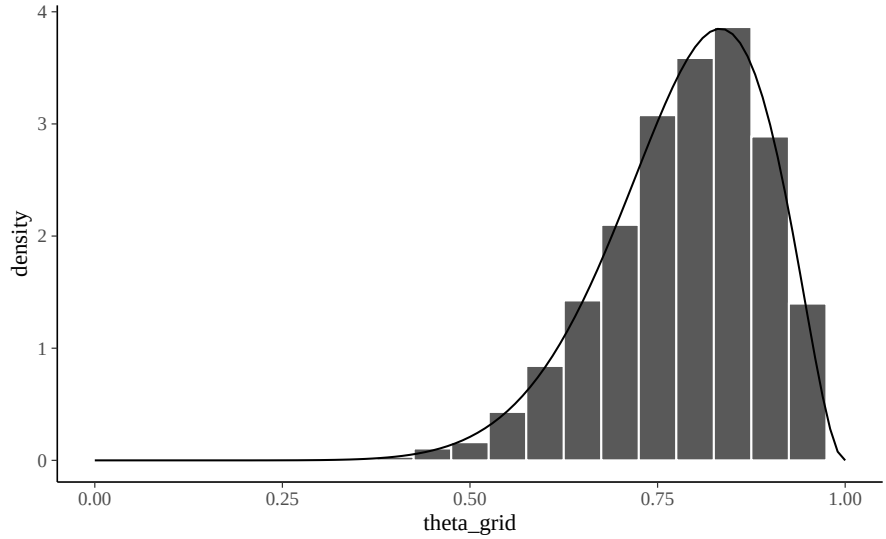


Figura 2.4: *Campionamento dalla distribuzione a posteriori discretizzata ottenuta con il metodo grid-based per $y = 9$ successi in 10 prove Bernoulliane, con distribuzione a priori Beta$(2, 2)$. È stata utilizzata una griglia di $n = 100$ punti. All'istogramma è stata sovrapposta la corretta distribuzione a posteriori, ovvero la densità Beta$(11, 3)$.*

2.3 Approssimazione quadratica

L'approssimazione quadratica è un altro metodo che può essere usato per superare il problema della "maledizione della dimensionalità". La motivazione di tale metodo è la seguente. Sappiamo che, in generale, la regione della distribuzione a posteriori che si trova in prossimità del suo massimo può essere ben approssimata dalla forma di una distribuzione Normale.[4]

L'approssimazione quadratica si pone due obiettivi.

[4]Descrivere la distribuzione a posteriori mediante la distribuzione Normale significa utilizzare un'approssimazione che viene, appunto, chiamata "quadratica" (tale approssimazione si dice quadratica perché il logaritmo di una distribuzione gaussiana forma una parabola e la parabola è una funzione quadratica – dunque, mediante questa approssimazione descriviamo il logaritmo della distribuzione a posteriori mediante una parabola).

1. Trovare la moda della distribuzione a posteriori. Ci sono varie procedure di ottimizzazione, implementate in R, in grado di trovare il massimo di una distribuzione.
2. Stimare la curvatura della distribuzione in prossimità della moda. Una stima della curvatura è sufficiente per trovare un'approssimazione quadratica dell'intera distribuzione. In alcuni casi, questi calcoli possono essere fatti seguendo una procedura analitica, ma solitamente vengono usate delle tecniche numeriche.

Una descrizione della distribuzione a posteriori ottenuta mediante l'approssimazione quadratica si ottiene mediante la funzione `quap()` contenuta nel pacchetto `rethinking`:[5]

```
suppressPackageStartupMessages(library("rethinking"))

mod <- quap(
  alist(
  N ~ dbinom(N + P, p), # verosimiglianza binomiale
  p ~ dbeta(2, 10) # distribuzione a priori Beta(2, 10)
  ),
  data = list(N = 23, P = 7)
)
```

Un sommario dell'approssimazione quadratica è fornito da

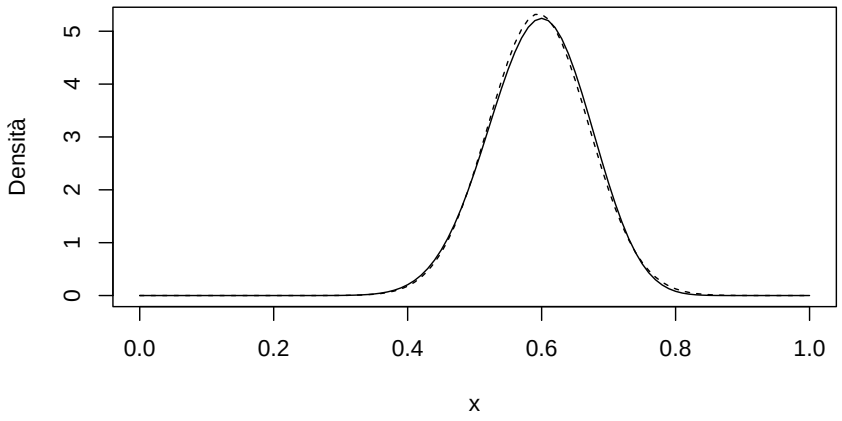
```
precis(mod, prob = 0.95)
#>   mean      sd   2.5%  97.5%
#> p  0.6 0.07746 0.4482 0.7518
```

Qui sotto è fornito un confronto tra la corretta distribuzione a posteriori (linea continua) e l'approssimazione quadratica (linea trateggiata).

```
N <- 23
P <- 7
```

[5] Il pacchetto `rethinking` è stato creato da McElreath (2020) per accompagnare il suo testo *Statistical Rethinking*[2]. Per l'installazione si veda https://github.com/rmcelreath/rethinking.

```
a <- N + 2
b <- P + 10
curve(dbeta(x, a, b), from=0, to=1, ylab="Densità")
# approssimazione quadratica
curve(
  dnorm(x, a/(a+b), sqrt((a*b)/((a+b)^2*(a+b+1)))),
  lty = 2,
  add = TRUE
)
```



Il grafico precedente mostra che l'approssimazione quadratica fornisce risultati soddisfacenti. Tali risultati sono simili (o identici) a quelli ottenuti con il metodo *grid-based*, con il vantaggio aggiuntivo di disporre di una serie di funzioni R in grado di svolgere i calcoli per noi. In realtà, però, l'approssimazione quadratica è poco usata perché, per problemi complessi, è più conveniente fare ricorso ai metodi Monte Carlo basati su Catena di Markov (MCMC) che verranno descritti nel Paragrafo successivo.

