

1 Metodi per la generazione casuale di numeri

Si sceglie di generare numeri casuali con delle simulazioni che hanno come vantaggio riproducibilità e flessibilità, come svantaggio la lentezza e la non precisione assoluta.

I numeri casuali hanno come caratteristiche desiderabili:

- Riproducibilità, ossia la possibilità di riottenere esattamente la stessa sequenza di numeri su macchine differenti.
- I tempi di riproducibilità della serie devono essere sufficientemente brevi
- La serie prodotta deve essere sufficientemente lunga

1.1 Metodo Middle-Square di von Neumann

Per generare un numero casuale di N cifre:

- Si parte da un numero x (seed) di N cifre scelto arbitrariamente.
- Lo si eleva al quadrato: x^2
- Si considerano le N cifre centrali di x^2 .

E' un metodo applicato iterativamente su ogni numero ottenuto per produrre il successivo.

In altre parole si prende un numero di dimensione N per poi generare un altro numero di dimensione N casuale da esso, e così via.

Ci si aspetta che la manipolazione algebrica effettuata possa produrre effettivamente dei numeri casuali, e sia quindi in grado di nascondere la sistematicità di generazione e la dipendenza tra i numeri dovuta all'iterazione. Questo, però, non accade; infatti il metodo è troppo sensibile al punto di partenza x . Inoltre, si ha un caso limite per cui se si ottiene 0 il generatore non riesce ad aggiungere un nuovo numero alla sequenza.

1.2 Metodo congruenziale misto (Lhemer)

Idea: il resto di una divisione si comporta in modo casuale.

Generatori congruenziali: $y = x \bmod m$, con x e m interi.

y è il resto della divisione tra x e m , $y = 0, 1, \dots, m - 1$ E' un algoritmo iterativo.

Si ha: $u_{j+1} = \frac{x_{j+1}}{m}$ con $u_j \in [0, 1]$

con $x_{j+1} = (ax_j + c) \bmod m$, con $m \in N$, $(a, c) \in \{0, 1, \dots, m - 1\}$.

Dove:

- x_0 : seme iniziale (seed) ($0 \leq x_0 < m$)
- a : moltiplicatore ($0 < a < m$)
- c : incremento ($0 \leq c < m$)
- m : modulo ($m > 0$)
- l'operazione $\bmod m$ consiste nel dividere un numero per m e prendere in considerazione il resto della divisione.
- Quindi x_{j+1} potrà essere al massimo $m - 1$

La serie che si ottiene contiene al massimo m numeri distinti, poi tenderanno a ripetersi, questo è un problema legato alla scelta dei parametri.

Quindi:

- m, a, c vengono scelti per massimizzare: ciclo, velocità e "casualità" dei valori u_j , cioè evitare le ripetizioni.
- il ciclo è il numero di u_j distinti che si ottengono prima di ripetere nuovamente la stessa frequenza.
- La massima lunghezza possibile è m

2 Verifica della casualità

2.1 Test Empirici

- Confronto tra momenti empirici e teorici:
per stabilire se i numeri generati provengono da una certa variabile casuale si possono confrontare media e varianza teorica della V.A. con media e varianza campionarie.
- Confronto tra distribuzione empirica e teorica:
si usano degli strumenti grafici, ad esempio si può confrontare la curva teorica della V.A. con l'istogramma delle densità di frequenza ottenuto, oppure si possono confrontare le funzioni di ripartizione.
- Verifica dell'assenza di autocorrelazione della serie:
con *acf* in R è possibile valutare delle eventuali autocorrelazioni tra i valori

2.2 Test di casualità: test del Chi-quadrato di adattamento

E' un test utilizzato per verificare la casualità di una serie di numeri, è un test non parametrico.

x_1, x_2, \dots, x_n è la sequenza di numeri generati da testare.

L'ipotesi nulla è data da $H_0 : F(X) = x$, con $x \in [0, 1]$, nel nostro caso:

$H_0 : X \sim U(0, 1)$, quindi il campione proviene da una Uniforme in $(0, 1)$

Operativamente si seguono i seguenti passaggi:

- Si divide l'intervallo $(0, 1)$ in k (con k fisso) sotto intervalli disgiunti I_j , con $j = 1, \dots, k$ di lunghezza $1/k$, tali per cui:
 - $I_j \cap I_l = \emptyset$ per $j \neq l$.
 - $\bigcup_{j=1}^k I_j = [0, 1]$
- Si confrontano le frequenze osservate n_j , $j = 1, \dots, k$, (cioè il conteggio dei numeri che ricadono in quel generico intervallo), con le frequenze attese n/k (sotto H_0).
Mi aspetto che i numeri siano equidistribuiti tra i sottointervalli:
 $E_j = n * P(x \in I_j) = n \int_{I_j} dx$ (frequenze attese)
 $O_j = n_j = \#(x_i \in I_j), i = 1, \dots, n$ (frequenze osservate)
- Statistica test: $\sum_{j=1}^k \frac{(O_j - E_j)^2}{E_j} \approx \chi_{k-1}^2$
- Fissato un livello di significatività α del test, si rifiuta H_0 se $X_{oss}^2 > X_{k-1, \alpha}^2$. In altri termini, un valore molto elevato della statistica test suggerisce l'idea che l'ipotesi nulla sia poco verosimile, ovvero indica una forte evidenza contro l'ipotesi nulla.

2.2.1 Caso semplice

Supponiamo di creare intervalli tutti della stessa ampiezza $1/k$.

Allora, si ha:

$$E_j = n \frac{1}{k} = \frac{n}{k}, \forall j \quad j = 1, \dots, k$$

$$\text{Statistica test: } X^2 = \frac{k}{n} \sum_{j=1}^k \left(n_j - \frac{n}{k}\right)^2$$

2.2.2 Difetti del test

- La scelta del numero di sottointervalli k influenza la statistica test
- Il test è asintotico, di fatto l'approssimazione vale se e solo se $E_j > 5$
- Discretizzazione dell'intervallo in $[0, 1]$, questo test infatti non è molto indicato per le funzioni continue.

Il singolo risultato vale per la specifica sequenza. Quindi è necessario svolgere il test più volte per valutare la validità dell'algoritmo.

In R si può usare la funzione *chisq.test*, che corrisponde a: *pchisq(Chi2, df = k - 1, lower.tail = F)*. Ovviamente, se il valore è alto accettiamo H_0 .

2.3 Test di casualità: test di Kolmogorov-Smirnov

Il test confronta la funzione di densità empirica con quella teorica attraverso un test, in particolare la distanza tra le due. Il test KS è non parametrico ed è valido per variabili continue.

Statistica test di KS:

$$D_n = \sup \left| F_n(x) - F(x) \right|, -\infty < x < +\infty$$

Sappiamo che la distribuzione di D_n è indipendente dalla distribuzione di X . Quindi il test va bene qualsiasi sia la X di partenza.

Infatti la frazione di valori campionari minori di un certo valore x è uguale alla frazione di valori $F(X_i) \leq F(x)$, dato che F è una funzione monotona crescente.

Nel nostro caso la variabile casuale $F(X_i) \equiv U_i \sim U(0, 1)$.

Possiamo allora scrivere:

$$D_n = \sup_{0 \leq u \leq 1} \left| \frac{\#(i : U_i \leq u)}{n} - u \right|$$

Da qui segue che:

- D_n è la distribuzione che si ottiene estraendo a caso n valori dalla densità uniforme e trovando il valore massimo della differenza scritta sopra.
- D_n è universale, non parametrica e dipende solo da n , cioè non dipende da X e la sua forma.

