Università di Catania

Metodi Matematici e Statistici

Dr. Giovanni Nastasi
giovanni.nastasi@unict.it

Dipartimento di Matematica e Informatica
Università degli Studi di Catania, Italy

CdS in Informatica
A.A. 2023-2024

Numeri pseudo-casuali

Introduzione

I **numeri pseudo-casuali** sono numeri ottenuti mediante algoritmi deterministici che producono sequenze che approssimativamente seguono delle opportune distribuzioni di probabilità.

Un algoritmo di questo tipo è detto **generatore di numeri pseudo-casuali** (pseudo-random number generator).

I numeri pseudo casuali hanno notevoli applicazioni nella simulazione di fenomeni aleatori, nell'applicazione dei metodi Monte Carlo, in crittografia, ecc.

Per generare invece numeri realmente casuali occorre invece utilizzare generatori hardware che sfruttano tipicamente fenomeni microscopici come il rumore termico o l'effetto fotoelettrico o altri fenomeni quantistici. L'output viene poi convertito in segnale digitale, cioè una sequenza di 0 e 1.

Un generatore ideale hardware di numeri casuali con distribuzione uniforme in $[0, 1]$ si può costruire considerando un'urna contenente 10 palline numerate da 0 a 9 ed effettuando infinite estrazioni con rimpiazzo che costituiscono le cifre decimali del numero.

Introduzione

Nei calcolatori i numeri reali sono sempre rappresentati usando un numero finito N di bit (di solito 32 o 64). Quindi per generare numeri aleatori R_n possiamo generare interi I_n uniformemente distribuiti fra 0 e $M = 2^N - 1$ e definire i numeri aleatori reali R_n ponendo $R_n = I_n/M$ che sono compresi tra 0 e 1.

I generatori basati sul **metodo delle congruenze lineari** (GCL) hanno la forma

$$I_{n+1} = (aI_n + b) \bmod m,$$

dove a si chiama moltiplicatore, b incremento ed m il modulo. I_0 è il seme (seed) che inizializza il generatore. Il valore di I al passo $n+1$ si ottiene a partire dal valore di I al passo n.

Esempio. Sia $I_0 = a = b = 7$ e $m = 10$. Si ha quindi $I_0 = 7$, $I_1 = 7 \cdot 7 + 7 \bmod 10 = 56 \bmod 10 = 6$, $I_2 = 7 \cdot 6 + 7 \bmod 10 = 9$, $I_3 = 7 \cdot 9 + 7 \bmod 10 = 0$, $I_4 = 7 \cdot 0 + 7 \bmod 10 = 7 = I_0$. Da qui in poi la sequenza si ripete. Il suo periodo T vale dunque 4.

Introduzione

Il modulo m dà quindi il massimo periodo che il generatore può assumere. Il periodo dipende anche da altri parametri per cui può essere molto più piccolo di m.

Un buon generatore di numeri pseudo-casuali dovrebbe avere le seguenti proprietà:

- velocità, perché di solito è necessario generare successioni lunghe di numeri di questo tipo per le simulazioni;
- lunga periodicità, per evitare che nelle simulazioni si verifichino delle correlazioni spurie.

I moderni linguaggi di programmazione hanno delle librerie native o importabili di generatori di numeri pseudo-casuali.

Focalizziamo l'attenzione sui generatori di numeri pseudo-casuali con distribuzione uniforme in $[0, 1]$.

Introduzione

Uno dei più comuni generatori di numeri pseudo-casuali è l'algoritmo Mersenne Twister (MT).

M. Matsumoto and T. Nishimura.
Mersenne Twister: A 623-dimensionally equidistributed uniform pseudorandom number generator.
ACM Trans. on Modeling and Computer Simulation Vol. 8, No. 1, January pp.3-30 (1998).
http://www.math.sci.hiroshima-u.ac.jp/m-mat/MT/emt.html

- MT è basato sui numeri primi di Mersenne.
- MT ha un periodo molto lungo: $2^{19937} - 1$.
- MT non è crittograficamente sicuro, cioè può essere usato per le simulazioni Monte Carlo ma non per applicazioni crittografiche. Tuttavia si può rendere sicuro se usato in combinazione con altri algoritmi (Secure Hash Algorithms).
- MT è disponibile come libreria in molti linguaggi di programmazione (C, C++, FORTRAN, MATLAB, Python, ...).

