

1 Uniform (Pseudo) Random Number Generation

Campionamento da $\mathcal{U}([0, 1])$: numeri random indipendenti e uniformemente distribuiti su $[0, 1]$. Come verifichiamo che la sequenza di numeri generati sia indistinguibile da una sequenza di numeri random?

L'obiettivo è di generare numeri random, ovvero produrre un flusso infinito di variabili aleatorie che sono indipendenti tra loro e identicamente distribuite secondo una qualche distribuzione di probabilità. In particolare, ci concentriamo sulla distribuzione uniforme. Se riusciamo a generare numeri random da una distribuzione uniforme allora possiamo generare numeri random da una qualsiasi altra distribuzione. Nella pratica però è impossibile creare un flusso infinito di variabili iid. Il meglio a cui possiamo ambire è di produrre una sequenza di numeri con proprietà statistiche indistinguibili da quelle di una vera sequenza di numeri random. Useremo dei test statistici per capire quanto sono random i numeri generati.

Gli algoritmi che generano numeri casuali sono detti *(Pseudo) Random Number Generators (RNG)* e generano una sequenza puramente deterministica di numeri che è indistinguibile (secondo un certo numero di test statistici) da una sequenza casuale.

Pseudo Random Number Generator.

Un algoritmo pseudo RNG produce una sequenza di numeri i cui elementi possono essere considerati indipendenti e identicamente distribuiti (iid). L'algoritmo è caratterizzato da una tupla $(S, f, \mu, \mathcal{U}, g)$ dove S definisce un insieme finito di stati, $f : S \rightarrow S$ è una funzione definita su S , μ è una distribuzione di probabilità su S , \mathcal{U} è lo spazio dell'output (e.g. per una uniforme $\mathcal{U} = [0, 1]$) e $g : S \rightarrow \mathcal{U}$ è una funzione definita su S .

Tutti gli algoritmi pseudo RNG hanno la seguente struttura:

1. Genera lo stato iniziale X_0 dalla distribuzione μ su S
2. **for** $k = 1, 2, \dots$
- $X_k = f(X_{k-1})$
- $U_k = g(X_k)$
- end for**

Lo stato iniziale X_0 è detto *seed*. Un RNG che parte da un dato seed genera sempre la stessa sequenza di numeri casuali U_1, U_2, \dots . Siccome lo spazio degli stati S è finito, tutti gli RNG sono periodici. Definiamo periodo il minor numero di step necessari per finire in uno stato già visitato precedentemente. L'ideale è avere un *periodo grande*: se sono necessarie M variabili aleatorie allora il periodo L deve essere $L \gg M$, altrimenti si perde l'indipendenza tra i campioni.

Ogni RNG ha necessità di essere veloce, efficiente e riproducibile. Inoltre, dovrebbe avere la possibilità di generare più sequenze in parallelo e dovrebbe evitare di produrre 0 e 1 (per evitare di dividere per 0 o altre complicazioni numeriche).

Common Pseudo RNGs

Linear Congruential Generator (LCG)

- Definiamo il modulo $m > 0$.
- Definiamo il moltiplicatore $a \in (0, m)$ e l'incremento $b \in (0, m)$.
- Definiamo lo spazio degli stati come $S = \{0, 1, \dots, m - 1\}$.
- Il passaggio $X_{k-1} \mapsto X_k$ è dato da $X_k = (aX_{k-1} + b) \bmod m$.
- L'output finale è dato da $U_k = X_k/m$.

I valori m, a, b sono costanti che definiscono il generatore. La sequenza $\{X_k\}_k$ è tale che X_k è un valore in $\{0, 1, 2, \dots, m - 1\}$. Siccome X_k può prendere solo m valori diversi, l'output di un LCG comincia a ripetersi dopo (al massimo) m steps. Per questo motivo è necessario che m sia il più alto possibile. Una scelta comune è $m = 2 \cdot 10^9$, ma non è più sufficiente per le applicazioni di oggi.

Matrix Congruential Generator (MCG)

Gli algoritmi LCG con $b = 0$ vengono detti *Multiplicative Congruential Generator*. È possibile generalizzare questi ultimi in q -dimensioni, ottenendo quindi i *Matrix Congruential Generator (MCG)* di ordine q .

- Definiamo il modulo $m > 0$ e una matrice $A \in \mathbb{N}^{q \times q}$ invertibile.
- Definiamo lo spazio degli stati come $S = \{0, 1, \dots, m - 1\}^q$.
- Il passaggio $X_{k-1} \mapsto X_k$ è dato da $X_k = A X_{k-1} \bmod m$.
- L'output finale è dato da $U_k = X_k/m \in (0, 1)^q$.

Per aumentare l'efficienza conviene che la matrice A sia sparsa, ovvero che abbia tanti zeri. Un caso particolare è quello in cui la matrice A è tale che $X_k = (a_1 X_{k-1} + a_2 X_{k-2} + \dots + a_q X_{k-q}) \bmod m$. L'output è nuovamente $U_k = X_k/m$. In questo caso il periodo è dato da $m^q - 1$.

Modulo-2 Generator

I *Modulo-2 Generator* sono MCG con $m = 2$. In particolare, sono definiti dalla matrice A tale che $X_k = (a_1 X_{k-1} + a_2 X_{k-2} + \dots + a_q X_{k-q}) \bmod 2$. Il periodo è dato da $2^q - 1$ per cui, per ottenere un periodo grande, è sufficiente scegliere q grande.

Un esempio di questa classe di generatori è il *Mersenne Twister*: ha un periodo di $2^{19937} - 1$, è veloce e supera tutti i test statistici.

Combined Generator

I *Combined Generator* combinano l'output di diversi generatori deboli (quindi di bassa qualità in termini di efficienza, velocità, periodo, etc.) per creare un generatore di alta qualità.

Un esempio è il generatore *Wichmann-Hill*, che combina tre LCG con diversi valori per m e a , mentre $b = 0$ per tutti. In particolare: si generano $X_k = a_1 X_{k-1} \bmod m_1$, $Y_k = a_2 Y_{k-1} \bmod m_2$, $Z_k = a_3 Z_{k-1} \bmod m_3$ e si definisce l'output $U_k = \text{decimal}(X_k/m_1 + Y_k/m_2 + Z_k/m_3)$. Questo generatore ha un periodo più grande di ciascuno dei LCG e performance abbastanza bene in termini di test statistici.

Goodness-of-fit

La sequenza prodotta è buona? Assomiglia, in termini di proprietà, a una sequenza di numeri random? Possiamo verificare la qualità di un RNG sia teoricamente (studiando le proprietà del generatore senza considerare gli output prodotti) sia empiricamente (eseguendo diversi test statistici sull'output). Ci concentriamo sull'analisi empirica e sfruttiamo i cosiddetti test di goodness-of-fit, ovvero test statistici non parametrici.

Supponiamo di avere un campione $\mathbf{X} = (X_1, \dots, X_n)$ e una distribuzione caratterizzata dalla funzione di ripartizione $F(\cdot)$. Vogliamo testare l'ipotesi H_0 che \mathbf{X} è stato campionato in modo indipendente da $F(\cdot)$. Definiamo la funzione di ripartizione empirica $\hat{F}_n(x) = \frac{\text{numero di elementi in } \mathbf{X} \text{ minori di } x}{n}$.

• QQ-plot (quantile)

Il QQ-plot è un test che confronta i quantili di $\hat{F}_n(\cdot)$ e di $F(\cdot)$, dove i primi sono definiti come $\hat{q}_{j/n} = \arg \min_x \{\hat{F}_n(x) \geq j/n\} = X^{(j)}$ e gli ultimi sono definiti come $q_t = \arg \min_x \{F(x) > t\}$. Se \mathbf{X} proviene da $F(\cdot)$ allora i punti $(q_{j/n}, \hat{q}_{j/n})$, $j = 1, \dots, n$, saranno allineati sulla diagonale.

• Kolmogorov-Smirnov test ($F(\cdot), \hat{F}_n(\cdot)$)

Il test di Kolmogorov-Smirnov confronta direttamente la distribuzione empirica $\hat{F}_n(\cdot)$ e la distribuzione reale $F(\cdot)$. La statistica di Kolmogorov-Smirnov è data da $D_n = \sup_{x \in \mathbb{R}} |\hat{F}_n(x) - F(x)|$ e viene confrontata con la variabile aleatoria di Kolmogorov K , che segue la distribuzione di Kolmogorov. Si ha che $\sqrt{n} D_n \rightarrow K$ (converge in distribuzione). Il test si costruisce usando i valori critici della distribuzione di Kolmogorov: rifiutiamo H_0 a livello α se $\sqrt{n} D_n > K_\alpha$, dove K_α è tale che $\mathbb{P}(K \leq K_\alpha) = 1 - \alpha$.