2.3.1 Calcolo del p-value

- n non elevato: ci sono dei valori tabulati di questa statistica test
- n elevato: è necessario conoscere la distribuzione della statistica test D_n

Abbiamo il seguente risultato asintotico:

$$\lim_{n \rightarrow \infty} \Pr(\sqrt{n}D_n \leq d) = 2 \sum_{j=1}^{\infty} (-1)^{j-1} e^{-2j^2 d^2} = H(d), \quad d > 0.$$

Fissato un livello di significatività α del test, il valore critico d^* è la soluzione dell'equazione:

$$\alpha \approx H(d^*) : \text{ se } \sqrt{n}D_n \leq d^*, \text{ allora si deve rifiutare } H_0 \text{ a livello } (1 - \alpha)$$

Consideriamo la serie di valori: $H(d) = 2e^{-2d^2} - 2e^{-8d^2} \pm \dots$. Se $d = 1$, il primo elemento della serie è $2e^{-2} \approx 0.27067$, il secondo $-2e^{-8} \approx -0.00067$. Notiamo quindi che la serie converge molto velocemente perciò è possibile approssimarla con il primo elemento: $H(d) = 2e^{-2d^2}$.

Al fine di trovare un valore critico d^*_α si avrà che: $\Pr(D_n \geq \frac{d}{\sqrt{n}}) \approx \alpha$

Se $\alpha \leq 0.2$ (cioè $\Pr(\sqrt{n}D_n \leq d) \approx 0.8$), una buona approssimazione è $2e^{-2d^2} \approx \alpha$, e risolvendo rispetto a d otteniamo:

$$d^*_\alpha \approx \sqrt{\frac{1}{2} \ln \left(\frac{2}{\alpha} \right)}.$$

Valori critici di $\sqrt{n}D_n$ sono ($n \rightarrow \infty$):

- $d^*_{0.05} = 1.358$
- $d^*_{0.01} = 1.628$
- $d^*_{0.001} = 1.950$

3 Inversione della funzione di ripartizione

3.1 Generazione di variabili casuali continue

Per generare valori da variabili aleatorie possiamo integrare la funzione di densità e poi invertire l'integrale trovato cioè invertire la funzione di ripartizione; questo perchè la funzione di ripartizione sta nell'intervallo $[0,1]$. In ciò consiste l'algoritmo **ITM**.

Per generare delle determinazioni casuali da una variabile casuale continua Y , di densità $f(y)$ e definita nell'intervallo (a, b) , occorre:

- Generare a caso un numero u da $U \sim U(0, 1)$.
- Risolvere, rispetto a y , l'equazione: $u = \int_a^x f(t) dt$ quindi $y = F^{-1}(u)$

3.2 Generazione di variabili casuali discrete

Data una variabile casuale discreta Y , che può assumere un insieme finito di valori y_1, y_2, \dots, y_k con probabilità rispettivamente p_1, p_2, \dots, p_k , tali che $\sum_{i=1}^k p_i = 1$, per invertire occorre costruire la funzione cumulativa $F(y_j)$: $F(y_j) = \sum_{i=1}^j p_i$.

Si noti che in tal caso il metodo di inversione si basa sulla definizione di inversa generalizzata $Y = F^{-1}(U) = \min\{y : F(y) \geq u\}$, dove $u \in [0, 1]$.

Per agire operativamente consideriamo l'intervallo $[0,1]$ e suddividiamolo in k intervalli disgiunti, assegnando ad ognuno di essi una lunghezza p_j pari alla probabilità che si verifichi il corrispondente evento $\{Y = y_j\}$. Quindi il numero casuale generato viene individuato dal sottointervallo entro cui cade il numero casuale u .

Poiché la probabilità che un numero casuale uniforme u cada in un particolare intervallo è esattamente pari alla lunghezza dell'intervallo:

$$\begin{aligned}\Pr(0 < U \leq p_1) &= p_1 \\ \Pr(p_1 < U \leq p_1 + p_2) &= p_2 \\ &\dots \\ \Pr(p_1 + p_2 + \dots + p_{k-1} < U \leq 1) &= p_k\end{aligned}$$

si dovrà considerare estratto il valore y_j corrispondente all'estremo superiore $p_1 + p_2 + \dots + p_j$ dell'intervallo entro cui è contenuto il numero u generato con la funzione `runif()`.

Per fare ciò l'algoritmo **ITM** si compone nel seguente modo.

- Generare un numero u da $U \sim U(0, 1)$.
- Trovare l'indice j ($j = 1, \dots, k$) per cui è verificata la disuguaglianza: $F(y_{j-1}) < u \leq F(y_j)$ (quando $j = 1$, definiamo $F(y_{j-1}) = 0$).
- Quindi il valore è y_j

Questi algoritmo possono essere combinati tra loro per esempio una somma di Bernoulli può diventare una binomiale. Le Bernoulli sono generate in maniera indipendente tramite ITM e poi sommate.

4 Algoritmo Accettazione-Rifiuto (AR)

In alcuni casi non è possibile invertire le funzioni per ricavarne i loro valori casuali e non sempre è possibile usare trasformazioni per ricavarsi determinati valori. Ecco perchè si vanno ad utilizzare i **metodi indiretti** che richiedono solo di conoscere la forma funzionale della densità target $f_Y(y)$ di interesse ed una costante moltiplicativa. Tra questi c'è l'algoritmo Accettazione-Rifiuto.

Se f è la densità di interesse ed appare come la densità marginale (in Y) di $f_{y,u}$ cioè $f_y = \int_0^{f(y)} du$ della distribuzione congiunta

$$(Y, U) \sim U\{(y, u) : 0 < u < f(y)\}$$

dove U non essendo direttamente legata al problema originale, è detta variabile ausiliaria.

Come si nota la densità congiunta di Y e U è uniforme. Quindi per generare da questa distribuzione congiunta possiamo semplicemente generare variabili casuali uniformi nello spazio vincolato $\{(y, u) : 0 < u < f(y)\}$. Inoltre, poiché la distribuzione marginale di Y è la distribuzione target originale, generando una variabile uniforme nello spazio $\{(y, u) : 0 < u < f(y)\}$ abbiamo generato (indirettamente) una variabile casuale da f . Si noti che questa generazione è stata prodotta senza utilizzare f se non attraverso il calcolo di $f(y)$.

Teorema fondamentale della simulazione: Simulare $Y \sim f(y)$ è equivalente a simulare $(Y, U) \sim U\{(y, u) : 0 < u < f(y)\}$. Poiché spesso non è semplice simulare dalla distribuzione congiunta nello spazio vincolato, per aggirare l'ostacolo una strategia è quella di simulare l'intera coppia (Y, U) in un insieme più grande, dove è più facile simulare, e poi considerare valida la coppia di numeri generata solo se il vincolo è soddisfatto.

L'obiettivo è quello di generare delle determinazioni y da una variabile casuale Y con densità $f_Y()$ (densità target). Si supponga di poter generare da una variabile casuale X con densità $g_X()$, detta densità strumentale, e da una variabile casuale $U \sim U(0, 1)$. Si supponga inoltre che il supporto di Y è contenuto in quello di $g()$.

Sia: $\sup \frac{f_Y(y)}{g_X(y)} = b < \infty$ e $g_X(y)$ si noti che $b \geq 1$ e che $f_Y(y) \leq b g_X(y), \forall y$.