2.4 Metodo Monte Carlo

I metodi più ampiamente adottati nell'analisi bayesiana per la costruzione della distribuzione a posteriori per modelli complessi sono i metodi di campionamento detti metodi Monte Carlo basati su catena di Markov (*Markov Chain Monte Carlo*, MCMC). Tali metodi consentono di decidere quali distribuzioni a priori e quali distribuzioni di verosimiglianza usare sulla base di considerazioni teoriche soltanto, senza dovere preoccuparsi di altri vincoli. Dato che è basata su metodi computazionalmente intensivi, la stima numerica MCMC della funzione a posteriori può essere svolta soltanto mediante software. In anni recenti i metodi Bayesiani di analisi dei dati sono diventati sempre più popolari proprio perché la potenza di calcolo necessaria per svolgere tali calcoli è ora alla portata di tutti. Questo non era vero solo pochi decenni fa.

Per introdurre i metodi MCMC consideriamo il caso di una verosimiglianza Binomiale e di una distribuzione a priori Beta. Sappiamo che, in tali circostanze, viene prodotta una distribuzione a posteriori Beta (si veda il capitolo **??**). Con una simulazione R è dunque facile ricavare dei campioni causali dalla distribuzione a posteriori. Maggiore è il numero di campioni, migliore sarà l'approssimazione della distribuzione a posteriori.

Consideriamo nuovamente i dati di Zetsche et al. (2019) (23 "successi" in 30 prove Bernoulliane) e applichiamo a quei dati lo stesso modello del Capitolo **??**:

$$
\begin{aligned}
y \mid \theta, n &\sim \text{Bin}(y = 23, n = 30 \mid \theta) \\
\theta_{prior} &\sim \text{Beta}(2, 10) \\
\theta_{post} &\sim \text{Beta}(y + a = 23 + 2 = 25, n - y + b = 30 - 23 + 10 = 17),
\end{aligned}
$$

Poniamoci il problema di stimare il valore della media a posteriori di θ. Nel caso presente, il risultato esatto è

$$\bar{\theta}_{post} = \frac{\alpha}{\alpha + \beta} = \frac{25}{25 + 17} \approx 0.5952.$$

Dato che la distribuzione a posteriori di θ è Beta$(25, 17)$, possiamo estrarre un campione casuale di osservazioni da tale distribuzione e calcolare la media:

```
set.seed(7543897)
print(mean(rbeta(1e2, shape1 = 25, shape2 = 17)), 6)
#> [1] 0.587548
```

È ovvio che l'approssimazione migliora all'aumentare del numero di osservazioni estratte dalla distribuzione a posteriori (legge dei grandi numeri):

```
print(mean(rbeta(1e3, shape1 = 25, shape2 = 17)), 6)
#> [1] 0.597659
```

```
print(mean(rbeta(1e4, shape1 = 25, shape2 = 17)), 6)
#> [1] 0.595723
```

```
print(mean(rbeta(1e5, shape1 = 25, shape2 = 17)), 6)
#> [1] 0.595271
```

Quando il numero di osservazioni (possiamo anche chiamarle "campioni") tratte dalla distribuzione a posteriori è molto grande, la distribuzione di tali campioni converge alla densità della popolazione (si veda l'Appendice **??**).[6]

Inoltre, le statistiche descrittive (es. media, moda, varianza, eccetera) dei campioni estratti dalla distribuzione a posteriori convergeranno ai corrispondenti valori della distribuzione a posteriori. La figura 2.5 mostra come, all'aumentare del numero di repliche, la media, la mediana, la deviazione standard e l'asimmetria convergono ai veri valori della distribuzione a posteriori (linee rosse tratteggiate).

[6] Si noti, naturalmente, che il numero dei campioni di simulazione è controllato dal ricercatore; è totalmente diverso dalla dimensione del campione che è fissa ed è una proprietà dei dati.

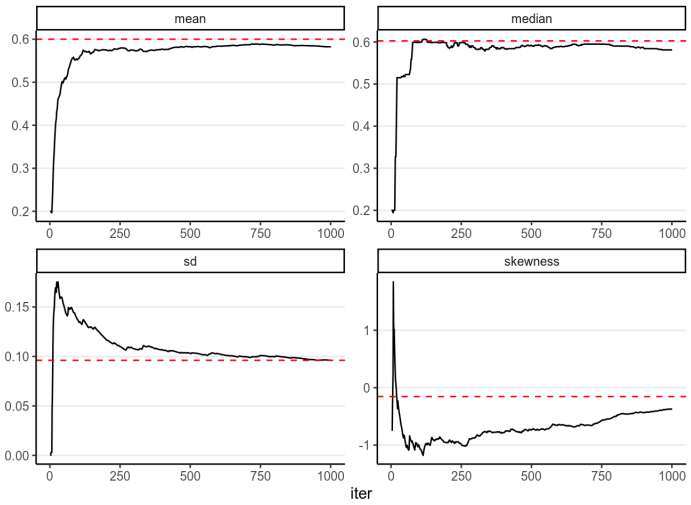


Figura 2.5: *Convergenza delle simulazioni Monte Carlo.*

2.5 Metodi MC basati su Catena di Markov

Nel Paragrafo 2.4 la simulazione Monte Carlo funzionava perché

- sapevamo che la distribuzione a posteriori era una Beta$(25, 17)$,
- era possibile usare le funzioni R per estrarre campioni casuali da tale distribuzione.