• χ^2 test (istogrammi)

L'idea del test χ^2 è di confrontare gli histogrammi invece delle funzioni di ripartizione: verifichiamo che l'istogramma del campione è simile all'istogramma della vera distribuzione.

$$\begin{aligned} \tilde{\mathbf{X}}_k &= [X_k^1, \dots, X_k^q]^T \\ &= \begin{bmatrix} 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \\ a_q & \cdots & a_1 \end{bmatrix} \end{aligned}$$

2 Random Variable Generation

Campionamento da distribuzioni generiche usando come base i numeri random iid $\mathcal{U}([0, 1])$.

Inverse-Transform Method (for invertible $f(\cdot)$)

Molte distribuzioni sono collegate all'uniforme tramite semplici trasformazioni. È possibile sfruttare queste trasformazioni e usare l'*Inverse-Transform Method* per ottenere campioni da una distribuzione unidimensionale invertibile. In particolare, si genera una variabile aleatoria uniforme U e successivamente si ottiene una variabile aleatoria X da una generica distribuzione attraverso la funzione di ripartizione. Questo è possibile sia nel caso discreto che continuo.

Discrete Inverse-Transform.

Consideriamo una variabile aleatoria discreta X che assume i valori $x_1 < x_2 < \dots < x_n$ con probabilità $\mathbb{P}(X = x_i) = p_i$, $\sum_{i=1}^n p_i = 1$. Definiamo inoltre $F_i = \mathbb{P}(X \leq x_i) = \sum_{j=1}^i p_j$.

In queste condizioni è possibile generare X come segue:

1. Genera $U \sim \mathcal{U}([0, 1])$
 2. Fissa $X = x_i$ se $F_{i-1} < U \leq F_i$
-

Infatti $\mathbb{P}(X = x_i) = \mathbb{P}(F_{i-1} < U \leq F_i) = \mathbb{P}(U \in (F_{i-1}, F_i]) = p_i$.

■ Sfruttando questo algoritmo possiamo campionare da una *Bernoulli*. Consideriamo $X \sim Be(p)$, dove $\mathbb{P}(X = 1) = p$ e $\mathbb{P}(X = 0) = 1 - p$. Partendo da $U \sim \mathcal{U}([0, 1])$ possiamo settare $X = 0$ se $U \leq 1 - p$ e $X = 1$ se $U > 1 - p$. In questo modo si ottiene $\mathbb{P}(X = 0) = \mathbb{P}(U \leq 1 - p) = 1 - p$ e $\mathbb{P}(X = 1) = \mathbb{P}(U > 1 - p) = 1 - (1 - p) = p$.

■ È possibile campionare da una *Binomiale*, sfruttando il suo legame con la Bernoulli. Una variabile aleatoria $Y \sim Bin(n, p)$ può essere scritta come la somma di n Bernoulli indipendenti, cioè $Y = X_1 + \dots + X_n$ con $X_1, \dots, X_n \sim Be(p)$. Per ottenere un campione $Y \sim Bin(n, p)$ è quindi sufficiente generare $X_1, \dots, X_n \sim Be(p)$ e fissare $Y = \sum_{i=1}^n X_i$.

Continuous Inverse-Transform.

Consideriamo una variabile aleatoria continua X che assume valori nell'intervallo $[a, b]$ e che possiede una funzione di ripartizione $F(\cdot)$ strettamente crescente tale che $F(x) = \mathbb{P}(X \leq x) : [a, b] \rightarrow [0, 1]$, $F(a) = 0$, $F(b) = 1$ e $\exists! F^{-1} : [0, 1] \rightarrow [a, b]$.

In queste condizioni è possibile generare X come segue:

1. Genera $U \sim \mathcal{U}([0, 1])$
 2. Fissa $X = F^{-1}(U)$
-

Infatti $\mathbb{P}(X \leq x) = \mathbb{P}(F^{-1}(U) \leq x) = \mathbb{P}(U \leq F(x)) = F(x)$.

■ Con questo algoritmo è possibile campionare da una *Esponenziale*. Consideriamo $X \sim Exp(\lambda)$ con $\lambda > 0$. La funzione di ripartizione di X è data da $F(x) = 1 - e^{-\lambda x}$ e la sua inversa si può ricavare partendo da $y = F(x) = 1 - e^{-\lambda x}$ e ottenendo $x = -\frac{1}{\lambda} \log(1 - y) = F^{-1}(y)$. Per ottenere quindi $X \sim Exp(\lambda)$ è sufficiente partire da $U \sim \mathcal{U}([0, 1])$ e fissare $X = F^{-1}(U) = -\frac{1}{\lambda} \log(1 - U)$. Notiamo che se $U \sim \mathcal{U}([0, 1])$ anche $1 - U \sim \mathcal{U}([0, 1])$, per cui si può direttamente fissare $X = -\frac{1}{\lambda} \log(U)$.

Quando $F(\cdot)$ non è continua o non è strettamente monotona, è possibile introdurre l'*inversa generalizzata* di $F(\cdot)$: $F^-(u) = \inf\{x : F(x) \geq u\}$. Dopodiché, per generare $X \sim F$ è sufficiente generare $U \sim \mathcal{U}([0, 1])$ e applicare la trasformazione $X = F^-(U)$.

Composition Method (for composed $f(\cdot)$)

Supponiamo di voler campionare una variabile aleatoria X che viene da una combinazione di funzioni di ripartizione, quindi $F(x) = \sum_{i=1}^n p_i F_i(x)$, con $\{p_i\}_{i=1}^n$ distribuzione di probabilità su $\{F_1, \dots, F_n\}$. L'idea è di decidere da quale $F_i(\cdot)$ campionare in base alle probabilità p_i e quindi campionare con uno dei metodi precedentemente introdotti.

Composition Method.

Consideriamo una variabile aleatoria $X \sim F(\cdot)$ con $F(\cdot)$ combinazione di funzioni di ripartizione, cioè $F(x) = \sum_{i=1}^n p_i F_i(x)$, con $\{p_i\}_{i=1}^n$ distribuzione di probabilità su $\{F_1, \dots, F_n\}$.

Si può generare X come segue:

1. Genera una v.a. discreta Y tale che $\mathbb{P}(Y = i) = p_i$
2. Genera $X \sim F_Y$ con uno dei metodi già introdotti

Ad esempio, si può generare Y e X con l'Inverse-transform method.

■ È possibile campionare $X \sim Laplace(0, 1)$, la cui funzione di densità è data da $f(x) = \frac{1}{2} e^{-|x|} = \frac{1}{2} e^{-x} \mathbf{1}_{X \geq 0} + \frac{1}{2} e^x \mathbf{1}_{X < 0}$. Possiamo vedere $f(\cdot)$ come la somma pesata di due funzioni di densità: $Exp(1)$ e $-Exp(1)$. Siccome i due pesi sono uguali, si può usare una $Be(\frac{1}{2})$. Per campionare X è dunque sufficiente generare $B \sim Be(\frac{1}{2})$ e $Y \sim Exp(1)$. Dopodiché si fissa $X = Y$ se $B = 1$ e $X = -Y$ se $B = 0$.

■ Supponiamo di voler campionare X proveniente da una combinazione di gaussiane. In particolare, consideriamo tre gaussiane così che la distribuzione di X è data da $p_1 N(\mu_1, \sigma_1^2) + p_2 N(\mu_2, \sigma_2^2) + p_3 N(\mu_3, \sigma_3^2)$. Con probabilità p_i useremo la gaussiana $N(\mu_i, \sigma_i^2)$ per il campionamento. Generiamo una variabile aleatoria discreta Y tale che $\mathbb{P}(Y = i) = p_i$. Infine, generiamo $X \sim N(\mu_Y, \sigma_Y^2)$.