Introduzione

Link alla documentazione Python:

- https://numpy.org/doc/stable/reference/random/index.html
- https://numpy.org/doc/stable/reference/random/bit_generators/index.html
- https://numpy.org/doc/stable/reference/random/bit_generators/mt19937.html

Si può testare la bontà di un generatore di numeri pseudo-casuali impostando un test del chi-quadro. Cioè si suddivide l'intervallo $[0, 1]$ in m sottointervalli di uguale ampiezza, si generano N numeri pseudo-casuali in $[0, 1]$ e si contano quanti numeri cadono in ciascun intervallo. Sia Y_i tale numero per $i = 1, 2, \ldots, m$. Si pone

$$X = \sum_{i=1}^{m} \frac{(Y_i - N/m)^2}{N/m}.$$

Per $N \to +\infty$ si ha che $X \sim \chi^2(m-1)$. L'ipotesi nulla è che le frequenze osservate siano uguali a quelle teoriche. Il rigetto dell'ipotesi nulla porta alla conclusione che il generatore sia inaccurato.

Generazione di numeri pseudo-casuali

Fino a questo punto abbiamo parlato della generazione di numeri pseudo-casuali che seguono approssimativamente una distribuzione uniforme in $[0, 1]$. Vedremo adesso come costruire algoritmi che generano numeri pseudo-casuali che seguono approssimativamente una distribuzione di probabilità assegnata. Le tecniche che vedremo sono due:

1. metodo di inversione della funzione di ripartizione;
2. metodo del rigetto.

Da questo momento in poi supporremo quindi di saper generare numeri pseudo-casuali con distribuzione uniforme in $[0, 1]$.

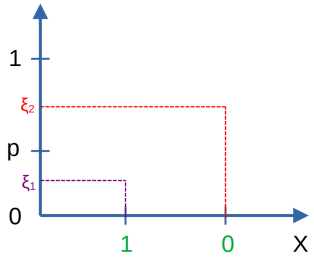
Metodo di inversione della funzione di ripartizione

Distribuzione binomiale

Consideriamo prima la distribuzione di Bernoulli, cioè $X \sim B(1,p)$, $0 < p < 1$.

Consideriamo la partizione $[0,1] = [0,p] \cup [p,1]$.

Sia $\xi \sim U([0,1])$ un numero pseudo-casuale.



Assumiamo:

$$X = \begin{cases} 1 & \text{if} \quad \xi \in [0,p[\\ 0 & \text{if} \quad \xi \in [p,1] \end{cases}$$

Per generare numeri pseudo-casuali con distribuzione $B(n,p)$, possiamo generare n numeri pseudo-casuali con distribuzione $B(1,p)$ e sommarli.

Metodo di inversione della funzione di ripartizione

Distribuzione multinomiale

Vogliamo prima generare numeri pseudo-casuali $X \sim B(1, p_1, p_2, \ldots, p_m)$.

Sia $F_k = \sum_{j=1}^{k} p_j$, con $k = 1, 2, \ldots, m$.

Allora

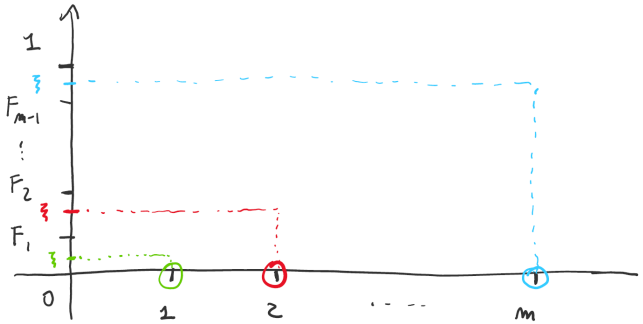
$$
\begin{aligned}
F_1 &= p_1 \\
F_2 &= p_1 + p_2 \\
&\vdots \\
F_k &= p_1 + p_2 + \ldots + p_k \\
&\vdots \\
F_m &= p_1 + p_2 + \ldots + p_m = 1
\end{aligned}
$$