Tuttavia, capita raramente di usare una distribuzione a priori coniugata alla verosimiglianza, quindi in generale le due condizioni descritte sopra non si applicano. Ad esempio, nel caso di una verosimiglianza binomiale e una distribuzione a priori Normale, la distribuzione a posteriori di θ è

$$p(\theta \mid y) = \frac{\mathrm{e}^{-(\theta-1/2)^2}\theta^y(1-\theta)^{n-y}}{\int_0^1 \mathrm{e}^{-(t-1/2)^2}t^y(1-t)^{n-y}dt}.$$

Una tale distribuzione non è implementata in R e dunque non possiamo campionare da $p(\theta \mid y)$. Per fortuna, gli algoritmi MCMC consentono il campionamento da una distribuzione a posteriori *senza che sia necessario conoscere la rappresentazione analitica di una tale distribuzione.* I metodi Monte Carlo basati su catena di Markov consentono di costruire sequenze di punti (detti catene di Markov) nello spazio dei parametri le

cui densità sono proporzionali alla distribuzione a posteriori — in altre parole, dopo aver simulato un grande numero di passi della catena si possono usare i valori così generati come se fossero un campione casuale della distribuzione a posteriori. Le tecniche MCMC sono attualmente il metodo computazionale maggiormente utilizzato per risolvere i problemi di inferenza bayesiana. Un'introduzione alle catene di Markov è fornita nell'Appendice **??**.

2.5.1 Campionamento mediante algoritmi MCMC

Un modo generale per ottenere una catena di Markov la cui distribuzione equivale alla distribuzione a posteriori $p(\theta \mid y)$ è quello di usare l'algoritmo di Metropolis. L'algoritmo di Metropolis è il primo algoritmo MCMC che è stato proposto, ed è applicabile ad una grande varietà di problemi inferenziali di tipo bayesiano. Tale algoritmo è stato in seguito sviluppato allo scopo di renderlo via via più efficiente. Lo presentiamo qui in una forma intuitiva.

2.5.2 Una passeggiata casuale sui numeri naturali

Per introdurre l'algoritmo di di Metropolis considereremo il campionamento da una distribuzione discreta.[7] Supponiamo di definire una distribuzione di probabilità discreta sugli interi $1, \dots, K$. Scriviamo in R la funzione `pd()` che assegna ai valori $1, \dots, 8$ delle probabilità proporzionali a 5, 10, 4, 4, 20, 20, 12 e 5.

```
pd <- function(x){
  values <- c(5, 10, 4, 4, 20, 20, 12, 5)
  ifelse(
    x %in% 1:length(values),
    values[x] / sum(values),
    0
  )
}
prob_dist <- tibble(
  x = 1:8,
```

[7] Seguiamo qui la trattazione di Albert and Hu (2019). Per una presentazione intuitiva dell'algoritmo di Metropolis, si vedano anche Kruschke (2014); McElreath (2020).

```
  prob = pd(1:8)
)
```

La figura 2.6 illustra la distribuzione di probabilità che è stata generata.

```
x <- 1:8
prob_dist %>%
  ggplot(aes(x = x, y = prob)) +
  geom_bar(stat = "identity", width = 0.06) +
  scale_x_continuous("x", labels = as.character(x), breaks = x) +
  labs(
    y = "Probabilità",
    x = "X"
  )
```

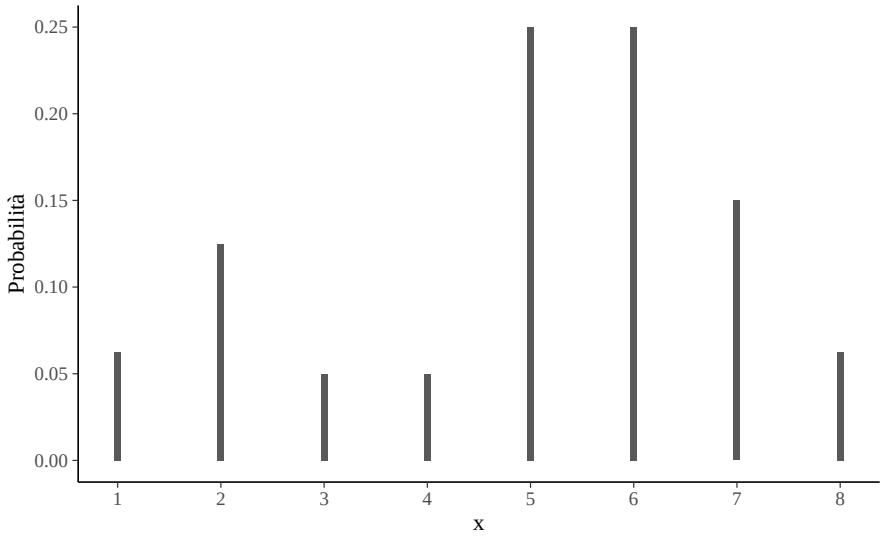


Figura 2.6: *Distribuzione di massa di probabilità per una variabile casuale avente valori 1, 2, ..., 8.*

L'algoritmo di Metropolis corrisponde alla seguente passeggiata casuale.

1. L'algoritmo inizia con un valore iniziale qualsiasi da 1 a $K = 8$ della variabile casuale.

2. Per simulare il valore successivo della sequenza, lanciamo una moneta equilibrata. Se esce testa, consideriamo come valore candidato il valore immediatamente precedente al valore corrente nella sequenza $1, \ldots, 8$; se esce croce, il valore candidato sarà il valore immediatamente successivo al valore corrente nella sequenza.
3. Calcoliamo il rapporto tra la probabilità del valore candidato e la probabilità del valore corrente:

$$R = \frac{pd(\text{valore candidato})}{pd(\text{valore corrente})}.$$

4. Estraiamo un numero a caso $\in [0, 1]$. Se tale valore è minore di R accettiamo il valore candidato come valore successivo della catena markoviana; altrimenti il valore successivo della catena rimane il valore corrente.

I passi da 1 a 4 definiscono una catena di Markov irriducibile e aperiodica sui valori di stato $\{1, 2, \ldots, 8\}$, dove il passo 1 fornisce il valore iniziale della catena e i passi da 2 a 4 definiscono la matrice di transizione P. Un modo di campionare da una distribuzione di massa di probabilità `pd` consiste nell'iniziare da una posizione qualsiasi e eseguire una passeggiata casuale costituita da un grande numero di passi, ripetendo le fasi 2, 3 e 4 dell'algoritmo di Metropolis. Dopo un grande numero di passi, la distribuzione dei valori della catena markoviana approssimerà la distribuzione di probabilità `pd`.