Acceptance-Rejection Method (for un-samplable $f(\cdot)$)

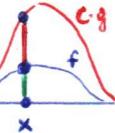
Consideriamo una variabile aleatoria $X \sim f(\cdot)$ e supponiamo di non essere in grado di campionare direttamente da $f(\cdot)$ (potrebbe essere difficile da invertire oppure potremmo non sapere la costante di normalizzazione di $f(\cdot)$). È possibile campionare da $f(\cdot)$ sfruttando una distribuzione ausiliaria $g(\cdot)$ tale che $C g(x) \geq f(x)$ per qualche $C \in \mathbb{R}$ e per tutte le possibili x . Generiamo un campione $Y \sim g(\cdot)$ e, indipendentemente da Y , generiamo un campione $U \sim \mathcal{U}([0, 1])$. Accettiamo il campione Y come campione di $f(\cdot)$ solo se $U \leq f(Y)/C g(Y)$. La sequenza dei campioni accettati sarà distribuita secondo $f(\cdot)$.

Acceptance-Rejection Method.

Consideriamo una variabile aleatoria $X \sim f(\cdot)$ e consideriamo $g(\cdot)$ funzione di densità tale che $C g(x) \geq f(x)$ per qualche $C \in \mathbb{R}$ e per tutte le possibili x . Si può generare X come segue:

1. Genera $Y \sim g(\cdot)$
2. Genera $U \sim \mathcal{U}([0, 1])$, indipendentemente da Y
3. Se $U \leq f(Y)/C g(Y)$ allora $X = Y$, altrimenti torna a 1.

In pratica, generiamo $Y \sim g(\cdot)$ e accettiamo con probabilità $f(Y)/C g(Y)$. Intuitivamente, l'idea è di generare un primo punto Y sull'asse delle x secondo la distribuzione $g(\cdot)$, spostarsi su quel punto e generare un secondo punto sul segmento $(0, C g(Y))$ secondo la distribuzione $\mathcal{U}([0, C g(Y)])$. Se il secondo punto si troverà sotto (o su) $f(Y)$ allora Y verrà accettato come campione di $f(\cdot)$.



Il metodo funziona anche nel caso in cui non conosciamo la costante di normalizzazione di $f(\cdot)$. Se $f(x) = K f(x)$ e conosciamo solo $f(\cdot)$ possiamo ugualmente applicare AR. In particolare, la probabilità con cui accettiamo i campioni (misura dell'efficienza del metodo) è data da:

$$\mathbb{P}\left(U \leq \frac{\tilde{f}(Y)}{C g(Y)}\right) = \frac{1}{C} \int_{\mathbb{R}} \tilde{f}(y) dy = \frac{1}{K C} \int_{\mathbb{R}} f(y) dy = \frac{1}{K C}$$

L'interpretazione geometrica della probabilità con cui accettiamo è il rapporto tra l'area sotto a $\tilde{f}(x)$ e l'area sotto a $C g(x)$. Pertanto, il metodo è tanto più efficiente quanto più $\tilde{f}(\cdot)$ e $C g(\cdot)$ sono vicine.

Transformation of Random Variable

Location-Scale Family

Un altro metodo molto efficace per campionare variabili aleatorie è quello di trasformare le funzioni di densità o, più precisamente, trasformare le variabili aleatorie. Si tratta di una serie di trasformazioni *ad-hoc*.

L'idea è di ottenere un campione da $f(\cdot)$ partendo da $\tilde{f}(\cdot)$ appartenente alla stessa *location-scale family*. Una famiglia di distribuzioni continue con funzioni di densità $\{f(x; \mu, \sigma), \mu \in \mathbb{R}, \sigma > 0\}$ della forma:

$$f(x; \mu, \sigma) = \frac{1}{\sigma} \tilde{f}\left(\frac{x-\mu}{\sigma}\right)$$

è detta *location-scale family*, con funzione di densità standard $\tilde{f}(\cdot)$.

Il parametro μ è chiamato *location* e il parametro σ è chiamato *scale*.

Per campionare una variabile aleatoria X da una funzione di densità $f(\cdot)$ appartenente a una location-scale family è sufficiente campionare dalla funzione di densità standard e adattare il campione alla funzione di densità particolare.

1. Genera $Z \sim \tilde{f}(\cdot) \equiv f(\cdot; 0, 1)$
2. Fissa $X = \mu + \sigma Z \sim f(\cdot; \mu, \sigma)$

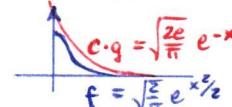
■ Questo metodo può essere usato per campionare da una generica gaussiana $N(\mu, \sigma^2)$: per ottenere $X \sim N(\mu, \sigma^2)$ è sufficiente campionare $Z \sim N(0, 1)$ e fissare $X = \mu + \sigma Z$. Come campioniamo da $N(0, 1)$?

È possibile applicare questo metodo anche con trasformazioni più generiche. Supponiamo che X sia una variabile aleatoria con funzione di densità $f_X(\cdot)$ e $Y = g(X)$, dove $g(\cdot)$ è una funzione monotona. Allora, definendo $\mathcal{Y} = \{y : y = g(x) \text{ per qualche } x\}$, otieniamo che la funzione di densità di Y è data da $f_Y(y) = f_X(g^{-1}(y)) |(g^{-1})'(y)| \mathbf{1}_{y \in \mathcal{Y}}$.

Standard Gaussian Generator - $N(0, 1)$

Acceptance-Rejection Method with $Exp(1)$

È possibile campionare da $N(0, 1)$ usando l'acceptance-rejection method e scegliendo come funzione ausiliare $Exp(1)$. In particolare, consideriamo solo la gaussiana standard nel primo quadrante, la cosiddetta *gaussiana positiva*. La sua funzione di densità è data da $f(x) = \sqrt{2/\pi} e^{-x^2/2}$ e scegliendo $g(x) = e^{-x}$ e $C = \sqrt{2} e/\pi$ otteniamo che $f(x) \leq C g(x)$ per $x \geq 0$. Ne segue che l'efficienza



del metodo è $1/C \approx 0.76$. Siccome il metodo ci limita al primo quadrante, per ottenere un campione $Z \sim N(0, 1)$ dobbiamo campionare $X \sim f(\cdot)$ e, indipendentemente, $B \sim Be(\frac{1}{2})$. Il campione finale sarà dato da $Z = X$ se $B = 1$, $Z = -X$ se $B = 0$.

- Per campionare $Z \sim N(0, 1)$ è possibile usare l'acceptance-rejection method partendo da:

- $f(x) = \sqrt{2/\pi} e^{x^2/2}$
- $g(x) = e^{-x}$
- $C = \sqrt{2e/\pi}$

Si può generare X come segue:

- Genera $Y \sim Exp(1)$
- Genera $U \sim U([0, 1])$
- Fissa $X = Y$ se $U \leq f(Y)/C g(Y)$, se no torna a 1.
- Genera $B \sim Be(\frac{1}{2})$
- Fissa $Z = X$ se $B = 1$, $Z = -X$ se $B = 0$

Box-Muller Transformation

Consideriamo due variabili aleatorie indipendenti $X, Y \sim U([0, 1])$ e consideriamo la trasformazione di Box-Muller $(v, w) = BM(x, y)$ $v = \sqrt{-2 \log(x)} \cos(2\pi y)$ e $w = \sqrt{-2 \log(x)} \sin(2\pi y)$. Le variabili aleatorie $(V, W) = BM(X, Y)$ sono indipendenti ($V \perp\!\!\!\perp W$) e tali che $V, W \sim N(0, 1)$. Questo risultato deriva dalla relazione:

$J = \begin{bmatrix} \frac{\partial x}{\partial v} & \frac{\partial x}{\partial w} \\ \frac{\partial y}{\partial v} & \frac{\partial y}{\partial w} \end{bmatrix}$

dove $f_{X,Y}(x, y) = 1$ è det(J) è il determinante della Jacobiana. La funzione $f_{V,W}$ è la funzione di densità di due gaussiane standard indipendenti. Quindi, per ottenere n campioni $N(0, 1)$ è sufficiente campionare $n/2$ volte $X \sim U([0, 1])$, $n/2$ volte $Y \sim U([0, 1])$ e applicare la trasformazione di Box-Muller.