L'algoritmo prende forma e si può scrivere come

- Genero indipendentemente u e v indipendentemente da $U(0, 1)$
- Genero x da $g_X()$ attraverso un algoritmo ITM da v .
- Effettua il test: se $u \leq \frac{f_Y(x)}{b g_X(x)}$, accetta $y = x$, altrimenti scarta x e ritorna al passo 1.

Sappiamo che:

- Il rapporto $0 < \frac{f(X)}{b g(X)} \leq 1$, è maggiore di 0 visto che è frazione di densità quindi funzioni positive.
- $\frac{1}{b}$ è il tasso di accettazione dei valori generati. Valori bassi di b corrispondono a schemi di campionamento efficienti.
- Il numero di volte N in cui è necessario eseguire i passi 1 e 2 dell'algoritmo prima di ottenere un primo successo (ovvero il numero di iterazioni necessarie prima di accettare un valore y) è a sua volta una variabile casuale con distribuzione Geometrica con probabilità di successo $\frac{1}{b} = P(U \leq \frac{f(X)}{b g(X)})$. La sua funzione di probabilità è data da: $P(N = n) = (1 - p)^{n-1}$, ($n \geq 1$). Pertanto, in media, il numero di iterazioni necessarie prima di ottenere un candidato valido dalla densità target è pari a $E(N) = b$.

4.1 Dimostrazione

Dobbiamo dimostrare che $P(X \leq y | U \leq \frac{f(X)}{bg(X)}) = F(y)$ dove $F(y)$ è la funzione di ripartizione di y .

Sappiamo che per le proprietà dei valori attesi condizionati $P(A|B) = \frac{P(B|A)*P(A)}{P(B)}$ quindi:

$$P(X \leq y | U \leq \frac{f(X)}{bg(X)}) = \frac{P(U \leq \frac{f(X)}{bg(X)} | X \leq y) * P(X \leq y)}{P(U \leq \frac{f(X)}{bg(X)})}$$

sappiamo che $P(U \leq \frac{f(X)}{bg(X)}) = 1/b$ e che $P(X \leq y) = G(y)$

inoltre sappiamo che $P(U \leq \frac{f(X)}{bg(X)} | X \leq y) = \frac{P(U \leq \frac{f(X)}{bg(X)}, X \leq y)}{G(y)}$ dove il numeratore può essere anche

scritto come $\int_{-\infty}^y \frac{f(x)*g(x)}{b*g(x)} dx$ quindi mi rimane $F(y)/b$ da qua se uniamo i componenti abbiamo

$\frac{F(y)*G(y)*b}{b*G(y)} = F(y)$ quindi abbiamo dimostrato che è corretto.

5 Metodi ad hoc per la generazione dalla v.c. Normale

Osserviamo che per la generazione di determinazioni da variabili casuali normali con generica media μ e varianza $\sigma^2 < +\infty$ ($Y \sim N(\mu, \sigma^2)$), basta disporre di un algoritmo per generare un valore z da una densità normale standard ($Z \sim N(0, 1)$) ed effettuare la trasformazione lineare: $y = \mu + \sigma z$

5.1 AR: Esempio half normal

Se la densità target: $Y = |Z|$ con $Z \sim N(0, 1)$. Si noti che $f(y) = 2\phi(y)$, ($y > 0$), con $\phi(\cdot)$ funzione di densità di una normale standard. Allora posso usare il seguente algoritmo:

- Genera due numeri casuali u_1 e u_2 .
- Genera una determinazione x indipendente da una Esponenziale di parametro 1 mediante la trasformazione: $-\log(u_1) = x$.
- Se $u_2 \leq \frac{f(x)}{b * g(x)}$, accetta $y = x$ quale determinazione della variabile casuale $Y = |Z|$; altrimenti rifiuta e ricomincia dal passo 1.
- Qualora si accetti $y = x$, si procede assegnando casualmente a y il segno ± 1 con probabilità 0.5. $z = \pm 1 \cdot y$ risulta essere una determinazione (esatta) di $Z \sim N(0, 1)$.

5.2 Teorema limite centrale

Visto che X_1, \dots, X_n n variabili casuali indipendenti e identicamente distribuite (i.i.d.), con $E[X_i] = \mu$ e $Var[X_i] = \sigma^2 < +\infty$, per $i = 1, \dots, n$ e $P_n(\sum_{i=1}^n X_i) \approx N(n\mu, n\sigma^2)$. Allora posso generare normali come somma di uniformi indipendenti

- Genera u_1, u_2, \dots, u_{12} da 12 variabili casuali uniformi in $(0, 1)$ indipendenti.
- Calcola $y = \sum_{i=1}^{12} u_i - 6$. La variabile casuale y così generata è una determinazione da una normale standard.

Ha dei limiti evidenti questo processo infatti la normale è compresa tra -6 e 6 e non tutto R, per creare un osservazione ne servono almeno 12. Ciò lo rende lento e grezzo.

5.3 Metodo di Box Miller

Se prendiamo due variabili normali indipendenti hanno formula: $f(Z_1, Z_2) = \frac{1}{2\pi} \exp\left(-\frac{Z_1^2 + Z_2^2}{2}\right)$.

Ora esprimiamo in coordinate polari $z_1 = r \cos(\theta)$ e $z_2 = r \sin(\theta)$

Sappiamo che $r = \sqrt{z_1^2 + z_2^2}$ e che il determinante della matrice jacobiana per la sostituzione ha risultato r . In questo modo se usiamo il teorema del difeomorfismo otteniamo che $f(r, \theta) = \frac{r}{2\pi} \exp\left(-\frac{r^2}{2}\right)$ da qui riusciamo a capire che $R^2 \sim \text{Exp}\left(\frac{1}{2}\right)$ e $\Theta \sim \text{Unif}(0, 2\pi)$ sono indipendenti, da qua basta usare un algoritmo che conosciamo.

Algoritmo:

- Genera u_1 e u_2 da due variabili casuali indipendenti con distribuzione uniforme nell'intervallo $(0, 1)$: $u_1, u_2 \sim \text{Unif}(0, 1)$
- Calcola r e θ come segue: $r = \sqrt{-2 \ln(u_1)}$ è stato ottenuto con l'ITM di EXP, poi calcolo $\theta = 2\pi u_2$
- Quindi, calcola z_1 e z_2 come segue:

$$z_1 = r \cos(\theta) = \sqrt{(-2 \ln(u_1))} * \cos(2\pi u_2)$$

$$z_2 = r \sin(\theta) = \sqrt{(-2 \ln(u_1))} * \sin(2\pi u_2)$$

In questo modo otteniamo due osservazioni da normali indipendenti.

5.4 Metodo di Marsaglia

Questo metodo sfrutta come prima le coordinate polari però viene aggiunto un test di accettazione rifiuto, per quanto sia contro intuitivo è l'algoritmo più veloce.

Data la condizione di accettazione $v_1^2 + v_2^2 \leq 1$, la coppia (V_1, V_2) si distribuisce uniformemente nel cerchio di raggio unitario e ha pertanto densità congiunta $(f(v_1, v_2) = \frac{1}{\pi})$. Infatti generiamo un uniforme su un cerchio tra tra -1 e 1.

La densità congiunta r e θ avrà formula $f(r, \theta) = \frac{r}{\pi}$ notiamo che r e θ sono indipendenti, la congiunta si è ottenuta con il teorema del difeomorfismo ed ha lo stesso determinante del metodo di prima. Inoltre sappiamo che $S = V_1^2 + V_2^2 = R^2$ ha distribuzione $U(0, 1)$, dato che $P(S \leq s) = s$ in quanto è pari al rapporto tra l'area del cerchio di raggio inoltre sappiamo che $S = r^2$ quindi è giusto che la sua marginale sia $2r$.