La funzione `random_walk()` implementa l'algoritmo di Metropolis. Tale funzione richiede in input la distribuzione di probabilità `pd`, la posizione di partenza `start` e il numero di passi dell'algoritmo `num_steps`.

```
random_walk <- function(pd, start, num_steps){
  y <- rep(0, num_steps)
  current <- start
  for (j in 1:num_steps){
    candidate <- current + sample(c(-1, 1), 1)
    prob <- pd(candidate) / pd(current)
    if (runif(1) < prob)
      current <- candidate
```

```
    y[j] <- current
  }
  return(y)
}
```

Di seguito, implementiamo l'algoritmo di Metropolis utilizzando, quale valore iniziale, $X = 4$. Ripetiamo la simulazione 10,000 volte.

```
out <- random_walk(pd, 4, 1e4)

S <- tibble(out) %>%
  group_by(out) %>%
  summarize(
    N = n(),
    Prob = N / 10000
  )

prob_dist2 <- rbind(
  prob_dist,
  tibble(
    x = S$out,
    prob = S$Prob
  )
)
prob_dist2$Type <- rep(
  c("Prob. corrette", "Prob. simulate"),
  each = 8
)
```

```
x <- 1:8
prob_dist2 %>%
  ggplot(aes(x = x, y = prob, fill = Type)) +
  geom_bar(
    stat = "identity",
    width = 0.1,
    position = position_dodge(0.3)
  ) +
```

```
  scale_x_continuous(
    "x",
    labels = as.character(x),
    breaks = x
  ) +
  scale_fill_manual(values = c("black", "gray80")) +
  theme(legend.title = element_blank()) +
  labs(
    y = "Probabilità",
    x = "X"
  )
```

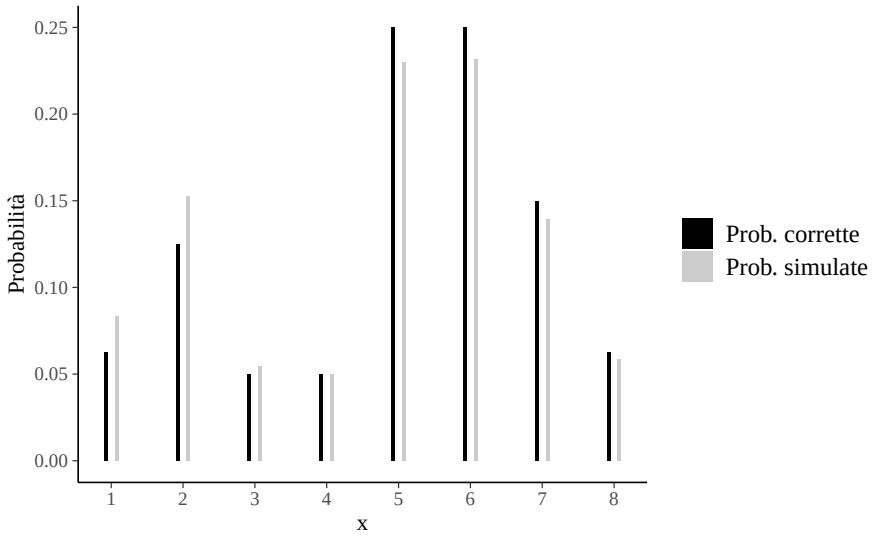


Figura 2.7: *L'istogramma confronta i valori prodotti dall'algoritmo di Metropolis con i corretti valori della distribuzione di massa di probabilità.*

La figura 2.7 confronta l'istogramma dei valori simulati dalla passeggiata casuale con l'effettiva distribuzione di probabilità `pd`. Si noti la somiglianza tra le due distribuzioni.

2.5.3 L'algoritmo di Metropolis

Vediamo ora come l'algoritmo di Metropolis possa venire usato per generare una catena di Markov irriducibile e aperiodica per la quale la distri-

buzione stazionaria è uguale alla distribuzione a posteriori di interesse.[8] In termini generali, l'algoritmo di Metropolis include due fasi.

- *Fase 1.* La selezione di un valore candidato θ' del parametro mediante il campionamento da una distribuzione proposta.
- *Fase 2.* La decisione tra la possibilità di accettare il valore candidato $\theta^{(m+1)} = \theta'$ o di mantenere il valore corrente $\theta^{(m+1)} = \theta$ sulla base del seguente criterio:
  - se $\mathcal{L}(\theta' \mid y)p(\theta') > \mathcal{L}(\theta \mid y)p(\theta)$ il valore candidato viene sempre accettato;
  - altrimenti il valore candidato viene accettato solo in una certa proporzione di casi.

Esaminiamo ora nei dettagli il funzionamento dell'algoritmo di Metropolis.

(a) Si inizia con un punto arbitrario $\theta^{(1)}$, quindi il primo valore della catena di Markov $\theta^{(1)}$ può corrispondere semplicemente ad un valore a caso tra i valori possibili del parametro.

(b) Per ogni passo successivo della catena, $m + 1$, si campiona un valore candidato θ' da una distribuzione proposta: $\theta' \sim \Pi(\theta)$. La distribuzione proposta può essere qualunque distribuzione, anche se, idealmente, è meglio che sia simile alla distribuzione a posteriori. In pratica, però, la distribuzione a posteriori è sconosciuta e quindi il valore θ' viene campionato da una qualche distribuzione simmetrica centrata sul valore corrente $\theta^{(m)}$ del parametro. Nell'esempio qui discusso, useremo la distribuzione gaussiana. Tale distribuzione sarà centrata sul valore corrente della catena e avrà una appropriata deviazione standard: $\theta' \sim \mathcal{N}(\theta^{(m)}, \sigma)$. In pratica, questo significa che, se σ è piccola, il valore candidato θ' sarà simile al valore corrente $\theta^{(m)}$.