- Si può ottenere due campioni $X, Y \sim N(0, 1)$ come segue:

- Genera $U_1, U_2 \sim U([0, 1])$
- Fissa $X = \sqrt{-2 \log(U_1)} \cos(2\pi U_2)$
 $Y = \sqrt{-2 \log(U_1)} \sin(2\pi U_2)$

Multivariate Random Variable Generation

Sia $\mathbf{X} = (X_1, \dots, X_n) \in \mathbb{R}^n$ il vettore aleatorio la cui funzione di ripartizione congiunta è $F(\mathbf{z}) = F((z_1, \dots, z_n)) = \mathbb{P}(X_1 \leq z_1, \dots, X_n \leq z_n)$ e la cui funzione di densità congiunta è $f(\mathbf{x}) = f((x_1, \dots, x_n))$. L'obiettivo è riuscire a generare \mathbf{X} . Il metodo dell'Inverse-transform non è utilizzabile siccome $F : \mathbb{R}^n \rightarrow \mathbb{R}$ non è invertibile. L'acceptance-rejection method può essere applicato, tuttavia è molto difficile trovare una funzione ausiliare $g(\cdot)$ efficiente. Ci concentriamo allora sul campionamento di \mathbf{X} in alcuni casi convenienti: il caso delle componenti indipendenti e il caso delle distribuzioni condizionate. Dopodiché considereremo l'uso delle trasformazioni e il caso speciale della gaussiana multivariata.

Independent Components

Supponiamo che le componenti X_i ($i = 1, \dots, n$) di \mathbf{X} siano indipendenti tra loro, ciascuna con la propria funzione di ripartizione $F_i(\cdot)$ e la propria funzione di densità $f_i(\cdot)$. Allora, per ottenere un campione di \mathbf{X} è possibile generare ciascuna X_i indipendentemente dalle altre usando metodi di campionamento univariato.

Si può campionare $\mathbf{X} = (X_1, \dots, X_n)$ con componenti indipendenti, ciascuna con la propria funzione di ripartizione F_i come segue:

- for $i = 1, \dots, n$
 Genera $X_i \sim F_i(\cdot)$
 end for
- Fissa $\mathbf{X} = (X_1, \dots, X_n)$

Conditional Distributions

Supponiamo che le componenti di \mathbf{X} siano dipendenti e che le distribuzioni condizionate di $X_i | X_{i-1}, \dots, X_1$ siano conosciute esplicitamente. In questo caso è possibile fattorizzare la funzione di ripartizione congiunta:

$$\begin{aligned} F(\mathbf{z}) &= \mathbb{P}(\mathbf{X} \leq \mathbf{z}) \\ &= \mathbb{P}(X_1 \leq z_1) \mathbb{P}(X_2 \leq z_2 | X_1 \leq z_1) \dots \mathbb{P}(X_n \leq z_n | X_1 \leq z_1, \dots, X_{n-1} \leq z_{n-1}) \\ &= F_{X_1}(z_1) F_{X_2|X_1}(z_2 | z_1) \dots F_{X_n|X_{n-1}, \dots, X_1}(z_n | z_{n-1}, \dots, z_1) \end{aligned}$$

Per ottenere un campione \mathbf{X} generiamo le sue componenti X_i dalle funzioni di ripartizione $F_{X_i|X_{i-1}, \dots, X_1}(\cdot | X_{i-1}, \dots, X_1)$ usando un metodo di campionamento univariato.

Si può campionare $\mathbf{X} = (X_1, \dots, X_n)$ con componenti dipendenti sfruttando le distribuzioni condizionate come segue:

- Genera $X_1 \sim F_{X_1}(\cdot)$
- for $i = 2, \dots, n$
 Genera $X_i \sim F_{X_i|X_{i-1}, \dots, X_1}(\cdot | X_{i-1}, \dots, X_1)$
 end for
- Fissa $\mathbf{X} = (X_1, \dots, X_n)$

Generation by Transformation: Copulas

Supponiamo di voler campionare $\mathbf{X} = (X_1, \dots, X_n)$ con componenti dipendenti e con distribuzioni marginali $F_i : \mathbb{R} \rightarrow [0, 1]$. Conosciamo solo le distribuzioni marginali, non la distribuzione congiunta. Per descrivere/modellare le relazioni tra le componenti introduciamo le *copule*. Una copula è una funzione di ripartizione $C : [0, 1]^n \rightarrow [0, 1]$ di n variabili aleatorie U_1, \dots, U_n uniformemente distribuite e dipendenti tra loro:

$$C(u_1, \dots, u_n) = \mathbb{P}(U_1 \leq u_1, \dots, U_n \leq u_n)$$

La copula è una funzione che collega tutte le marginali F_i e permette di esprimere la distribuzione congiunta di \mathbf{X} :

$$F_{\mathbf{X}}(\mathbf{x}) = \mathbb{P}(X_1 \leq x_1, \dots, X_n \leq x_n) = C(F_1(x_1), \dots, F_n(x_n))$$

L'uso delle copule è equivalente a dare alla generazione di (U_1, \dots, U_n) il compito di rendere dipendenti le componenti di \mathbf{X} . La copula C contiene quindi le informazioni sulle dipendenze tra le componenti X_1, \dots, X_n , mentre le funzioni di ripartizione $F_i(\cdot)$ contengono le informazioni sulle distribuzioni marginali. Il teorema di Sklar ci assicura che ogni distribuzione multivariata congiunta può essere scritta in termini di una copula e di distribuzioni marginali univariate. Nella pratica quindi, partendo da una procedura in grado di generare $(U_1, \dots, U_n) \sim C$ possiamo ottenere il campione \mathbf{X} fissando $\mathbf{X} = (X_1, \dots, X_n) = (F_1^{-1}(U_1), \dots, F_n^{-1}(U_n))$.

■ Un tipico esempio è la *copula gaussiana*. Supponiamo di voler generare un campione $\mathbf{X} = (X_1, \dots, X_n)$ con distribuzioni marginali F_1, \dots, F_n . Generiamo $\mathbf{Y} = (Y_1, \dots, Y_n) \sim N(0, \Sigma)$, con Σ matrice $n \times n$ reale, simmetrica e definita positiva. Denotiamo $\sigma_i = \sqrt{Var(Y_i)}$ e definiamo il vettore $\mathbf{U} = (U_1, \dots, U_n) = (\Phi(Y_1/\sigma_1), \dots, \Phi(Y_n/\sigma_n))$, dove la funzione $\Phi(\cdot)$ è la funzione di ripartizione della gaussiana standard. La funzione $\Phi(\cdot)$ porta ciascuna componente di \mathbf{Y} in $[0, 1]$, per cui il vettore aleatorio \mathbf{U} è composto da n variabili aleatorie uniformemente distribuite e dipendenti tra loro. Pertanto $C_G^\Sigma(u_1, \dots, u_n) = \mathbb{P}(U_1 \leq u_1, \dots, U_n \leq u_n)$ è una copula e viene chiamata *copula gaussiana*. Per generare un vettore \mathbf{X} con copula C_G^Σ e marginali F_1, \dots, F_n è sufficiente fissare $\mathbf{X} = (F_1^{-1}(U_1), \dots, F_n^{-1}(U_n))$. Come campioniamo da $N(0, \Sigma)$?

Gaussian Copula.

Si può campionare $\mathbf{X} = (X_1, \dots, X_n)$ con componenti dipendenti e con distribuzioni marginali F_1, \dots, F_n sfruttando una copula gaussiana C_G^Σ come segue:

- Genera $\mathbf{Y} \sim N(0, \Sigma)$
- Calcola $\mathbf{U} = (\Phi(\frac{Y_1}{\sigma_1}), \dots, \Phi(\frac{Y_n}{\sigma_n}))$
- Fissa $\mathbf{X} = (F_1^{-1}(U_1), \dots, F_n^{-1}(U_n))$

Multivariate Gaussian Distribution

Vogliamo campionare $\mathbf{X} \sim N(\mu, \Sigma)$, dove $\mu \in \mathbb{R}^n$ è il vettore della media e $\Sigma \in \mathbb{R}^{n \times n}$ è la matrice covarianza, quindi simmetrica definita positiva. Siccome è definita positiva, Σ si può sempre fattorizzare in $\Sigma = AA^T$. Ci sono due modi per calcolare A .

Fattorizzazione di Choleski

Con Choleski otteniamo A matrice triangolare inferiore.

Decomposizione Spettrale

Si può diagonalizzare $\Sigma = VDV^T$ con $D = diag(\lambda_1, \dots, \lambda_n)$ matrice degli autovalori e V matrice (ortonormale) degli autovettori.

Fissiamo quindi $A = VD^{1/2}$, dove $D^{1/2} = diag(\sqrt{\lambda_1}, \dots, \sqrt{\lambda_n})$.

Partendo dalla fattorizzazione di Σ e generando $\mathbf{Y} \sim N(0, I)$ è possibile ottenere $\mathbf{X} \sim N(\mu, \Sigma)$ come $\mathbf{X} = \mu + A\mathbf{Y}$.