Consideriamo la partizione

$$[0, 1] = [0, F_1[\cup [F_1, F_2[\cup \ldots \cup [F_{m-1}, 1].$$

Metodo di inversione della funzione di ripartizione

Sia $\xi \sim U([0,1])$ un numero pseudo-casuale.



Assumiamo:

$$i = \begin{cases} 1 & \text{if} \quad \xi \in [0, F_1[\\ 2 & \text{if} \quad \xi \in [F_1, F_2[\\ \vdots & \\ m & \text{if} \quad \xi \in [F_{m-1}, 1] \end{cases}$$

Poniamo $X = (X_1, X_2, \ldots, X_m)$ con $X_j = \begin{cases} 1 & \text{if} \quad j = i, \\ 0 & \text{if} \quad j \neq i. \end{cases}$

Per generare numeri pseudo-casuali con distribuzione $B(n, p_1, p_2, \ldots, p_m)$, possiamo generare n numeri pseudo-casuali con distribuzione $B(1, p_1, p_2, \ldots, p_m)$ e sommarli.

Esercizio

Scrivere un algoritmo per generare numeri pseudo casuali con distribuzione multinomiale

$$B(1, 1/4, 1/2, 1/4).$$

Generare 2000 di questi numeri e costruire un istogramma verificando l'accordo con la distribuzione teorica di probabilità.

Metodo di inversione della funzione di ripartizione

Nel caso continuo, dimostriamo le seguenti proprietà.

Proprietà. Sia X una v.a. di densità $f(x)$ con f continua e con supporto $(a, b) \subseteq \mathbb{R}$. Sia $F(x)$ la funzione di ripartizione di X. Se $F(x)$ è strettamente crescente allora si ha che

$$F(X) \sim U([0, 1]).$$

Dimostrazione. Siano $c_1, c_2 \in [0, 1]$ con $c_1 < c_2$. Poiché $F(x)$ è strettamente crescente allora è invertibile, pertanto si ha

$$\begin{aligned} P(c_1 \leq F(X) \leq c_2) &= P\left(F^{-1}(c_1) \leq X \leq F^{-1}(c_2)\right) \\ &= F\left(F^{-1}(c_2)\right) - F\left(F^{-1}(c_1)\right) = c_2 - c_1. \end{aligned}$$

Quindi la probabilità che $F(X)$ sia nell'intervallo $[c_1, c_2]$ è pari all'ampiezza dell'intervallo, quindi $F(X)$ ha distribuzione uniforme.

Metodo di inversione della funzione di ripartizione

Proprietà. Sia $X \sim U([0,1])$ e $F : (a,b) \to \mathbb{R}$ una funzione invertibile che rappresenti la funzione di ripartizione di una qualche distribuzione. Allora la variabile aleatoria $Y = F^{-1}(X)$ ha F come funzione di ripartizione.

Dimostrazione. Sia $t \in (a,b)$. Allora

$$F_Y(t) = P(Y \leq t) = P\left(F^{-1}(X) \leq t\right) = P(X \leq F(t)) = F_X(F(t)) = F(t),$$

ove l'ultimo passaggio segue dal fatto che $X \sim U([0,1])$.

Metodo di inversione della funzione di ripartizione

Esempio. Sia $X \sim U([0,1])$ e $Y = a + X(b-a)$, con $a, b \in \mathbb{R}$, $a < b$. Allora $Y \sim U([a,b])$.

Dimostrazione. In $[a,b]$ la f.r. di Y è

$$F_Y(t) = \frac{t-a}{b-a}.$$

Allora si ha

$$y = \frac{t-a}{b-a} \qquad \Longrightarrow \qquad y(b-a) + a = t \qquad \Longrightarrow \qquad F^{-1}(y) = a + y(b-a).$$

Pertanto

$$Y = F^{-1}(X) = a + X(b-a)$$

è uniforme in $[a,b]$.