(c) Una volta generato il valore candidato θ' si calcola il rapporto tra la densità della distribuzione a posteriori non normalizzata nel punto θ' [ovvero, il prodotto tra la verosimiglianza $\mathcal{L}(y \mid \theta')$ nel punto θ' e la distribuzione a priori nel punto θ'] e la densità della distribuzione a posteriori non normalizzata nel punto $\theta^{(m)}$

[8] Una illustrazione visiva di come si svolge il processo di "esplorazione" dell'algoritmo di Metropolis è fornita in questo post[9].

[ovvero, il prodotto tra la verosimiglianza $\mathcal{L}(y \mid \theta^{(m)})$ nel punto $\theta^{(m)}$ e la distribuzione a priori nel punto $\theta^{(m)}$]:

$$\alpha = \frac{p(y \mid \theta')p(\theta')}{p(y \mid \theta^{(m)})p(\theta^{(m)})}. \tag{2.2}$$

Si noti che, essendo un rapporto, la (2.2) cancella la costante di normalizzazione.

(d) Il rapporto α viene utilizzato per decidere se accettare il valore candidato θ', oppure se campionare un diverso candidato. Possiamo pensare al rapporto α come alla risposta alla seguente domanda: alla luce dei dati, è più plausibile il valore candidato del parametro o il valore corrente? Se α è maggiore di 1 ciò significa che il valore candidato è più plausibile del valore corrente; in tali circostanze il valore candidato viene sempre accettato. Altrimenti, si decide di accettare il valore candidato con una probabilità minore di 1, ovvero non sempre, ma soltanto con una probabilità uguale ad α. Se α è uguale a 0.10, ad esempio, questo significa che la plausibilità a posteriori del valore candidato è 10 volte più piccola della plausibilità a posteriori del valore corrente. Dunque, il valore candidato verrà accettato solo nel 10% dei casi. Come conseguenza di questa strategia di scelta, l'algoritmo di Metropolis ottiene un campione casuale dalla distribuzione a posteriori, dato che la probabilità di accettare il valore candidato è proporzionale alla densità del candidato nella distribuzione a posteriori. Dal punto di vista algoritmico, la procedura descritta sopra viene implementata confrontando il rapporto α con un valore casuale estratto da una distribuzione uniforme Unif$(0, 1)$. Se $\alpha > u \sim$ Unif$(0, 1)$ allora il punto candidato θ' viene accettato e la catena si muove in quella nuova posizione, ovvero $\theta^{(m+1)} = \theta'^{(m+1)}$. Altrimenti $\theta^{(m+1)} = \theta^{(m)}$ e si campiona un nuovo valore candidato θ'.

(e) Il passaggio finale dell'algoritmo calcola l'*accettanza* in una specifica esecuzione dell'algoritmo, ovvero la proporzione dei valori candidati θ' che sono stati accettati come valori successivi nella sequenza.

L'algoritmo di Metropolis prende come input il numero M di passi da simulare, la deviazione standard σ della distribuzione proposta e la densità a priori, e ritorna come output la sequenza $\theta^{(1)}, \theta^{(2)}, \dots, \theta^{(M)}$. La chiave del successo dell'algoritmo di Metropolis è il numero di passi fino a che la catena approssima la stazionarietà. Tipicamente i primi da 1000 a 5000 elementi sono scartati. Dopo un certo periodo k (detto di *burn-in*), la catena di Markov converge ad una variabile casuale che è distribuita secondo la distribuzione a posteriori. In altre parole, i campioni del vettore $\left(\theta^{(k+1)}, \theta^{(k+2)}, \dots, \theta^{(M)}\right)$ diventano campioni di $p(\theta \mid y)$.

2.5.4 Una applicazione concreta

Per fare un esempio concreto, consideriamo nuovamente i 30 pazienti esaminati da Zetsche et al. (2019). Di essi, 23 hanno manifestato aspettative distorte negativamente sul loro stato d'animo futuro. Utilizzando l'algoritmo di Metropolis, ci poniamo il problema di ottenere la stima a posteriori di θ (probabilità di manifestare un'aspettativa distorta negativamente), dati 23 "successi" in 30 prove, imponendo su θ la stessa distribuzione a priori usata nel Capitolo **??**, ovvero Beta$(2, 10)$.

Per calcolare la funzione di verosimiglianza, avendo fissato i dati di Zetsche et al. (2019), definiamo la funzione `likelihood()`

```
likelihood <- function(param, x = 23, N = 30) {
  dbinom(x, N, param)
}
```

che ritorna l'ordinata della verosimiglianza binomiale per ciascun valore del vettore `param` in input.

La distribuzione a priori Beta$(2, 10)$ è implementata nella funzione `prior()`:

```
prior <- function(param, alpha = 2, beta = 10) {
  dbeta(param, alpha, beta)
}
```

Il prodotto della densità a priori e della verosimiglianza è implementato nella funzione `posterior()`:

```
posterior <- function(param) {
  likelihood(param) * prior(param)
}
```

L'Appendice **??** mostra come un'approssimazione della distribuzione a posteriori $p(\theta \mid y)$ per questi dati possa essere ottenuta mediante il metodo basato su griglia.

2.5.5 Implementazione

Per implementare l'algoritmo di Metropolis utilizzeremo una distribuzione proposta gaussiana. Il valore candidato sarà dunque un valore selezionato a caso da una gaussiana di parametri μ uguale al valore corrente nella catena e $\sigma = 0.9$. In questo esempio, la deviazione standard σ è stata scelta empiricamente in modo tale da ottenere una accettanza adeguata. L'accettanza ottimale è di circa 0.20 e 0.30 — se l'accettanza è troppo grande, l'algoritmo esplora uno spazio troppo ristretto della distribuzione a posteriori.[10]

```
proposal_distribution <- function(param) {
  while(1) {
    res = rnorm(1, mean = param, sd = 0.9)
    if (res > 0 & res < 1)
      break
  }
  res
}
```

Nella presente implementazione del campionamento dalla distribuzione proposta è stato inserito un controllo che impone al valore candidato di essere incluso nell'intervallo [0, 1].[11]

L'algoritmo di Metropolis viene implementato nella seguente funzione:

[10] L'accettanza dipende dalla distribuzione proposta: in generale, tanto più la distribuzione proposta è simile alla distribuzione target, tanto più alta diventa l'accettanza.