Si può campionare $\mathbf{X} = (X_1, \dots, X_n) \sim N(\mu, \Sigma)$ come segue:

- Ricava la fattorizzazione $\Sigma = AA^T$
- Genera $\mathbf{Y} \sim N(0, I)$, cioè $\mathbf{Y} = (Y_1, \dots, Y_n)$, $Y_i \sim N(0, 1)$
- Fissa $\mathbf{X} = \mu + A\mathbf{Y}$

Infatti, l'output dell'algoritmo, \mathbf{X} , è un vettore gaussiano multivariato tale che $\mathbb{E}[\mathbf{X}] = \mu$ e $Cov(\mathbf{X}) = \mathbb{E}[(\mathbf{X} - \mu)(\mathbf{X} - \mu)^T] = \mathbb{E}[A\mathbf{Y}\mathbf{Y}^TA] = \Sigma$.

Conditional Multivariate Gaussian Distribution

Supponiamo di voler campionare $\mathbf{X} = (X_1, \dots, X_n)$ e supponiamo che solo le variabili $(X_{k+1}, \dots, X_n) = \mathbf{Z}$ sono osservabili. L'idea è di generare le rimanenti variabili $(X_1, \dots, X_k) = \mathbf{Y}$ secondo la distribuzione condizionata $\mathbf{Y}|\mathbf{Z} = \mathbf{z}$, che sappiamo essere gaussiana. Questo perché, se abbiamo un vettore aleatorio gaussiano e lo condizioniamo ad un sottoinsieme delle sue componenti otteniamo che il condizionato è ancora gaussiano.

In particolare, dividiamo il vettore $\mathbf{X} \sim N(\mu, \Sigma)$ in:

$$\mathbf{X} = (X_1, \dots, X_k, X_{k+1}, \dots, X_n) = (\mathbf{Y}, \mathbf{Z})$$

dove \mathbf{Z} è osservabile e i \mathbf{Y} non è osservabile. Di conseguenza, media e varianza possono essere divise in:

$$\mu = (\mu_Y, \mu_Z) \quad \Sigma = \begin{pmatrix} \Sigma_{YY} & \Sigma_{YZ} \\ \Sigma_{ZY} & \Sigma_{ZZ} \end{pmatrix}$$

dove: $\mu_Y = \mathbb{E}[\mathbf{Y}]$

$$\mu_Z = \mathbb{E}[\mathbf{Z}]$$

$$\Sigma_{YY} = \mathbb{E}[(\mathbf{Y} - \mu_Y)(\mathbf{Y} - \mu_Y)^T]$$

$$\Sigma_{ZZ} = \mathbb{E}[(\mathbf{Z} - \mu_Z)(\mathbf{Z} - \mu_Z)^T]$$

$$\Sigma_{YZ} = \mathbb{E}[(\mathbf{Y} - \mu_Y)(\mathbf{Z} - \mu_Z)^T] = \Sigma_{ZY}^T$$

La distribuzione condizionata di \mathbf{Y} dato \mathbf{Z} è una gaussiana multivariata

$$\mathbf{Y}|\mathbf{Z} = \mathbf{z} \sim N(\boldsymbol{\mu}_{Y|Z}, \Sigma_{Y|Z})$$

dove: $\boldsymbol{\mu}_{Y|Z} = \boldsymbol{\mu}_Y + \Sigma_{YZ}\Sigma_{ZZ}^{-1}(\mathbf{z} - \boldsymbol{\mu}_Z)$
 $\Sigma_{Y|Z} = \Sigma_{YY} - \Sigma_{YZ}\Sigma_{ZZ}^{-1}\Sigma_{ZY}$

Sia $\mathbf{X} = (X_1, \dots, X_n) \sim N(\boldsymbol{\mu}, \Sigma)$. Supponiamo che $(X_1, \dots, X_k) = \mathbf{Y}$ non siano osservabili e che $(X_{k+1}, \dots, X_n) = \mathbf{Z}$ siano osservabili.

Sia \mathbf{z} il vettore delle osservazioni (realizzazione di \mathbf{Z}).

Si può generare $\mathbf{Y}|Z = \mathbf{z}$ come segue:

1. Generiamo $\mathbf{X} \sim N(\boldsymbol{\mu}, \Sigma)$
2. Indichiamo $\mathbf{X} = (X_1, \dots, X_k, X_{k+1}, \dots, X_n) = (\mathcal{Y}, \mathcal{Z})$
3. Fissiamo $\mathbf{Y} = \mathcal{Y} + \Sigma_{YZ}\Sigma_{ZZ}^{-1}(\mathbf{z} - \boldsymbol{\mu}_Z)$

In alternativa si può partire direttamente dalla distribuzione di $\mathbf{Y}|Z = \mathbf{z}$ e fattorizzare (Cholesky/decomp. spettrale) la matrice $\Sigma_{Y|Z} = AA^T$. Tuttavia, questo secondo approccio è più pesante (computazionalmente) rispetto a quello proposto, nonostante in quello proposto ci sia la necessità di invertire una matrice.

Gaussian Process Generation

I *Processi Gaussiani* possono essere pensati come una generalizzazione dei vettori gaussiani multivariati. Consideriamo $I \subset \mathbb{R}^d$ e definiamo una collezione di variabili aleatorie indicizzate da I : $\{X_t\}_{t \in I}$. La collezione $\{X_t\}_{t \in I}$ è detta *processo stocastico* se $d = 1$ e *random field* se $d \geq 1$. Un *processo gaussiano* è un *processo stocastico/random field per cui ogni sottoinsieme finito delle sue componenti si comporta come un vettore aleatorio gaussiano*, cioè $\forall n \in \mathbb{N}$ e $\forall t_1, \dots, t_n \in I$ si ha che il vettore aleatorio $\mathbf{X} = (X_{t_1}, \dots, X_{t_n})$ è gaussiano (multivariato).

Un processo gaussiano è interamente caratterizzato dalla funzione media $\mu_X : I \rightarrow \mathbb{R}$ e dalla funzione covarianza $C_X : I \times I \rightarrow \mathbb{R}$:

$$\mu_X(t) = \mathbb{E}[X_t] \quad C_X(t, s) = \mathbb{E}[(X_t - \mu_X(t))(X_s - \mu_X(s))]$$

Per questo motivo scriveremo $X \sim GP(\mu_X, C_X)$.

Un processo gaussiano è detto *debolmente stazionario* se $C_X(s, t)$ dipende unicamente da $(s - t)$. È invece detto *fortemente stazionario* se è debolmente stazionario e $\mu_X(t)$ non dipende da t .

Partendo da media e varianza è possibile generare un processo gaussiano nei punti t_1, \dots, t_n semplicemente campionando un vettore aleatorio gaussiano $\mathbf{X} = (X_{t_1}, \dots, X_{t_n}) \sim N(\boldsymbol{\mu}, \Sigma)$ con media $\boldsymbol{\mu} = (\mu_X(t_1), \dots, \mu_X(t_n))$ e varianza $\Sigma_{ij} = C_X(t_i, t_j)$ usando uno dei metodi già visti: si può ad esempio fattorizzare la matrice covarianza $\Sigma = AA^T$, generare $\mathbf{Z} \sim N(0, I)$ e applicare la trasformazione $\mathbf{X} = \boldsymbol{\mu} + A\mathbf{Z}$.

Gaussian Process Generation.

È possibile campionare un processo gaussiano $X \sim GP(\mu_X, C_X)$ nei punti $t_1, \dots, t_n \in I$ come segue:

1. Calcola $\boldsymbol{\mu} = (\mu_X(t_1), \dots, \mu_X(t_n))$ e $\Sigma_{ij} = C_X(t_i, t_j)$
2. Calcola la decomposizione di Cholesky $\Sigma = AA^T$
3. Genera $Z_1, \dots, Z_n \sim N(0, 1)$ e definisci $\mathbf{Z} = (Z_1, \dots, Z_n)$
4. Fissa $\mathbf{X} = \boldsymbol{\mu} + A\mathbf{Z}$

Trattandosi di vettori gaussiani, se partissimo da $\mathbf{Z} = (X_{t_1}, \dots, X_{t_n})$ e volessimo generare $\mathbf{Y} = (X_{t_{n+1}}, \dots, X_m)$ potremmo direttamente generare $\mathbf{Y}|Z = \mathbf{z}$ sfruttando l'algoritmo relativo alla generazione di vettori gaussiani multivariati condizionati a vettori gaussiani multivariati (più leggero rispetto a Cholesky).