Metodo di inversione della funzione di ripartizione

Esempio. Sia $X \sim U([0,1])$ e $Y = -\frac{\log X}{\lambda}$, con $\lambda > 0$. Allora $Y \sim \mathrm{EXP}(\lambda)$.

Dimostrazione. Se $t > 0$ allora la f.r. di una legge esponenziale di parametro $\lambda > 0$ è $F(t) = 1 - e^{-\lambda t}$. Allora si ha

$$y = 1 - e^{-\lambda t} \qquad \Longrightarrow \qquad \log(1-y) = -\lambda t \qquad \Longrightarrow \qquad F^{-1}(y) = -\frac{\log(1-y}{\lambda}.$$

Pertanto

$$Y = -\frac{\log(1-X)}{\lambda}$$

ha distribuzione esponenziale di parametro λ. Infine, osservando che se $X \sim U([0,1])$ allora anche $1 - X \sim U([0,1])$ segue che

$$Y = -\frac{\log X}{\lambda}$$

ha distribuzione esponenziale di parametro λ.

Generazione di numeri random con distribuzione normale

Siano X e Y due v.a. indipendenti ciascuna di legge $N(0,1)$. Allora $Z = X^2 + Y^2 \sim \chi^2(2)$. Ma una legge $\chi^2(2)$ è una legge esponenziale di parametro $\lambda = 1/2$.

Inoltre, generando $\theta \sim U([0, 2\pi])$ si ha che

$$X = \sqrt{Z}\cos\theta, \qquad Y = \sqrt{Z}\sin\theta.$$

Si ottiene quindi il seguente algoritmo.

1. Si generano ξ_1 e ξ_2 con distribuzione $U([0,1])$.
2. Si pone $\eta_1 = \sqrt{-2\log(\xi_1)}\cos(2\pi\xi_2)$ e $\eta_2 = \sqrt{-2\log(\xi_1)}\sin(2\pi\xi_2)$ che risultano entrambi di legge $N(0,1)$.

Per generare numeri random di legge $N(\mu, \sigma^2)$ basta generare $\xi \sim N(0,1)$ e porre $\eta = \mu + \sigma\xi$.

Generazione di numeri random di legge χ^2 e di Poisson

Numeri random con legge $\chi^2(n)$.

Per generare numeri random con legge $\chi^2(n)$ basta generare n numeri random indipendenti con legge $\chi^2(1)$ e sommarli. Per fare ciò, si genera $\xi \sim N(0,1)$ e si pone $\eta = \xi^2$.

Numeri random con legge di Poisson.

Richiede alcune proprietà delle catene di Markov. Sarà affrontato in seguito.

Esercizio

Si scrivano delle funzioni di Python per generare numeri random con distribuzione

$$U([-1,1]), \qquad \mathrm{EXP}(3), \qquad N(1,2), \qquad \chi^2(5).$$

Si generino $N = 1000$ di questi numeri e si confrontino i risultati con le distribuzioni teoriche mediante istogramma e mediante grafico quantile-quantile.

Metodo del rigetto o di reiezione

Una procedura piuttosto generale per generare numeri random di cui si conosce la densità è fornita dal metodo del rigetto, il quale è applicabile anche ai casi multidimensionali. Consideriamo il caso di una v.a. scalare di densità f(x) continua con supporto in $[a, b] \subseteq \mathbb{R}$.

Introduciamo il seguente algoritmo.

1. Si genera la coppia (ξ, η) con $\xi \sim U([a, b])$ e $\eta \sim U([0, M])$ con $M = \max_{[a,b]} f(x)$.
2. Se $0 \leq \eta \leq f(\xi)$ si accetta ξ e si pone $X = \xi$, altrimenti si rigetta e si torna al punto 1.