[11] Si possono trovare implementazioni dell'algoritmo di Metropolis più eleganti di quella presentata qui. Lo scopo dell'esercizio è quello di illustrare la logica soggiacente all'algoritmo di Metropolis, non quello di proporre un'implementazione efficente dell'algoritmo.

```
run_metropolis_MCMC <- function(startvalue, iterations) {
  chain <- vector(length = iterations + 1)
  chain[1] <- startvalue
  for (i in 1:iterations) {
    proposal <- proposal_distribution(chain[i])
    r <- posterior(proposal) / posterior(chain[i])
    if (runif(1) < r) {
      chain[i + 1] <- proposal
    } else {
      chain[i + 1] <- chain[i]
    }
  }
  chain
}
```

Avendo definito le funzioni precedenti, generiamo una catena di valori θ:

```
set.seed(123)
startvalue <- runif(1, 0, 1)
niter <- 1e4
chain <- run_metropolis_MCMC(startvalue, niter)
```

Mediante le istruzioni precedenti otteniamo una catena di Markov costituita da 10,001 valori. Escludiamo i primi 5,000 valori considerati come burn-in. Ci restano dunque con 5,001 valori che verranno considerati come un campione casuale estratto dalla distribuzione a posteriori $p(\theta \mid y)$.

L'accettanza è pari a

```
burnIn <- niter / 2
acceptance <- 1 - mean(duplicated(chain[-(1:burnIn)]))
acceptance
#> [1] 0.2511
```

il che conferma la bontà della deviazione standard ($\sigma = 0.9$) scelta per la distribuzione proposta.

A questo punto è facile ottenere una stima a posteriori del parametro θ. Per esempio, la stima della media a posteriori è:

```
mean(chain[-(1:burnIn)])
#> [1] 0.5922
```

Una figura che mostra l'approssimazione di $p(\theta \mid y)$ ottenuta con l'algoritmo di Metropolis, insieme ad un *trace plot* dei valori della catena di Markov, viene prodotta usando le seguenti istruzioni:

```
p1 <- tibble(
  x = chain[-(1:burnIn)]
) %>%
  ggplot(aes(x)) +
  geom_histogram() +
  labs(
    x = expression(theta),
    y = "Frequenza",
    title = "Distribuzione a posteriori"
  ) +
  geom_vline(
    xintercept = mean(chain[-(1:burnIn)])
  )
p2 <- tibble(
  x = 1:length(chain[-(1:burnIn)]),
  y = chain[-(1:burnIn)]
) %>%
  ggplot(aes(x, y)) +
  geom_line() +
  labs(
    x = "Numero di passi",
    y = expression(theta),
    title = "Valori della catena"
  ) +
  geom_hline(
    yintercept = mean(chain[-(1:burnIn)]),
    colour = "gray"
  )
p1 + p2
```

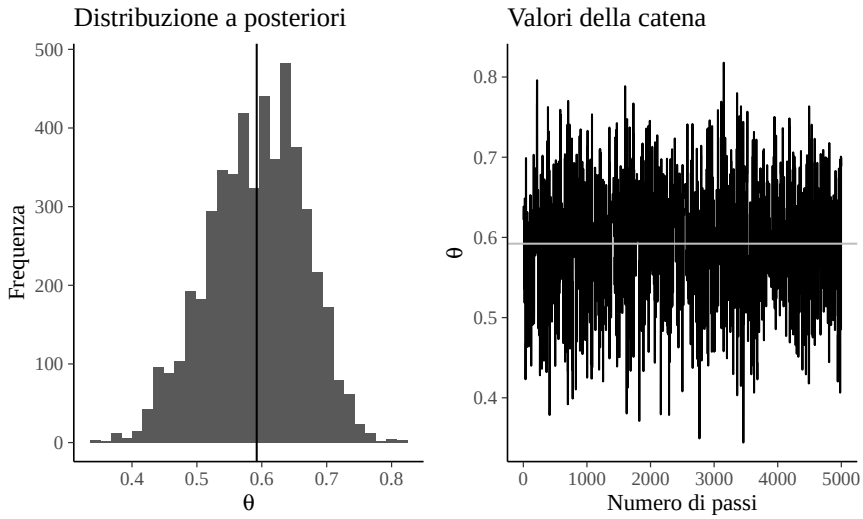


Figura 2.8: *Sinistra. Stima della distribuzione a posteriori della probabilità di una aspettativa futura distorta negativamente per i dati di Zetsche et al. (2019). Destra. Trace plot dei valori della catena di Markov escludendo il periodo di burn-in.*

2.5.6 Input

Negli esempi discussi in questo Capitolo abbiamo illustrato l'esecuzione di una singola catena in cui si parte un unico valore iniziale e si raccolgono i valori simulati da molte iterazioni. È possibile che i valori di una catena siano influenzati dalla scelta del valore iniziale. Quindi una raccomandazione generale è di eseguire l'algoritmo di Metropolis più volte utilizzando diversi valori di partenza. In questo caso, si avranno più catene di Markov. Confrontando le proprietà delle diverse catene si esplora la sensibilità dell'inferenza alla scelta del valore di partenza. I software MCMC consentono sempre all'utente di specificare diversi valori di partenza e di generare molteplici catene di Markov.

2.6 Stazionarietà

Un punto importante da verificare è se il campionatore ha raggiunto la sua distribuzione stazionaria. La convergenza di una catena di Markov

alla distribuzione stazionaria viene detta "mixing".

2.6.1 Autocorrelazione

Informazioni sul "mixing" della catena di Markov sono fornite dall'autocorrelazione. L'autocorrelazione misura la correlazione tra i valori successivi di una catena di Markov. Il valore m-esimo della serie ordinata viene confrontato con un altro valore ritardato di una quantità k (dove k è l'entità del ritardo) per verificare quanto si correli al variare di k. L'autocorrelazione di ordine 1 (*lag 1*) misura la correlazione tra valori successivi della catena di Markow (cioè, la correlazione tra $\theta^{(m)}$ e $\theta^{(m-1)}$); l'autocorrelazione di ordine 2 (*lag 2*) misura la correlazione tra valori della catena di Markow separati da due "passi" (cioè, la correlazione tra $\theta^{(m)}$ e $\theta^{(m-2)}$); e così via.