3 Random Inputs Parametrization

Il passaggio critico nella simulazione di un sistema stocastico è quello di caratterizzare correttamente gli input aleatori. Spesso è necessario ridurre la dimensione dell'input (per trattare ad esempio processi gaussiani) e questo comporta degli errori di approssimazione. Inoltre, è fondamentale che gli input vengano rappresentati (da variabili aleatorie) come mutuamente indipendenti.

Random Parameters

Se gli input aleatori di un sistema sono i parametri del sistema, la procedura di parametrizzazione è semplice (in quanto gli input sono già sotto forma di parametri). Nei casi più semplici infatti, il modello matematico dipende da n parametri $\mathbf{Y} = (Y_1, \dots, Y_n) \in \mathbb{R}^n$ caratterizzati da una funzione di ripartizione $F_{\mathbf{Y}}(\cdot)$. L'obiettivo è quello di trovare un insieme finito di variabili aleatorie mutuamente indipendenti $\mathbf{Z} = (Z_1, \dots, Z_p) \in \mathbb{R}^p$, con $1 \leq p \leq n$, e una funzione $T : \mathbb{R}^p \rightarrow \mathbb{R}^n$ tali che $\mathbf{Y} = T(\mathbf{Z})$.

- Consideriamo un'equazione differenziale ordinaria con due parametri:

$$\begin{cases} u'(t, \omega) = -\alpha(\omega) u(t, \omega) & t \in (0, T) \\ u(0, \omega) = \beta(\omega) \end{cases}$$

con α e β aleatori. Il vettore dei parametri del sistema è quindi dato da $\mathbf{Y}(\omega) = (\alpha(\omega), \beta(\omega)) \in \mathbb{R}^2$. Se α e β sono indipendenti possiamo semplicemente definire il vettore aleatorio $\mathbf{Z}(\omega) = \mathbf{Y}(\omega)$. In alternativa, esiste una funzione $f(\cdot)$ che lega α e β , quindi tale che $f(\alpha, \beta) = 0$. In tal caso è possibile definire una variabile aleatoria $Z(\omega)$ che parametrizza questa relazione: $\alpha(\omega) = a(Z(\omega))$, $\beta(\omega) = b(Z(\omega))$, $f(a, b) = 0$.

Random Fields and Dimensionality Reduction

In molti casi gli input aleatori sono processi stocastici indicizzati da un insieme I di dimensione infinita. Sia $\{Y_t\}_{t \in I}$ il processo stocastico che modella gli input aleatori. L'obiettivo è quello di trovare un insieme finito di variabili aleatorie mutuamente indipendenti $\mathbf{Z} = (Z_1, \dots, Z_d)$, con $d \geq 1$ finito, e una funzione R tale che $Y_t = R(\mathbf{Z})$. Siccome d è un valore finito, la trasformazione non può essere esatta: $Y_t \approx R(\mathbf{Z})$.

Formal definition of Random Fields

Quando l'input aleatorio è *spatially-dependent*, cioè varia casualmente da un punto all'altro di un certo dominio, va rappresentato da un *random field*. Consideriamo un dominio fisico $D \subset \mathbb{R}^d$, con $d \geq 1$, e uno spazio Ω (spazio dei possibili esiti casuali). Un random field $a(x, \omega) : D \times \Omega \rightarrow \mathbb{R}$ è una collezione infinita di variabili aleatorie. In particolare, per una $x \in D$ fissata abbiamo che $a(x, \cdot)$ è una variabile aleatoria su Ω , per una $\omega \in \Omega$ fissata abbiamo che $a(\cdot, \omega)$ è una realizzazione del random field su D .

Consideriamo il caso scalare ($d = 1$) e per ogni random field (processo stocastico) definiamo il valore atteso $\bar{a} : D \rightarrow \mathbb{R}$, la funzione covarianza $Cov_a : D \times D \rightarrow \mathbb{R}$ e la funzione varianza $Var_a : D \rightarrow \mathbb{R}$:

- $\bar{a} = \mathbb{E}[a(x, \cdot)]$
- $Cov_a(x_1, x_2) = \mathbb{E}[(a(x_1, \cdot) - \bar{a}(x_1))(a(x_2, \cdot) - \bar{a}(x_2))]$
- $Var_a(x) = Cov_a(x, x)$

Un random field è detto *debolmente stazionario* se la sua legge è invariante per traslazione, ovvero se $a(x + h, \omega) \sim a(x, \omega) \forall h \in \mathbb{R}$. In questo caso $\mathbb{E}[a]$ è indipendente da x e $Cov_a(x_1, x_2) = Cov_a^*(x_2 - x_1)$. Un random field debolmente stazionario è detto *isotropo* se la funzione covarianza dipende solo da $\|x_1 - x_2\|$, cioè $Cov_a(x_1, x_2) = Cov_a^*(\|x_1 - x_2\|)$. In particolare, isotropo è diverso da debolmente stazionario perché non tiene conto della direzione.

Karhunen-Loeve Expansion

Un random field è caratterizzato dal valore atteso e dalle funzioni di varianza e covarianza. È possibile rappresentare un random field come una serie di infiniti termini costruita sfruttando la media e la varianza del random field. Questa rappresentazione è detta *Karhunen-Loeve Expansion*.

Consideriamo un dominio fisico $D \subset \mathbb{R}^d$, con $d \geq 1$, e uno spazio Ω . Definiamo un random field $a(x, \omega)$ con media $\mathbb{E}[a]$ e funzione covarianza $Cov_a(\cdot, \cdot)$. Supponiamo che la funzione covarianza sia simmetrica, non-negativa e continua. In aggiunta, definiamo l'operatore di Hilbert-Schmidt $T : L^2(D) \rightarrow L^2(D)$ come $(Tf)(x) = \int_D Cov_a(x, y) f(y) dy$. In questo contesto, il *teorema di Mercer* (*) garantisce l'esistenza di una sequenza di valori $\lambda_1 \geq \lambda_2 \geq \dots \geq 0$ tale che $\lim_n \lambda_n = 0$ e una corrispondente sequenza di funzioni $b_i : D \rightarrow \mathbb{R}$ tali che:

$$(Tb_i)(x) = \int_D Cov_a(x, y) b_i(y) dy = \lambda_i b_i(x)$$

$$\int_D b_i(x) b_j(x) dx = \delta_{ij}$$

Si tratta di una *eigen-decomposizione* dove λ_i sono gli autovettori e $b_i(x)$ sono le autofunzioni. La seconda equazione ci garantisce che le autofunzioni sono tra di loro ortogonali. Questo risultato è un'estensione infinito-dimensionale della decomposizione ai valori singolari (SVD).

(*) Mercer Theorem.

Sia $D \subset \mathbb{R}^d$ un dominio compatto e $K : D \times D \rightarrow \mathbb{R}$ un kernel continuo, simmetrico e definito semi-positivo. Sia $T_K : L^2(D) \rightarrow L^2(D)$ l'operatore di Hilbert-Schmidt definito come $(T_K f)(x) = \int_D K(x, y) f(y) dy$.

Allora esiste $\{b_i\}_{i \in \mathbb{N}}$ base ortonormale di $L^2(D)$ formata da autofunzioni di T_K tali che la corrispondente sequenza di autovalori $\{\lambda_i\}_{i \in \mathbb{N}}$ è non-negativa. Le autofunzioni degli autovalori non nulli sono continue in D e

$$K(x, y) = \sum_{i=1}^{\infty} \lambda_i b_i(x) b_i(y)$$

Inoltre:

$$\lim_N \sup_{x, y \in D} |K(x, y) - \sum_{i=1}^N \lambda_i b_i(x) b_i(y)| = 0$$

Definendo da una sequenza di variabili aleatorie $\{Y_i(\omega)\}_{i \in \mathbb{N}}$:
 proiettiamo il field
 (a cui ritroviamo la
 media) in diverse
 diver. $b_i(x)$

$$Y_i(\omega) = \frac{1}{\sqrt{\lambda_i}} \int_D (a(x, \omega) - \mathbb{E}[a](x)) b_i(x) dx$$

scorrelate tra di loro, con media zero e varianza unitaria, è possibile rappresentare il random field con la *Karhunen-Loeve Expansion*:

$$a(x, \omega) = \mathbb{E}[a](x) + \sum_{i=1}^{\infty} \sqrt{\lambda_i} b_i(x) Y_i(\omega)$$

È inoltre possibile troncare la serie dopo N elementi, ottenendo così una approssimazione finito-dimensionale del random field:

$$a_N(x, \omega) = \mathbb{E}[a](x) + \sum_{i=1}^N \sqrt{\lambda_i} b_i(x) Y_i(\omega)$$

In particolare, dal *teorema di Mercer* concludiamo che $a_N(x, \omega)$ è la miglior approssimazione (in termini di varianza) a N elementi di $a(x, \omega)$.