Proviamo che X ha densità $f(x)$. Per $a \leq t \leq b$ si ha

$$P(X \in [a, t]) = P(\xi \in [a, t] \,|\, \xi \text{ accettato}) = \frac{P(\xi \in [a, t],\ \xi \text{ accettato})}{P(\xi \text{ accettato})}$$

Metodo del rigetto o di reiezione

Tenendo conto che ξ è di legge uniforme in $[a, b]$ e che η è di legge uniforme in $[0, M]$, valutiamo il numeratore

$$
\begin{aligned}
&P(\xi \in [a,t],\, \xi \text{ accettato}) = P(\xi \text{ accettato} \,|\, \xi \in [a,t]) P(\xi \in [a,t]) \\
&= \int_a^t P(\xi \text{ accettato} \,|\, \xi = y) P(\xi = y)\, dy = \int_a^t P(\xi \text{ accettato} \,|\, \xi = y) \frac{1}{b-a}\, dy \\
&= \int_a^t P(\eta \le f(\xi) \,|\, \xi = y) \frac{1}{b-a}\, dy = \int_a^t \frac{f(y)}{M} \frac{1}{b-a}\, dy = \frac{1}{M(b-a)} \int_a^t f(y)\, dy.
\end{aligned}
$$

Invece il denominatore è

$$
P(\xi \text{ accettato}) = P(\xi \in [a,b],\, \xi \text{ accettato}) = \frac{1}{M(b-a)} \int_a^b f(y)\, dy = \frac{1}{M(b-a)}
$$

in cui si è usata la proprietà precedente con $t = b$ e ricordando che f è una densità.

Metodo del rigetto o di reiezione

Per quanto appena visto, si ha

$$P(X \in [a,t]) = \int_a^t f(y)\,dy,$$

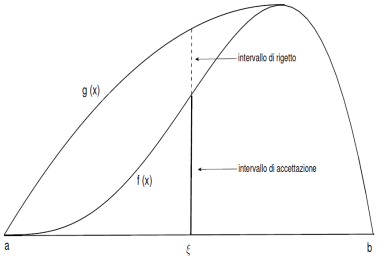
che è quello che si voleva provare.

Possiamo, quindi, generare numeri random di densià $f(x)$, generando punti (X,Y) con distribuzione uniforme nel sottografico di $f(x)$ e prendendo la prima componente X.

Osservazione. Se $f(x)$ è molto piccata allora il sottoinsieme di $[a,b]$ ove $f(x) \ll 1$ ha misura grande, ovvero l'area del rettangolo $[a,b] \times [0,M]$ è molto grande rispetto a quella del sottografico, e ciò comporta un numero alto di rigetti. Si può migliorare l'efficienza dell'algoritmo introducendo il *metodo del rigetto ottimizzato* o *campionamento ad importanza*.

Metodo del rigetto ottimizzato

Supponiamo che esista una funzione $g(x)$ definita in $[a, b]$ regolare tale che $f(x) \leq g(x)$ per ogni $x \in [a, b]$.



Poniamo $g^*(x) = \frac{g(x)}{M^*}$ ove $M^* = \int_a^b g(t)\, dt$.

Se $g^*(x)$ ha una f.r. $G(x)$ invertibile allora risulta agevole generare numeri random di densità g^*.

Metodo del rigetto ottimizzato

Si ottiene il seguente algoritmo.

1. Si genera $\xi = G^{-1}(\xi_1)$ con $\xi_1 \sim U([0,1])$ e si genera $\eta \sim U([0,1])$.
2. Se $\eta g(\xi) \leq f(\xi)$ si accetta ξ e si pone $X = \xi$, altrimenti si rigetta e si torna al punto 1.

Anche in questo caso X ha densità $f(x)$. Infatti

$$P(\xi \in [a,t],\, \xi \text{ accettato}) = \int_a^t P(\xi \text{ accettato} \,|\, \xi = y) \frac{g(y)}{M^*}\, dy$$

$$= \int_a^t P(\eta\xi \leq f(\xi) \,|\, \xi = y) \frac{g(y)}{M^*}\, dy = \int_a^t \frac{f(y)}{g(y)} \frac{g(y)}{M^*}\, dy = \frac{1}{M^*} \int_a^t f(y)\, dy$$

e quindi procedendo in modo analogo al caso del rigetto semplice si ha la tesi.