L'autocorrelazione di ordine k è data da ρ_k e può essere stimata come:

$$\begin{aligned}\rho_k &= \frac{\text{Cov}(\theta_m, \theta_{m+k})}{\mathbb{V}(\theta_m)} \\ &= \frac{\sum_{m=1}^{n-k}(\theta_m - \bar{\theta})(\theta_{m-k} - \bar{\theta})}{\sum_{m=1}^{n-k}(\theta_m - \bar{\theta})^2} \qquad \text{con} \quad \bar{\theta} = \frac{1}{n}\sum_{m=1}^{n}\theta_m. \end{aligned} \tag{2.3}$$

Per fare un esempio pratico, simuliamo dei dati autocorrelati con la funzione R `colorednoise::colored_noise()`:

```
suppressPackageStartupMessages(library("colorednoise"))
set.seed(34783859)
rednoise <- colored_noise(
  timesteps = 30, mean = 0.5, sd = 0.05, phi = 0.3
)
```

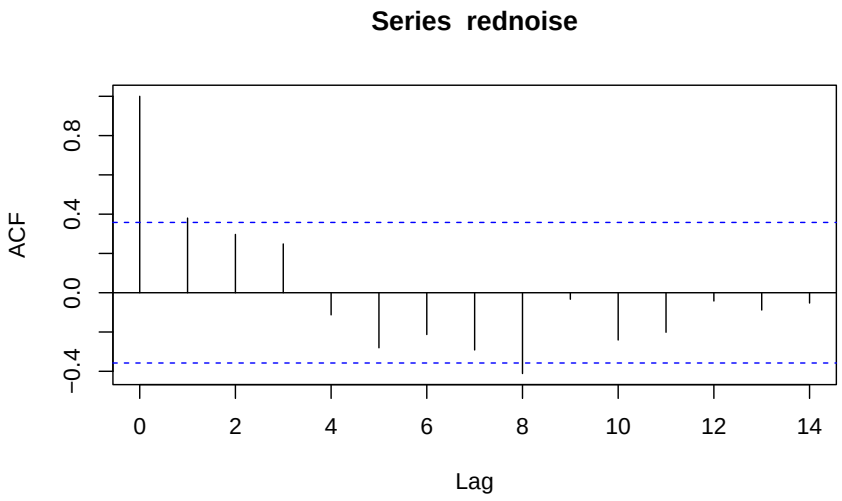
L'autocorrelazione di ordine 1 è semplicemente la correlazione tra ciascun elemento e quello successivo nella sequenza. Nell'esempio, il vettore `rednoise` è una sequenza temporale di 30 elementi. Il vettore `rednoise[-length(rednoise)]` include gli elementi con gli indici da 1 a 29 nella sequenza originaria, mentre il vettore `rednoise[-1]` include gli elementi 2:30. Gli elementi delle coppie ordinate dei due vettori avranno dunque gli indici $(1, 2), (2, 3), ... (29, 30)$ degli elementi della sequenza originaria. La correlazione di Pearson tra i vettori `rednoise[-length(rednoise)]` e

`rednoise[-1]` corrisponde dunque all'autocorrelazione di ordine 1 della serie temporale.

```
cor(rednoise[-length(rednoise)], rednoise[-1])
#> [1] 0.3967
```

Il Correlogramma è uno strumento grafico usato per la valutazione della tendenza di una catena di Markov nel tempo. Il correlogramma si costruisce a partire dall'autocorrelazione ρ_k di una catena di Markov in funzione del ritardo (*lag*) k con cui l'autocorrelazione è calcolata: nel grafico ogni barretta verticale riporta il valore dell'autocorrelazione (sull'asse delle ordinate) in funzione del ritardo (sull'asse delle ascisse). In R, il correlogramma può essere prodotto con una chiamata a `acf()`:

```
acf(rednoise)
```



Il correlogramma precedente mostra come l'autocorrelazione di ordine 1 sia circa pari a 0.4 e diminuisce per lag maggiori; per lag di 4, l'autocorrelazione diventa negativa e aumenta progressivamente fino ad un lag di 8; eccetera.

In situazioni ottimali l'autocorrelazione diminuisce rapidamente ed è effettivamente pari a 0 per piccoli lag. Ciò indica che i valori della catena di Markov che si trovano a più di soli pochi passi di distanza gli uni

dagli altri non risultano associati tra loro, il che fornisce conferma del "mixing" della catena di Markov, ossia di convergenza alla distribuzione stazionaria. Nelle analisi bayesiane, una delle strategie che consentono di ridurre l'autocorrelazione è quella di assottigliare l'output immagazzinando solo ogni m-esimo punto dopo il periodo di burn-in. Una tale strategia va sotto il nome di *thinning*.

2.6.2 Test di convergenza

Un test di convergenza può essere svolto in maniera grafica mediante le tracce delle serie temporali (*trace plot*), cioè il grafico dei valori simulati rispetto al numero di iterazioni. Se la catena è in uno stato stazionario le tracce mostrano assenza di periodicità nel tempo e ampiezza costante, senza tendenze visibili o andamenti degni di nota. Un esempio di *trace plot* è fornito nella figura 2.8 (destra).

Ci sono inoltre alcuni test che permettono di verificare la stazionarietà del campionatore dopo un dato punto. Uno è il test di Geweke che suddivide il campione, dopo aver rimosso un periodo di burn in, in due parti. Se la catena è in uno stato stazionario, le medie dei due campioni dovrebbero essere uguali. Un test modificato, chiamato Geweke z-score, utilizza un test z per confrontare i due subcampioni ed il risultante test statistico, se ad esempio è più alto di 2, indica che la media della serie sta ancora muovendosi da un punto ad un altro e quindi è necessario un periodo di burn-in più lungo.