La *Karhunen-Loeve Expansion* è quindi un modo per parametrizzare un oggetto aleatorio infinito-dimensionale (random field/processo stocastico). La velocità di convergenza di $a_N(x, \omega)$ a $a(x, \omega)$ dipende dal decadimento degli autovalori (che dipende dalla regolarità della funzione covarianza e dalla lunghezza di correlazione). L'iperparametro N viene scelto in modo che la somma dei termini trascurati sia sufficientemente piccola.

I due casi limite sono quello in cui vi è zero correlazione e quello in cui vi è infinita correlazione. Nel primo caso $Cov_a(x, y) = \delta_{x,y}$ e $\lambda_i = 1 \forall i \in \mathbb{N}$. Non si ha quindi una decadenza di autovalori. Nel secondo caso invece $Cov_a(x, y) = 1$ e $\lambda_i = 0 \forall i > 1$. La decadenza degli autovalori è immediata, c'è dipendenza da una sola variabile aleatoria.

La varianza totale del random field, definita come l'integrale di $Var_a(x)$ su tutto il dominio D , può essere riscritta come:

$$\begin{aligned} \int_D Var_a(x) dx &= \int_D \sum_{i=1}^{\infty} \lambda_i b_i(x)^2 dx \\ &= \sum_{i=1}^{\infty} \lambda_i \int_D b_i(x)^2 dx = \sum_{i=1}^{\infty} \lambda_i \end{aligned}$$

Pertanto la versione troncata a N termini della *Karhunen-Loeve Expansion* spiega $\sum_{i=1}^N \lambda_i / \sum_{i=1}^{\infty} \lambda_i$ della varianza totale.

Partendo dalla versione troncata della *Karhunen-Loeve Expansion* è possibile generare $a(x, \omega)$ semplicemente campionando Y_1, \dots, Y_N . Il campione di $a(x, \omega)$ sarà quindi dato da $\mathbb{E}[a](x) + \sum_{i=1}^N \sqrt{\lambda_i} b_i(x) Y_i(\omega)$. Notiamo però che le variabili Y_i sono solo scorrelate, non indipendenti. Nel caso di processo gaussiano i due termini sono sinonimi, nel caso generale no.

Back to Gaussian Random Fields

Ricordiamo che un random field $a(x, \omega)$ è detto gaussiano se qualsiasi sottosinsieme finito delle sue componenti si comporta come un vettore aleatorio gaussiano, cioè se per qualsiasi $x_1, \dots, x_n \in D \subset \mathbb{R}^d$, $d \geq 1$, il vettore aleatorio $\mathbf{Y}(\omega) = (a(x_1, \omega), \dots, a(x_n, \omega))$ è gaussiano. Se $d = 1$ il random field viene chiamato processo gaussiano, se $d > 1$ allora si parla di gaussian random field.

Nel caso di random field gaussiano, le variabili Y_i della *Karhunen-Loeve Expansion* sono $N(0, 1)$ indipendenti. Pertanto, la *Karhunen-Loeve Expansion* è molto conveniente in caso di campionamento di processi gaussiani o gaussian random field.

Alcuni modelli per la funzione covarianza, nel caso di gaussian random field, sono *esponenziale* (ha la stessa regolarità di un moto Browniano), *gaussiano* (abbastanza smooth) o *Matern* (dipende dai suoi parametri). I modelli distano molto tra loro anche nel caso in cui la varianza del random field, σ^2 , e la lunghezza di correlazione, ℓ_c , siano per tutti uguali.

4 Monte Carlo Methods

Fino ad ora ci siamo concentrati su come simulare. Ora, invece, l'obiettivo è di sfruttare le simulazioni di un modello statistico per studiarne le proprietà. Se campioniamo un gran numero di volte un dato modello, allora i campioni rifletteranno il comportamento statistico del modello.

Molti problemi sono riconducibili al calcolo del valore atteso di una variabile aleatoria. L'obiettivo è quindi calcolare $\mathbb{E}[\Psi(X)]$, dove $X \sim f(\cdot)$ è una variabile aleatoria (la cui distribuzione è caratterizzata dalla funzione di densità $f(\cdot)$) e $\Psi : \mathbb{R} \rightarrow \mathbb{R}$ è una funzione che determina una *Quantity of Interest* (QoI). Alcune volte è possibile procedere analiticamente, ovvero calcolando $\mathbb{E}[\Psi(X)] = \int \Psi(x) f(x) dx$. Quando questo non è possibile si può provare con dei metodi di quadratura, tuttavia questo è impraticabile quando X è un vettore aleatorio di dimensione troppo elevata. Introduciamo il metodo di Monte Carlo. L'integrazione con Monte Carlo si basa sulla *Legge dei Grandi Numeri*: se $\{\mathbf{X}^{(i)}\}_{i \in \mathbb{N}}$ è una sequenza di variabili aleatorie indipendenti e identicamente distribuite con la stessa distribuzione di X , allora con probabilità 1 si avrà $\mathbb{E}[\Psi(X)] = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=1}^N \Psi(\mathbf{X}^{(i)})$.

Monte Carlo Method.

Sia $X \sim f(\cdot)$ una variabile aleatoria e $\Psi : \mathbb{R} \rightarrow \mathbb{R}$ una funzione. Il metodo Monte Carlo per il calcolo della stima di $\mathbb{E}[\Psi(X)]$ è un metodo numerico basato sull'approssimazione:

$$\mathbb{E}[\Psi(X)] \approx \frac{1}{N} \sum_{i=1}^N \Psi(\mathbf{X}^{(i)})$$

dove $\{\mathbf{X}^{(i)}\}_{i=1}^N$ sono campioni indipendenti e identicamente distribuiti secondo la stessa distribuzione di X .

Per poter stimare $\mathbb{E}[\Psi(X)]$ è quindi necessario generare un gran numero di campioni di X . È possibile procedere con uno dei metodi introdotti nelle sezioni precedenti. Assumiamo che non sia un problema ottenere campioni e ci concentriamo solo sulle proprietà del metodo Monte Carlo.

Properties of the Monte Carlo Estimator

Consideriamo una variabile aleatoria Z e supponiamo che questa sia direttamente la QoI (è equivalente a considerare Ψ come funzione identità). L'obiettivo è calcolare $\mu = \mathbb{E}[Z]$. Secondo il metodo Monte Carlo, è sufficiente generare N campioni $Z^{(1)}, \dots, Z^{(N)}$ indipendenti e identicamente distribuiti secondo la distribuzione di Z e fissare $\hat{\mu} = \frac{1}{N} \sum_{i=1}^N Z^{(i)}$. Non stiamo facendo assunzioni sulla distribuzione di Z , tuttavia è necessario che $\text{Var}(Z) = \sigma^2 < \infty$. Siccome $\hat{\mu}$ è costruita a partire da variabili aleatorie, anche $\hat{\mu}$ è una variabile aleatoria e ha le seguenti proprietà:

- $\hat{\mu}$ è *unbiased*, cioè $\mathbb{E}[\hat{\mu}] = \mu$
- $\text{Var}(\hat{\mu}) = \sigma^2/N$
- $\hat{\mu} \rightarrow \mu$ quasi certamente per $N \rightarrow \infty$
- $\frac{\sqrt{N}(\hat{\mu} - \mu)}{\sigma} \rightarrow N(0, 1)$ in distribuzione per $N \rightarrow \infty$

Il terzo risultato si basa sulla *Legge dei Grandi Numeri*.

(Strong) Law of Large Numbers.

Sia Z una variabile aleatoria e siano $Z^{(1)}, Z^{(2)}, Z^{(3)}, \dots$ dei campioni indipendenti e identicamente distribuiti tali che $\mathbb{E}[|Z^{(i)}|] < \infty$ e $\mathbb{E}[Z^{(i)}] = \mu$. Allora $\hat{\mu} \rightarrow \mu$ quasi certamente per $N \rightarrow \infty$, cioè:

$$\mathbb{P}\left(\frac{1}{N} \sum_{i=1}^N Z^{(i)} \rightarrow \mu\right) = 1$$

L'ultimo risultato è basato invece sul *Teorema Centrale del Limite*.