Algoritmo:

- Genero u_1 ed u_2 da $U(0, 1)$.
- Pongo $v_i = 2u_i - 1$, ovvero $V_i \sim U(-1, 1)$, per $i = 1, 2$.
- Se $s = v_1^2 + v_2^2 > 1$, rifiuto e torno al punto 1. Altrimenti, se accetto, (V_1, V_2) è uniforme nel cerchio unitario.
- Condizionatamente all'accettazione del passo 2, si pongono:

$$z_1 = r \cos(\theta) = v_1 \sqrt{\frac{-2}{s} \log s}$$

$$z_2 = r \sin(\theta) = v_2 \sqrt{\frac{-2}{s} \log s}$$

Questo deriva dalla trasformazione infatti $\cos(\theta) = v_1/r$ e $\sin(\theta) = v_2/r$ dove $r = \sqrt{s}$ e la formula la recuperiamo da quella di Box e Miller

Questo algoritmo ha come tasso di accettazione $\pi/4$ cioè area del cerchio unitario fratto area del quadrato con lato di dimensione 2.

5.5 Metodo di creazione di normali correlate

Dato un insieme di normale $Y = (Y_1, \dots, Y_p)^T \sim \mathcal{N}_p(\mu, \Sigma)$, con $\mu = E[Y] = (E[Y_1], E[Y_2], \dots, E[Y_p])^T$, e la matrice di varianze e covarianze Σ di dimensione $p \times p$ possiamo usare la **decomposizione di Cholesky** e ottengo che $\Sigma = AA^T$ si ha: $Y \sim \mathcal{N}(0, I_p) \Rightarrow AY + \mu \sim \mathcal{N}_p(\mu, \Sigma)$. In R, `t(chol(Sigma))` calcola la matrice A , dove sigma è la Σ . In questo modo possiamo creare osservazioni di normali multivariate non indipendenti.

6 Rapporto di Uniformi

Se $h(x)$ è una funzione non negativa con integrale finito, se considero l'insieme

$$C_h = \{(u, v) : 0 \leq u \leq \sqrt{h(v/u)}\}$$

allora tale insieme C_h ha area finita. Se (U, V) è uniforme $U(0, 1)$ in C_h , allora $X = V/U$ ha densità

$$f(x) = \frac{h(x)}{\int h(y) dy}$$

Se K è il valore dell'area di C_h allora facendo il cambio di variabile $(u, v) \rightarrow (u = w, v = x * w)$ con il teorema del difeomorfismo e con determinate Jacobiano pari a w , si ha che l'area di C_h è

$$\iint_{C_h} dudv = \iint_0^{\sqrt{h(x)}} wdw dx = \frac{1}{2} \int h(x) dx = K < \infty$$

Inoltre, se (U, V) è uniforme in C_h , la sua densità è $f(u, v) = \frac{1}{K}$ essendo l'area dell'integrale. Facendo il cambio di variabile (W, X) , si ottiene $f(w, x) = \frac{1}{K}w$ e quindi posso marginalizzare per ottenere la funzione di distribuzione di X .

$$f(x) = \int_0^{\sqrt{h(x)}} f(w, x) dw = \frac{1}{K} \int_0^{\sqrt{h(x)}} wdw dx = \frac{h(x)}{\int h(y) dy}.$$

Quindi, se si riesce a generare da una variabile casuale uniforme in C_h , allora V/U è una variabile casuale con densità $\frac{h(x)}{\int h(y) dy}$.

A volte può essere difficile generare da una variabile casuale uniforme in C_h . Si può allora combinare questo metodo con il metodo AR. Se $h(x)$ e $x^2h(x)$ sono limitate, si può dimostrare che $C_h \subset [0, a] \times [b^-, b^+]$, dove:

- $a = \sup_x \sqrt{h(x)}$, $x \in R$
- $b^- = \inf_x x\sqrt{h(x)}$, $x \leq 0$
- $b^+ = \sup_x x\sqrt{h(x)}$, $x \geq 0$

Quindi l'**algoritmo** diventa:

- Genera (u, v) da due variabili casuali uniformi
- trasformo le variabili $U = a * u$ e $V = b^- + (b^+ - b^-) * v$.
- Se $(U, V) \in C_h$, cioè se $u^2 \leq h(x)$ restituisci $x = V/U$, altrimenti ritorna al passo 1.

7 Metodi Monte Carlo

I termini metodi Monte Carlo o simulazioni Monte Carlo sono usati per far riferimento a tecniche che coinvolgono la simulazione al computer.

Se prendiamo $\int_D f(x) dx = \psi$ e una variabile aleatoria Y tale che $E(m(Y)) = \psi$ dove però $E(m(Y)) = \int_D m(y)g(y) dy = \psi$.

Se Y_n i.i.d allora lo stimatore per ψ è $\hat{\psi} = \frac{\sum_{i=1}^R m(y_i)}{R}$, quindi corrisponde alla media campionaria. Lo stimatore ha due importanti proprietà :

- **Non distorsione** infatti $E(\hat{\psi}) = E(\frac{\sum_{i=1}^R m(y_i)}{R}) = E(m(y)) = \psi$
- **Varianza** che cade a 0 se $n \rightarrow \infty$. Infatti $Var(\hat{\psi})$ essendo Y indipendenti si possono sommare le varianze quindi $= \frac{Var(m(y))}{R} = \frac{E(m(y)-E(m(y)))^2}{R}$ quindi la sua $Var(\hat{\psi}) = \frac{\sum_{i=1}^R (m(y_i) - \overline{m(y)})^2}{R-1} * R^{-1}$. Notiamo subito quindi che la velocità di caduta dello SE è un O piccolo di \sqrt{R} .
- **Normalità** essendo i.i.d le Y vale il teorema centrale del limite per lo stimatore $\hat{\psi}$ quindi possiamo trovare le sue bande di confidenza con i valori di una normale quindi $\hat{\psi} \pm Z_{\alpha/2} * \sqrt{Var(\hat{\psi})}$.

Operativamente andremo a seguire il seguente algoritmo:

- Genero Y_i da $U(0, 1)$
- Applico la trasformazione su Y e calcolo lo stimatore $\hat{\psi} = \frac{\sum_{i=1}^R m(y_i)}{R}$

Il metodo può essere usato anche per capire la probabilità di rifiutare un certo valore da una particolare distribuzione. Lo si può fare generando un insieme di valori casuali e confrontando se esse sono minori di tale valore e dividerle per il numero di valori generati.

7.1 Variabili di controllo

L'algoritmo di monte Carlo spesso non è efficiente e per renderlo tale si inserisce la variabile di controllo. Immaginiamo ora di avere $n(y) \sim m(y)$ con $E(n(y)) = \mu$ nota.

Se le coppie di $(n(y_i), m(y_i))$ sono indipendenti tra di loro e se se disponiamo come $m(y_i) - c(n(y_i) - \mu)$, con c che appartiene ai reali. Troveremo lo stimatore $\hat{\psi}_c$ che è uguale alla media campionaria delle osservazioni quindi $\frac{\sum_{i=1}^R m(y_i) - c(n(y_i) - \mu)}{R} = \overline{m(y_i)} - c(\overline{n(y_i)} - \mu)$ e troviamo anche la misura di errore che moltiplica c .