Osservazione. L'efficienza del rigetto ottimizzato dipende dalla scelta della funzione g. La scelta ottimale è quella che minimizza M^*.

Esercizio

Generare numeri pseudo-casuali con distribuzione

$$f(x) = \frac{1 + \cos x}{2\pi}, \qquad x \in [-\pi, \pi].$$

Eseguire un confronto statistico tra i numeri generati e la distribuzione teorica.

Simulazione Monte Carlo

Il termine **metodo Monte Carlo** si riferisce a un insieme di tecniche che fanno uso di variabili aleatorie, generate artificialmente da un computer, per risolvere problemi matematici.

Il nome Monte Carlo apparve per la prima volta nell'articolo:

N. Metropolis, S. Ulam.
The Monte Carlo Method.
Journal of the American Statistical Association. Vol. 247 (1949).

Applicazioni:

- Tecniche di integrazione
- Generazione numerica di variabili aleatorie
- Simulazione di processi stocastici

Applicazioni del metodo MC in fisica

- Negli anni '40 il metodo MC è stato applicato (J. von Neumann, E. Fermi, et al.) per risolvere il problema della diffusione dei neutroni in materiali fissili (si veda N. Metropolis, The beginning of the Monte Carlo method, Los Alamos Science, 1987).

- Con la locuzione Direct Simulation Monte Carlo (DSMC) ci si riferisce a una tecnica MC per modellizzare il comportamento di un gas rarefatto. È stato introdotto per la prima volta da G.A. Bird nel 1970 (si veda G.A. Bird, The DSMC Method, CreateSpace Ind. Pub., 2013).

- Trent'anni dopo è stato dimostrato che la DSMC converge alla soluzione dell'equazione di Boltzmann (si veda W. Wagner, A Convergence Proof for Bird's Direct Simulation Monte Carlo Method for the Boltzmann Equation, Journal of Statistical Physics, 66, 1992).

- Negli anno '80 la DSMC è stata adottata per simulare la dinamica delle cariche all'interno di un reticolo cristallino. Tale approccio è descritto, ad esempio, in C. Jacoboni, Theory of Electron Transport in Semiconductors, Springer-Verlag Berlin Heidelberg, 2010.

Base teorica del metodo MC

Da un punto di vista teorico il metodo MC si basa su i seguenti teoremi.

- **Legge dei grandi numeri.**
  Sia $(X_n)_{n\in\mathbb{N}}$ una successione di variabili aleatorie indipendenti definite su uno stesso spazio di probabilità con media μ e varianza $\sigma^2 > 0$ entrambe finite. Allora, si ha

$$\lim_{n\to\infty} P(|\bar{X}_n - \mu| > \eta) = 0, \qquad \forall \eta > 0,$$

  dove $\bar{X}_n = \frac{1}{n}\sum_{i=1}^{n} X_i$ è la media campionaria.

- **Teorema del limite centrale.**
  Sia $(X_n)_{n\in\mathbb{N}}$ una successione di variabili aleatorie indipendenti aventi la stessa distribuzione, con media μ e varianza $\sigma^2 > 0$ entrambe finite. Allora, si ha

$$S_n^* = \frac{\bar{X}_n - \mu}{\sigma}\sqrt{n} \xrightarrow{\mathcal{L}} N(0,1),$$

  dove $\mathcal{L}$ indica la convergenza in legge e $N(0,1)$ è la distribuzione normale standard.

Osservazioni.