Considerazioni conclusive

In generale, la distribuzione a posteriori dei parametri di un modello statistico non può essere determinata per via analitica. Tale problema, invece, viene affrontato facendo ricorso ad una classe di algoritmi per il campionamento da distribuzioni di probabilità che sono estremamente onerosi dal punto di vista computazionale e che possono essere utilizzati nelle applicazioni pratiche solo grazie alla potenza di calcolo dei moderni computer. Lo sviluppo di software che rendono sempre più semplice l'uso dei metodi MCMC, insieme all'incremento della potenza di calcolo dei computer, ha contribuito a rendere sempre più popolare il metodo dell'inferenza bayesiana che, in questo modo, può essere estesa a problemi

di qualunque grado di complessità.

Nel 1989 un gruppo di statistici nel Regno Unito si pose il problema di simulare le catene di Markov su un personal computer. Nel 1997 ci riuscirono con il primo rilascio pubblico di un'implementazione Windows dell'inferenza bayesiana basata su Gibbs sampling, detta BUGS. Il materiale presentato in questo capitolo descrive gli sviluppi contemporanei del percorso che è iniziato in quel periodo.

A

Simbologia di base

Per una scrittura più sintetica possono essere utilizzati alcuni simboli matematici.

- $\log(x)$: il logaritmo naturale di x.
- L'operatore logico booleano $\wedge$ significa "e" (congiunzione forte) mentre il connettivo di disgiunzione $\vee$ significa "o" (oppure) (congiunzione debole).
- Il quantificatore esistenziale $\exists$ vuol dire "esiste almeno un" e indica l'esistenza di almeno una istanza del concetto/oggetto indicato. Il quantificatore esistenziale di unicità $\exists!$ ("esiste soltanto un") indica l'esistenza di esattamente una istanza del concetto/oggetto indicato. Il quantificatore esistenziale $\nexists$ nega l'esistenza del concetto/oggetto indicato.
- Il quantificatore universale $\forall$ vuol dire "per ogni."
- $\mathcal{A}, \mathcal{S}$: insiemi.
- $x \in A$: x è un elemento dell'insieme A.
- L'implicazione logica "$\Rightarrow$" significa "implica" (se …allora). $P \Rightarrow Q$ vuol dire che P è condizione sufficiente per la verità di Q e che Q è condizione necessaria per la verità di P.
- L'equivalenza matematica " $\Longleftrightarrow$ " significa "se e solo se" e indica una condizione necessaria e sufficiente, o corrispondenza biunivoca.
- Il simbolo | si legge "tale che."
- Il simbolo $\triangleq$ (o $:=$) si legge "uguale per definizione."
- Il simbolo Δ indica la differenza fra due valori della variabile scritta a destra del simbolo.
- Il simbolo $\propto$ si legge "proporzionale a."
- Il simbolo $\approx$ si legge "circa."
- Il simbolo $\in$ della teoria degli insiemi vuol dire "appartiene" e indica l'appartenenza di un elemento ad un insieme. Il simbolo $\notin$ vuol dire "non appartiene."
- Il simbolo $\subseteq$ si legge "è un sottoinsieme di" (può coincidere con l'insieme stesso). Il simbolo $\subset$ si legge "è un sottoinsieme proprio di."

- Il simbolo # indica la cardinalità di un insieme.
- Il simbolo $\cap$ indica l'intersezione di due insiemi. Il simbolo $\cup$ indica l'unione di due insiemi.
- Il simbolo $\emptyset$ indica l'insieme vuoto o evento impossibile.
- In matematica, argmax identifica l'insieme dei punti per i quali una data funzione raggiunge il suo massimo. In altre parole, $\text{argmax}_x f(x)$ è l'insieme dei valori di x per i quali $f(x)$ raggiunge il valore più alto.
- a, c, α, γ: scalari.
- x, y: vettori.
- X, Y: matrici.
- $X \sim p$: la variabile casuale X si distribuisce come p.
- $p(\cdot)$: distribuzione di massa o di densità di probabilità.
- $p(y \mid x)$: la probabilità o densità di y dato x, ovvero $p(y = Y \mid x = X)$.
- $f(x)$: una funzione arbitraria di x.
- $f(X; \theta, \gamma)$: f è una funzione di X con parametri θ, γ. Questa notazione indica che X sono i dati che vengono passati ad un modello di parametri θ, γ.
- $\mathcal{N}(\mu, \sigma^2)$: distribuzione gaussiana di media μ e varianza $sigma^2$.
- Beta(α, β): distribuzione Beta di parametri α e β.
- $\mathcal{U}(a, b)$: distribuzione uniforme con limite inferiore a e limite superiore b.
- Cauchy(α, β): distribuzione di Cauchy di parametri α (posizione: media) e β (scala: radice quadrata della varianza).
- $\mathcal{B}(p)$: distribuzione di Bernoulli di parametro p (probabilità di successo).
- Bin(n, p): distribuzione binomiale di parametri n (numero di prove) e p (probabilità di successo).
- $\mathbb{KL}(p \mid\mid q)$: la divergenza di Kullback-Leibler da p a q.

Bibliografia

Albert, J. and Hu, J. (2019). *Probability and Bayesian Modeling*. Chapman and Hall/CRC.

Horn, S. and Loewenstein, G. (2021). Underestimating learning by doing. *Available at SSRN 3941441*.

Johnson, A. A., Ott, M., and Dogucu, M. (2022). *Bayes Rules! An Introduction to Bayesian Modeling with R*. CRC Press.

Kruschke, J. (2014). *Doing Bayesian data analysis: A tutorial with R, JAGS, and Stan*. Academic Press.

Martin, O. A., Kumar, R., and Lao, J. (2022). *Bayesian Modeling and Computation in Python*. CRC Press.

McElreath, R. (2020). *Statistical rethinking: A Bayesian course with examples in R and Stan*. CRC Press, Boca Raton, Florida, 2nd edition edition.

van de Schoot, R., Depaoli, S., King, R., Kramer, B., Märtens, K., Tadesse, M. G., Vannucci, M., Gelman, A., Veen, D., Willemsen, J., and Yau, C. (2021). Bayesian statistics and modelling. *Nature Reviews Methods Primer*, 1(1):1–26.

Zetsche, U., Bürkner, P.-C., and Renneberg, B. (2019). Future expectations in clinical depression: Biased or realistic? *Journal of Abnormal Psychology*, 128(7):678–688.