Questo stimatore ha le seguenti proprietà

- **Non distorsione** infatti $E(m(y_i) - c(n(y_i) - \mu)) = E(m(y_i)) - c(E(n(y_i)) - \mu) = E(m(y_i)) = \psi$
- **Varianza** tende a 0 ed è minore o uguale di quella con approccio classico infatti $Var(m(y_i) - c(n(y_i) - \mu)) = Var(m(y_i)) - 2*c*Cov(m(y_i), n(y_i)) + c^2*Var(n(y_i))$ e li chiameremo $\frac{\sigma_y^2 + c^2*\sigma_x^2 - 2*c*\sigma_{xy}}{R}$ noi vogliamo trovare il c che minimizza la varianza che con una semplice derivata troviamo che $c = \frac{\sigma_{xy}}{\sigma_x^2}$ quindi $Var(\hat{\psi}) = \frac{\sigma_y^2 - \frac{\sigma_{xy}^2}{\sigma_x^2}}{R}$ e da qui si può notare che essendo differenza tra positivi è minore della varianza dello stimatore non controllato. Quindi inserire il controllo notiamo che migliora drasticamente la varianza.
- La varianza migliore alla velocità di $\frac{1}{1-\rho_{xy}^2}$, questo si ottiene dividendo la varianza e la varianza con controllo, notiamo che più la correlazione tra le due funzioni più è forte più la varianza diminuirà velocemente.

A volte C però non è noto o calcolabile analiticamente ecco perchè andiamo a stimarlo usando covarianza e varianza stimata quindi $c = \frac{\sigma_{xy}}{\sigma_x^2}$, però per calcolarlo usiamo un campione esplorativo generato indipendentemente da Y .

Operativamente seguiamo il seguente algoritmo:

- Genero y da una uniforme $0,1$
- se c'è bisogno calcolo c con i due metodi che conosciamo se no salto questo passo
- Calcolo lo stimatore come $\frac{\sum_{i=1}^R m(y_i) - c(n(y_i) - \mu)}{R} = \overline{m(y_i)} - c(\overline{n(y_i)} - \mu)$

7.1.1 Interpretazione lineare

Possiamo osservare che c ha una forte analogia con β_1 della regressione lineare infatti $\beta_1 = c$ e $\beta_0 = \overline{m(y)} - \hat{c} * \overline{n(y)}$ allora troveremo che la regressione lineare su μ sarà $\beta_0 + \beta_1 * \mu = \hat{\psi}_c$. Quindi possiamo calcolare c anche come coefficiente della regressione lineare tra $m(y) \sim n(y)$.

7.2 Variabili di controllo multiple

Se immaginiamo che ogni funzione $m(y_i)$ è in grado di generare $n_k(y_i)$ funzioni simili, se usiamo queste funzioni tutte come funzioni di controllo allora rendiamo ancora più efficiente l'algoritmo.

In questo caso avremmo c come vettore di numeri stimabile in modo facile come coefficiente β_1, \dots, β_n da una regressione lineare multi variata.

Trovando così il nuovo stimatore come $\frac{\sum_{i=1}^R m(y_i) - \sum_{k=1}^N c_k(n_k(y_i) - \mu_k)}{R}$.

7.3 Variabili antitetiche

Due variabili sono dette antitetiche se sono identicamente distribuite ed hanno correlazione negativa. Sono utili per la correzione del metodo di Monte Carlo.

Ipotizziamo che la variabile uniforme U sottostimi lo stimatore $\hat{\psi}$, allora introduciamo la variabile $1 - U$ che da costruzione è correlata negativamente con U allora la nuova variabile andrà a sovrastimare $\hat{\psi}$; quindi per trovare la stima più corretta andiamo a fare le medie dei due stimatori per formarne uno più preciso.

Questo ragionamento vale anche con le trasformazioni infatti se prendiamo F funzione monotona allora $F^{-1}(U)$ e $F^{-1}(1 - U)$ sono antitetiche. Sono considerabili come le funzioni inverse id un ITM.

Se $m(y)$ è monotona allora anche $Z = m(F^{-1}(U))$ e $Z_* = m(F^{-1}(1 - U))$ rimangono antitetiche così da rendere l'algoritmo di Monte Carlo con le trasformazioni applicabile.

Lo stimatore sarà quindi $\hat{\psi}_a = \frac{\sum_{i=1}^R z(Y_r) + z^*(Y_r)}{2R}$, lo stimatore ha le seguenti proprietà:

- **Non distorsione** infatti $E(\frac{\sum_{i=1}^R z(Y_r) + z^*(Y_r)}{2R}) = \frac{E(z(Y_r)) + E(z^*(Y_r))}{2}$ essendo identicamente distribuiti hanno lo stesso valore atteso ψ quindi il valore atteso è ψ
- **Varianza** tende a 0 ed è minore o uguale di quella con approccio classico infatti $Var(\frac{\sum_{i=1}^R z(Y_r) + z^*(Y_r)}{2R}) = \frac{Var(z(Y_r) + z^*(Y_r))}{4R}$ che essendo identicamente distribuiti avranno varianza uguale, ma essendo correlati negativamente avranno covarianza negativa quindi otteniamo $= \frac{2Var(z(Y_r)) - 2\rho * Var(z(Y_r))}{4R} = \frac{Var(z(Y_r)) - \rho * Var(z(Y_r))}{2R}$, notiamo quindi che essendo differenza di positivi la varianza risulta minore all'originale
- Anche in questo caso la velocità di discesa della varianza dipende da quanto sono negativamente correlate le variabili.
- Se volessimo confrontare l'efficienza di questo metodo con gli altri dobbiamo ricordarci di usare come sample campionario due volte l'ampiezza del numero di simulazioni per gli altri metodi.

7.4 Iterazioni del metodo Monte Carlo

Una valutazione più generale e corretta si basa sulla simulazione ripetuta del metodo di Monte Carlo. Si ripete la stima per un numero N di serie parallele di lunghezza R e si valuta la variabilità delle stime. Chiaramente questo presuppone la generazione di $N \cdot R$ valori simulati ed un costo computazionale più oneroso, a volte questa può essere l'unica via semplice per valutare la variabilità nella stima.

8 Metodo Bootstrap

E' un metodo non parametrico quindi non fa assunzioni di alcun tipo e che si basa su tre idee principali:

- Abbiamo un campione di dati e rendiamo tale campione l'intera popolazione
- Generiamo B campioni dalla popolazione F , i campioni vengono generati pescando n valori della popolazione, però ogni valore viene poi reinserito nella popolazione; quindi due valori possono ripetersi nel campione.
- Approssimare con \hat{F}_n la funzione di ripartizione dell'intera popolazione
- Usiamo il principio plugin per stimare θ cioè calcoliamo lo stimatore pensando che \hat{F}_n approssimi bene F .

Se supponiamo di dovere stimare un particolare parametro $\theta(x)$ possiamo fare:

- Generiamo B campioni bootstrap
- Stimiamo in ogni popolazione il parametro
- facciamo la media di tutti i parametri stimati in precedenza

Con questo metodo ignoriamo l'intera popolazione e ci concentriamo sui sottocampioni e pensiamo di stimare la popolazione con la funzione di ripartizione dei singoli sottocampioni.

In questo modo avrò che lo stimatore $\theta^* = \frac{\sum_{i=1}^B \theta(x)^{*i}}{B}$ che ha varianza $VAR(\hat{\theta}) = \frac{\sum_{i=1}^B (\theta(x)^{*i} - \theta^*)^2}{B-1}$. La varianza inizia a stabilizzarsi quando $B = 200$, sappiamo che se $B \rightarrow \infty$ allora la varianza Bootstrap converge a quella plugin.