- **Legge dei grandi numeri.**
  Assicura che per $n \gg 0$ le medie empiriche delle quantità di interesse convergono in probabilità alle medie teoriche.
- **Teorema del limite centrale.**
  Fornisce una stima dell'ordine di convergenza del metodo MC.
  Infatti, per la regola del 3σ, si ha

$$1 \approx P(|S_n^*| \leq 3) = P\left(\left|\frac{\bar{X}_n - \mu}{\sigma}\right| \sqrt{n} \leq 3\right)$$

  quindi $|\bar{X}_n - \mu| \leq \frac{3\sigma}{\sqrt{n}}$, cioè $\bar{X}_n$ tende a μ come $\frac{1}{\sqrt{n}}$.

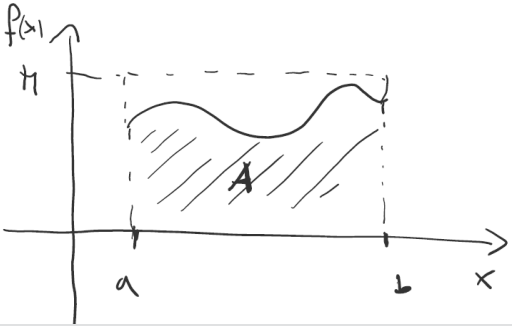
Metodo hit or miss

Sia $f : [a, b] \to \mathbb{R}$ una funzione continua tale che $f(x) \geq 0$ per ogni $x \in [a, b]$.

Si vuole calcolare

$$I = \int_a^b f(x)\, dx.$$

Consideriamo il rettangolo $R = [a, b] \times [0, M]$, essendo $M = \max_{[a,b]} f(x)$ che esiste per il Teorema di Weierstrass.



Metodo hit or miss

La probabilità che, scegliendo un punto a caso in R, esso cada nella regione A è

$$p = \frac{I}{M(b-a)}.$$

Per stimare p, generiamo N coppie (ξ_k, η_k) con $\xi_k \sim U([a,b])$ e $\eta_k \sim U([0,M])$.
Sia N_S il numero di punti che cadono in A.
Quindi $N_S \sim B(N,p)$.
Allora, per la legge dei grandi numeri, si ha

$$\frac{N_S}{N} \to p, \qquad \text{per} \quad N \to +\infty.$$

Pertanto

$$I_S = \frac{N_S}{N} M(b-a).$$

Metodo hit or miss

Osserviamo che

$$N_S \sim B\left(N, \frac{I}{M(b-a)}\right).$$

Pertanto

$$\begin{aligned}\mathrm{VAR}(I_S) &= \left(\frac{M(b-a)}{N}\right)^2 \mathrm{VAR}(N_S) = \left(\frac{M(b-a)}{N}\right)^2 N \frac{I}{M(b-a)}\left(1 - \frac{I}{M(b-a)}\right) \\ &= \frac{1}{N} I(M(b-a) - I).\end{aligned}$$

Esercizio. Sia $X \sim N(0,1)$. Si calcoli numericamente con il metodo hit or miss la probabilità $p = P(0.5 \leq x \leq 2)$.

Metodo hit or miss

Osservazione. Il metodo hit or miss si può utilizzare per calcolare l'area di un dominio di $\mathbb{R}^n$. Es: calcolo dell'area di un'ellisse nel piano. Il valore esatto è πab, essendo a e b le lunghezze dei due semiassi.

Osservazione. Il metodo hit or miss si può utilizzare per calcolare integrali doppi, tripli, ecc. In tal caso l'efficienza del metodo diventa comparabile o addirittura migliore di quella delle formule di quadratura. Es: calcolare il seguente integrale

$$\iiint_D [-(x^2 + y^2 + z^2) + 8]\, dx\, dy\, dz,$$

ove $D = [0, 2] \times [0, 2] \times [0, 2]$.

Riferimenti bibliografici

1. V. Romano, Metodi matematici per i corsi di ingegneria, Città Studi, 2018.
2. P. Baldi, Calcolo delle probabilità e statistica, Mc Graw-Hill, Milano, 1992.
3. R. Scozzafava, Incertezza e probabilità, Zanichelli, 2001.
4. D. C. Montgomery, G. C. Runger, Applied statistics and probability for engineers, 7th Edition, J. Wiley, 2018.