Da una popolazione di n elementi possiamo avere al massimo $\binom{2n-1}{n}$ numero di sottocampioni.

Utilizzando questo algoritmo introduciamo due tipi di errori:

- Statistico nello stimare F della popolazione con \hat{F}_n dei sottocampioni composti da un numero di valori finiti.
- Di simulazione Monte Carlo visto che stimiamo F con un numero finito di sottocampioni.

Questi errori portano a una distorsione che viene calcolata come $b = \theta^* - \hat{\theta}$.

Dove $\hat{\theta}$ è chiamata stima plugin della popolazione cioè lo stimatore applicato sulla popolazione e non sui campioni.

La distorsione si dispone come $\beta(F) = E(\theta | y_1, \dots, y_n \sim F) - \theta(F)$.

Vale la pena correggere la stima se $b > 0.25 * \sqrt{VAR(\hat{\theta})}$.

Il metodo bootstrap crea una distribuzione per θ che è utile per stimare la variabilità, la forma della distribuzione dello stimatore plug-in e per fini inferenziali.

Però ogni distribuzione bootstrap è centrata sulla media campionaria del campione corrispondente anziché essere centrato sulla media della popolazione μ , ciò fa intuire che il metodo bootstrap non fornisce una stima migliore del parametro della popolazione μ , non importa quanti campioni vengono utilizzati, sono centrati su \bar{x} più una variazione casuale e non sono centrati su μ .

8.1 Limiti del metodo Bootstrap e Bootstrap parametrico

Il metodo non è sempre efficace infatti esso fallisce se la stima della funzione di ripartizione dipende da un parametro che varia a seconda del campione. Ecco che in queste situazioni usiamo il Bootstrap parametrico cioè conosciamo la distribuzione della popolazione F , quindi non vado più a campionare su di essa ma campiono più volte su una popolazione identica ma generata in maniera casuale da me. In questo modo vado a migliorare le analisi.

8.2 Bootstrap bivariato

Se la popolazione da analizzare ha due variabili e non una, non si possono campionare le due variabili in maniera indipendente perchè andremo a peggiorare il risultato. Infatti il sotto campione creato sarà formato da coppie estratte dalla popolazione in maniera congiunta.

8.3 Bootstrap come regressore lineare

Sappiamo che la regressione lineare sui residui ha delle forti assunzioni:

- media uguale a 0
- omoschedasticità dei residui
- incorrelazione dei residui
- per fare i test bisogna supporre anche che i residui si dispongano in maniera normale

Ecco che a volte usare il bootstrap conviene perchè non ha bisogno di tutte queste forti premesse.

8.3.1 Metodo delle unità statistiche(Random-resemping)

Questo metodo non ha alcuno bisogno di premesse sui residui. L'unica che facciamo è che le covariate x siano indipendenti. L'algoritmo si sviluppa nel seguente modo:

- Si creano $Z_i = (Y_i, X_{1i}, \dots, X_{ni})$ indipendenti tra di loro.
- Estraggo dalla popolazione B campioni di ampiezza n , il campionamento avviene sempre con reinserimento.
- Calcolo su ogni campione la statistica di interesse.
- Faccio una media campionaria con i valori ottenuti al passo prima quindi $\theta^* = \frac{\sum_{i=1}^B \theta(\hat{x})^{*i}}{B}$.

Lo stimatore trovato θ^* dovrebbe avere le stesse caratteristiche della cosa che stavamo cercando ed essendo un metodo Bootstrap possiamo valutare il bias calcolato come $b = \theta^* - \hat{\theta}$.

Il metodo è utile se per esempio ci sono pochi dati e vogliamo aumentare la numerosità campionaria.

8.3.2 Metodo dei residui (Fixed-resemping)

A volte non si possono ricampionare i valori escludendone alcuni a discapito di altri perchè tutti i dati selezionati sono utili e necessari ecco che si usa questo metodo.

Il metodo dei residui ha delle premesse più forti cioè che tutti i residui sono I.I.D. ad una funzione di densità.

L'algoritmo si sviluppa nel seguente modo:

- Si calcolano \hat{Y} con la regressione su tutta la popolazione disponibile
- Si trovano i residui $r = Y - \hat{Y}$, possiamo analizzare sia i residui normali che quelli standardizzati ($r_{std} = \frac{r}{SE \sqrt{1-H}}$, con $SE = \sigma$ cioè varianza residua, H valore j-esimo della matrice HAT).
- Creo $y^* = r^* + y$ con r scelto con il solito metodo di ripescaggio con reinserimento da tutti i residui disponibili, in questo modo si aggiunge randomicità al modello.
- Calcolo su ogni campione la statistica di interesse.
- Faccio una media campionaria con i valori ottenuti al passo prima quindi $\theta^* = \frac{\sum_{i=1}^B \theta(\hat{x})^{*i}}{B}$.

Con questo metodo però otterremo un basso bias e standard error molto vicini agli usuali standard error calcolati analiticamente.

8.4 Metodi Jack-Knife

Dato un campione $x = (x_1 \dots x_n)$ con x che si distribuisce come una $F(\theta)$ non nota, se vogliamo stimare un qualsiasi parametro $\hat{\theta}$ detta stima plugin (stimata su tutto il campione x), la stimiamo nel seguente modo:

- Stimiamo il parametro su tutta la popolazione escluso un valore quindi otterremo n stimatori chiamati $\hat{\theta}_{-i} = \frac{\sum_{j=1, i \neq j}^n x_j}{n-1}$
- Stimiamo poi $\hat{\theta}_{\cdot} = \frac{\sum_{i=1}^n \hat{\theta}_{-i}}{n}$.
- Calcoliamo il Bias $= (\hat{\theta}_{\cdot} - \hat{\theta}) * (n-1)$
- Quindi la stima corretta di $\hat{\theta}$ per il metodo Jackknife è $\theta_{jack}^{\hat{}} = \hat{\theta} - Bias = n\hat{\theta} - (n-1) * \hat{\theta}_{\cdot}$
- lo standard error per $\hat{\theta}$ secondo la stima Jackknife è $SE_{jack} = \sqrt{\sum_{i=1}^n (\hat{\theta}_{-i} - \hat{\theta}_{\cdot})^2 * \frac{n-1}{n}}$

Con questo metodo se $\hat{\theta}$ è non distorto allora il Bias è = a 0.

Infatti se $\hat{\theta} = \bar{x}$ allora $\hat{\theta}_{-i} = \frac{\sum_{j=1, i \neq j}^n x_j}{n-1} = \frac{\sum_{j=1}^n x_j - x_i}{n-1} = \frac{n\bar{x} - x_i}{n-1}$ da qua ricaviamo che $x_i = n\bar{x} - (n-1)\hat{\theta}_{-i}$
Quindi $\hat{\theta}_{\cdot} = \frac{\sum_{i=1}^n \hat{\theta}_{-i}}{n} = \frac{\sum_{i=1}^n \frac{n\bar{x} - x_i}{n-1}}{n}$ cioè $\frac{n\bar{x} - \bar{x}}{n-1} = \bar{x}$ in questo caso possiamo notare la non distorsione.

Si può dimostrare anche che la $Var(\hat{\theta}) = Var(\hat{\theta})_{jack}$ infatti $Var(\hat{\theta})_{jack} = \sum_{i=1}^n (\hat{\theta}_{-i} - \hat{\theta}_{\cdot})^2 * \frac{n-1}{n}$ grazie all'equazione precedente sappiamo che $\hat{\theta}_{\cdot} = \bar{x}$ e che $\hat{\theta}_{-i} = \frac{n\bar{x} - x_i}{n-1}$ quindi $Var(\hat{\theta})_{jack} = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{(n-1)*n}$ cioè $\frac{Var(\theta)}{n}$ come la varianza che otterremo nel modo classico.

8.4.1 Metodo degli pseudo-valori

Se diciamo che $t_i = n\hat{\theta} - (n-1)*\hat{\theta}_{-i}$ ci aspettiamo che i t siano delle stime di θ , inoltre notiamo che per come è costruito $t_i \approx x_i$ inoltre se calcoliamo la media campionaria di t $\bar{t} = \frac{\sum_{i=1}^n t_i}{n} = n\hat{\theta} - (n-1)*\hat{\theta}_{\cdot}$ che come possiamo notare è la stima $\theta_{jack}^{\hat{}}$ corretta.

Quindi se vogliamo lo $SE(\hat{\theta})_{jack} = \sqrt{Var(\bar{t})} = \sqrt{\frac{\sum_{i=1}^n (t_i - \bar{t})^2}{n*(n-1)}} = \sqrt{\sum_{i=1}^n (\hat{\theta}_{-i} - \hat{\theta}_{\cdot})^2 * \frac{n-1}{n}} = \sqrt{Var(\hat{\theta}_{-i}) * \frac{(n-1)^2}{n}}$

8.4.2 Metodo D-Jackknife

Funziona allo stesso modo del metodo jackknife però invece che togliere solo l'i-esima osservazione togliere un numero $D > 1$ di osservazioni. Facendo così però il numero di campioni jackknife creabili sono il binomio $\binom{n}{D}$ e ogni campione creato sarà di lunghezza $n - D$, ciò va a modificare la stima del parametro θ e dello standard error calcolato su θ .

$$SE(\hat{\theta})_{jack} = \sqrt{\sum_{i=1}^{\binom{n}{D}} (\hat{\theta}_i - \hat{\theta}_{\cdot})^2 * \frac{n-D}{D*\binom{n}{D}}}$$

Questo metodo viene usato solo se n è grande e $D \in [\sqrt{n}, n]$. Ha come principale obiettivo la risoluzione dei casi in cui θ non abbiamo una distribuzione smooth (per esempio nel caso $\theta = mediana$)

8.5 Quando scegliere metodo Bootstrap o Jackknife

Il metodo Jackknife non indaga sulla distribuzione dello stimatore e quindi ha uno standard error poco affidabile.

Questo metodo funziona bene solo per stimatori lineari perchè esso è un'approssimazione lineare al bootstrap, ovvero concorda con il bootstrap (ad eccezione di un fattore di $\sqrt{\frac{n-1}{n}}$), mentre è inadeguato se la statistica di interesse non è lineare.

8.5.1 Stima bootstrap per statistiche lineari

Se $\hat{\theta} = \sum_{i=1}^n \alpha(x_i)$ allora la stima bootstrap $\hat{\theta}^* = \frac{1}{n} \sum_{i=1}^n \alpha(x_i^*) = \frac{1}{n} \sum_{j=1}^n \alpha(x_j^*) * n_j$.

Dove n_j è la frequenza osservata di x_j nel campione bootstrap (x_1^*, \dots, x_n^*) , esso può assumere anche il valore 0. Poiché il ricampionamento di x_i avviene con probabilità $\frac{1}{n}$, la variabile $N = (N_1, \dots, N_n)$ segue una distribuzione multinomiale con vettore dei parametri $(1/n, \dots, 1/n, N)$.

Da qui ne consegue che $E(N_i) = 1$ e quindi $E(\hat{\theta}) = \sum_{i=1}^n \alpha(E(x_i)) * E(N_i) = \hat{\theta}$ quindi è uno stimatore non distorto come la stima Jackknife.

Inoltre sappiamo che $\text{Var}(N_i) = 1 - 1/n$ $\text{Cov}(N_i, N_j) = -1/n$ ($i \neq j$).

Quindi $\text{Var}(\hat{\theta})_{Boot} = \frac{1}{n^2} \sum_{i,j=1}^n \alpha(X_i) \alpha(X_j) \text{Cov}(N_i, N_j) = \frac{\sum_{i=1}^n (\alpha(X_i) - \alpha_{(.)})^2}{n^2}$ dove $\alpha_{(.)} = \hat{\theta}$, da qua capiamo che $\text{Var}(\hat{\theta})_{Boot} = \text{Var}(\hat{\theta}) * \frac{n-1}{n}$.

9 Intervalli di confidenza

Un intervallo di confidenza 0.95 è un intervallo nel quale teoricamente lo stimatore è contenuto il $0.95 * n$ volte dove n è il numero delle prove, noi quando lo costruiamo ci fidiamo sia così in particolare se la distribuzione è la conosciamo.

Sappiamo dalla teoria che se $X \approx F(\theta)$ allora per il TCL $\frac{\hat{\theta} - \theta}{SE(\hat{\theta})} \approx N(0, 1)$ se $n \gg 0$.

Andremo a creare con questa distribuzione intervalli di confidenza simmetrici che includono bene se la funzione è una normale o molto vicina ad essa.

Se però la distribuzione dello stimatore non è normale avremmo che l'intervallo nominale di 0.95 sarà diverso da quello reale che otteniamo con una prova frequentista (detto anche livello di copertura).

Più generalmente possiamo dire che più la \hat{F} si avvicina alla F reale più la l'intervallo frequentista si avvicina all'intervallo teorico.

F non è sempre nota e approssimarla a funzioni che conosciamo non è corretto quindi possiamo studiarla con il metodo bootstrap così da creare intervalli bootstrap.

9.1 Intervallo di confidenza normale

Si usa se $\hat{\theta}$ è approssimativamente normale però questa distribuzione non è centrata con il valore θ ma ha un Bias $b = \hat{\theta}^* - \hat{\theta}$, dove $\hat{\theta}^*$ è calcolato con il metodo bootstrap, quindi $\hat{\theta} \approx N(\theta + b, \sigma^2)$ da qua capiamo che il suo intervallo di confidenza $1 - 2\alpha$ è $IC = (\hat{\theta} - b \pm Z_{1-\alpha})$

9.2 Intervallo di confidenza di student

Sappiamo che $\frac{\hat{\theta} - \theta}{SE(\hat{\theta})} \approx T_{n-1}$, dalla teoria classica possiamo ricostruire gli intervalli di confidenza in maniera analoga a come se si disponessero su una normale, solamente cambiando la distribuzione.

Questo ragionamento ci permette di studentizzare la nostra serie di campioni bootstrap.

Questo metodo ha il vantaggio che tiene conto dell'assimetria dei dati. Però la non presenza di valori tabulati aumenta di molto la fatica computazionale nel calcolo. Inoltre l'IC student non è invariante rispetto alle trasformazioni e su pochi dati tende ad essere inefficiente.

Si svolge il seguente algoritmo:

- Stimiamo θ e SE con il metodo bootstrap. Se SE si può calcolare con una formula si calcola analiticamente, invece se non la abbiamo, otteniamo SE attraverso una stima boostrap, quindi in questo caso abbiamo un bootstrap annidiato questo fa aumentare di molto la fatica computazionale.
- Calcolo la distribuzione come $z^* = \frac{\hat{\theta}^{*b} - \hat{\theta}}{SE(\hat{\theta}^{*b})}$
- Con la distribuzione posso calcolare i quantili empirici $z_{(1-\alpha)}^*$ e $z_{(\alpha)}^*$
- Calcolo l'intervallo di confidenza come $IC = (\hat{\theta} - z_{(1-\alpha)}^* * SE(\hat{\theta}), \hat{\theta} - z_{(\alpha)}^* * SE(\hat{\theta}))$

9.3 Intervallo di confidenza base

Se $\hat{\theta} - \theta \approx G$ dove G non è nota, possiamo usare il bootstrap per stimarla.

Otteniamo la distribuzione di G come $G^{*b} = \hat{\theta}_{*b} - \hat{\theta}$ da qua otteniamo i quantili empirici $(G_{1-\alpha}, G_{\alpha})$. che possiamo usare nell'intervallo di confidenza, in questo modo otteniamo che $IC = (\hat{\theta} - G_{1-\alpha}^*, \hat{\theta} - G_{\alpha}^*)$

9.4 Intervallo di confidenza sui percentili

Se uso il bootstrap per stimare $\hat{\theta}$ ottengo anche la distribuzione di esso, cioè tutti i valori $\hat{\theta}^{*b}$ vanno a formare G . Posso calcolare quindi anche i quantili empirici di tale distribuzione $(G_{1-\alpha}, G_{\alpha})$, da qua creare $IC = (\hat{\theta}_{(B+1)\alpha}^*, \hat{\theta}_{(B+1)(1-\alpha)}^*)$. Inoltre sia IC percentile e IC base sono invarianti rispetto alle trasformazioni monotone (rispettano ITP).

Più generalmente sappiamo che vale il seguente **Lemma**: Se $\hat{\theta}$ non si approssima come una normale ma $\phi = h(\theta) \approx N(\phi, c^2)$, possiamo ottenere $IC(\theta) = (h^{-1}(\hat{\phi} - Z_{1-\alpha}), h^{-1}(\hat{\phi} + Z_{1-\alpha}))$. Grazie al metodo bootstrap possiamo stimare h senza fare particolari assunzioni. Se h è monotona e la conosciamo riusciamo a stimare tutto anche in maniera analitica, inoltre il livello di confidenza reale corrisponde a quello nominale.

9.5 Intervallo di confidenza BCA (Bias conected and accelerated)

È un miglioramento dell'IC percentile. Seguiamo l'algoritmo degli IC percentili trovando la distribuzione di $\hat{\theta}$, invece che prendere i quantili campionari di interesse prendiamo dei quantili corretti.

$$IC = (\hat{\theta}^*_{(B+1)\alpha}, \hat{\theta}^*_{(B+1)(1-\alpha)})$$

- $\alpha' = \phi(\hat{w} + \frac{\hat{w} + Z_{\alpha}}{1 - \hat{a} * (\hat{w} + Z_{\alpha})})$
- $\alpha'' = \phi(\hat{w} + \frac{\hat{w} + Z_{1-\alpha}}{1 - \hat{a} * (\hat{w} + Z_{1-\alpha})})$

Dove:

- Calcolo \hat{w} come le frequenza relative di $\theta^{*b} < \hat{\theta}$ quindi sarà $\hat{w} = \phi^{-1}(\frac{count(\theta^{*b} < \hat{\theta})}{B})$. Il parametro indica la distorsione dello stimatore, se la funzione fosse simmetrica allora $w = 0$.
- Calcolo $\hat{a} = \frac{\sum_{i=1}^n (\hat{\theta}_{(.)} - \hat{\theta}_{-i})^3}{6 * (\sum_{i=1}^n (\hat{\theta}_{(.)} - \hat{\theta}_{-i})^2)^{3/2}}$, notiamo che per stimare $\hat{\theta}_{(.)} - \hat{\theta}_{-i}$ ricorriamo al metodo jack-knife. Il parametro è chiamato acceleratore perchè è correlato con il tasso di cambio di $SE(\hat{\theta})$ rispetto al valore del parametro θ , misurato su una scala normalizzata. Tale parametro si usa per correggere la forte assunzione che la varianza rimane costante, che empiricamente non è vero.
- Possiamo vedere che se $\hat{a} = 0$ e $\hat{w} = 0$ allora i quantili calcolati sono uguali a quelli della normale standard quindi l'intervallo sarebbe uguale a quello calcolato con il metodo dei percentili.

Questo tipo di intervallo corregge la non normalità dei dati, la loro eventuale distorsione, e la loro varianza ipotizzando che non sia sempre fissa come è nella realtà.

Sappiamo che per il lemma ITP il quantile è invariante per trasformazione e che sono n-corretti cioè corretti al secondo ordine. Ciò significa che all'aumentare di n l'IC approssima il livello di copertura a velocità n^{-1}

10 Test di Ipotesi

Se ho un insieme di dati $Y = Y_1 \dots Y_n$ e voglio testare una certa H_0 avrò bisogno di una determinata statistica test $T(Y)$ con la sua statistica test osservata che è $t_{obs}(y)$.

Di solito nell'inferenza classica se questo valore è alto rifiutiamo H_0 infatti avremo che $P(T > t_{obs}) = P_{value}$ che è basso; come notiamo però per avere un risultato dobbiamo ipotizzare di conoscere la funzione di distribuzione T , questa è una condizione molto forte, aggirabile però con il metodo bootstrap. Infatti possiamo stimare T condizionata da H_0 con il metodo bootstrap.

Stimando però aggiungiamo due errori:

- Errore campionamento: Disponiamo solo di un campione di ampiezza n e non l'intera popolazione
- Errore ricampionamento bootstrap: Usiamo solo B campioni bootstrap anziché un numero infinito.

Algoritmo per valutare se $F=G$

Se ho $Z = (Z_1 \dots Z_n)$ creati da F e $Y = (Y_1 \dots Y_n)$ creati da G

$H_0 : F = G$ contro $H_1 : F \neq G$ e $\mu_z > \mu_y$

- Unisco z e y $x = c(Z, Y)$
- Creo x^{*b} con il metodo bootstrap
- Creo la statistica $\hat{t}^{*b} = z^{*b} - y^{*b}$ che è un vettore di statistiche, dove z^{*b} media delle prime n osservazioni e y^{*b} media delle ultime m osservazioni.
- Valuto dove $\frac{\#(\hat{t}^{*b} > \hat{t})}{B} = P_{value}$ se il P_{value} è più alto della soglia critica non rifiuto H_0

Algoritmo per valutare se $\mu_z = \mu_y$

Se ho $Z = (Z_1 \dots Z_n)$ creati da F e $Y = (Y_1 \dots Y_n)$ creati da G

$H_0 : \mu_z = \mu_y$ contro $H_1 : \mu_z > \mu_y$

- Unisco z e y $x = c(Z, Y)$ e calcolo \bar{x}
- Creo $\tilde{z}_i = z_i - \bar{z} + \bar{x}$ e $\tilde{y}_i = y_i - \bar{y} + \bar{x}$ così otteniamo dei valori centrati su \bar{x}
- Ricampiono con metodo bootstrap \tilde{z}^{*b} e \tilde{y}^{*b}
- Creo la statistica $\hat{t}^{*b} = \frac{z^{*b} - y^{*b}}{\sqrt{\frac{var(z^{*b})}{n} + \frac{var(y^{*b})}{n}}}$ che è un vettore di statistiche.
- Valuto dove $\frac{\#(\hat{t}^{*b} > \hat{t})}{B} = P_{value}$ se il P_{value} è più alto della soglia critica non rifiuto H_0