Central Limit Theorem.

Sia Z una variabile aleatoria e siano $Z^{(1)}, Z^{(2)}, Z^{(3)}, \dots$ dei campioni indipendenti e identicamente distribuiti tali che $\mathbb{E}[Z^{(i)}] = \mu$ e $\text{Var}(Z^{(i)}) = \sigma^2 < \infty$. Sia $\hat{\mu} = \frac{1}{N} \sum_{i=1}^N Z^{(i)}$. Allora

$$\frac{\sqrt{N}(\hat{\mu} - \mu)}{\sigma} \rightarrow N(0, 1) \text{ in distribuzione per } N \rightarrow \infty$$

Partendo dal teorema centrale del limite è possibile ricavare degli intervalli di confidenza asintotici di livello $1 - \alpha$:

$$I_\alpha = \left[\hat{\mu} - z_{1-\frac{\alpha}{2}} \frac{\sigma}{\sqrt{N}}, \hat{\mu} + z_{1-\frac{\alpha}{2}} \frac{\sigma}{\sqrt{N}} \right]$$

Dove z_α è l' α -quantile della normale standard, cioè $\Phi(z_\alpha) = \alpha$, con Φ funzione di ripartizione della gaussiana standard. Il significato di I_α è che $\mathbb{P}(\mu \in I_\alpha) \rightarrow 1 - \alpha$ per $N \rightarrow \infty$. Equivalentemente possiamo dire che con probabilità $1 - \alpha$, asintoticamente:

$$|\mu - \hat{\mu}| \leq z_{1-\frac{\alpha}{2}} \frac{\sigma}{\sqrt{N}}$$

Il teorema centrale del limite mostra che l'errore del metodo Monte Carlo è dell'ordine di $N^{-1/2}$, che comporta una bassa velocità di convergenza (per ridurre l'errore di 10 bisogna moltiplicare N di 100). Tuttavia, Monte Carlo richiede solo un'ipotesi per funzionare ($\text{Var}(Z) < \infty$). In aggiunta, l'ordine dell'errore non dipende dalla dimensione in cui si trova Z .

La precedente stima dell'errore non è pratica, siccome $\sigma^2 = \text{Var}(Z)$ di solito è sconosciuto. Possiamo sostituire σ^2 con un suo stimatore $\hat{\sigma}^2$ anch'esso *unbiased* e convergente quasi certamente a σ^2 :

$$\hat{\sigma}^2 = \frac{1}{N-1} \sum_{i=1}^N (Z^{(i)} - \hat{\mu})^2$$

Quindi, un intervallo di confidenza computabile è dato da:

$$\hat{I}_\alpha = \left[\hat{\mu} - z_{1-\frac{\alpha}{2}} \frac{\hat{\sigma}}{\sqrt{N}}, \hat{\mu} + z_{1-\frac{\alpha}{2}} \frac{\hat{\sigma}}{\sqrt{N}} \right]$$

Sotto ipotesi più rigorose è possibile migliorare l'errore. Se supponiamo infatti che le $Z^{(i)}$ sono indipendenti e identicamente distribuite con media zero e varianza unitaria, allora, asintoticamente:

$$|\mu - \hat{\mu}| \leq \frac{\sigma}{\sqrt{N}} \sqrt{2 \log(\log(N))}$$

dove guadagnamo $\sqrt{2 \log(\log(N))}$ che prova a bilanciare \sqrt{N} .

Practical Aspects

L'output di una simulazione Monte Carlo dovrebbe sempre avere, oltre a una stima puntuale, una stima dell'errore (ad esempio, l'intervallo di confidenza asintotico di livello $1 - \alpha$ \hat{I}_α). In pratica, la scelta di N è fatta in modo da ottenere una certa tolleranza, ad esempio:

$$|\hat{I}_\alpha| = |2 z_{1-\frac{\alpha}{2}} \frac{\hat{\sigma}}{\sqrt{N}}| \leq 2 \text{ tol}$$

Per ottenere N stimiamo $\hat{\sigma}^2_N$, con $\hat{\sigma}_N$ non troppo grande, e fissiamo:

$$N = \frac{z_{1-\alpha/2}^2 \hat{\sigma}_N^2}{\text{tol}^2}$$

Dopodiché applichiamo Monte Carlo e ricalcoliamo $\hat{\sigma}_N^2$. Se $\hat{\sigma}_N^2 > \hat{\sigma}_N^2$ allora fissiamo $\tilde{N} = N$ e ricalcoliamo una nuova N .

Monte Carlo in Higher Dimensions

Le precedenti considerazioni si possono facilmente estendere se consideriamo un vettore aleatorio $Z = (Z_1, \dots, Z_m)$. L'obiettivo è stimare il valore atteso di Z , definito da $\mu = \mathbb{E}[Z]$. Partendo da $Z^{(1)}, \dots, Z^{(N)}$, cioè N campioni indipendenti e identicamente distribuiti secondo la distribuzione di Z , la stima di μ secondo il metodo Monte Carlo è data da:

$$\hat{\mu} = \frac{1}{N} \sum_{i=1}^N Z^{(i)}$$

La stima della matrice covarianza di Z è invece data da:

$$\hat{C} = \frac{1}{N-1} \sum_{i=1}^N (Z^{(i)} - \hat{\mu})(Z^{(i)} - \hat{\mu})^T$$

Le stime $\hat{\mu}$ e \hat{C} sono tali che $\hat{\mu} \rightarrow \mu$ e $\hat{C} \rightarrow C$. Inoltre, vale la relazione $(\hat{\mu} - \mu)^T \hat{C}^{-1} (\hat{\mu} - \mu) \rightarrow \chi_m^2$ in distribuzione. Per cui è possibile ottenere una regione di confidenza asintotica a livello $1 - \alpha$:

$$\hat{I}_\alpha = \{y \in \mathbb{R}^m : (\hat{\mu} - \mu)^T \hat{C}^{-1} (\hat{\mu} - \mu) \leq \chi_{m,1-\alpha}^2\}$$

Monte Carlo to compute Integrals

Consideriamo il modello stocastico $Z = \Psi(\mathbf{X})$, dove $\mathbf{X} = (X_1, \dots, X_n)$ è un vettore aleatorio la cui distribuzione congiunta è caratterizzata dalla funzione di densità $f(\cdot)$ e $\Psi : \mathbb{R}^n \rightarrow \mathbb{R}$ è una funzione. L'obiettivo è di calcolare $\mu = \mathbb{E}[Z] = \int_{\mathbb{R}^n} \Psi(\mathbf{x}) f(\mathbf{x}) d\mathbf{x}$. Possiamo ottenere un'approssimazione di μ con il metodo Monte Carlo: è sufficiente generare N campioni $\mathbf{X}^{(i)}$ indipendenti e identicamente distribuiti secondo la distribuzione di \mathbf{X} e fissare $\hat{\mu} = \frac{1}{N} \sum_{i=1}^N \Psi(\mathbf{X}^{(i)})$.

Siccome $\hat{\mu} \approx \int_{\mathbb{R}^n} \Psi(\mathbf{x}) f(\mathbf{x}) d\mathbf{x}$, la soluzione di Monte Carlo può essere vista come un metodo di quadratura per l'approssimazione dell'integrale ($\mathbf{X}^{(i)}$ sono i nodi e $1/N$ sono i pesi da assegnare a ciascun nodo). Partendo da questa osservazione è possibile ragionare in modo inverso e sfruttare il metodo Monte Carlo per approssimare un generico integrale, senza doverlo inserire in un contesto statistico.

Consideriamo il problema del calcolo del generico integrale

$$I = \int_{\mathbb{R}^n} \Psi(\mathbf{x}) w(\mathbf{x}) d\mathbf{x}$$

dove $\Psi : \mathbb{R}^n \rightarrow \mathbb{R}$ è l'integrandi e $w : \mathbb{R}^n \rightarrow \mathbb{R}$ è la funzione non-negativa di pesi, cioè $\int_{\mathbb{R}^n} w(\mathbf{x}) d\mathbf{x} = 1$. Possiamo ottenere un'approssimazione di I con il metodo Monte Carlo: è sufficiente campionare N variabili aleatorie $\mathbf{X}^{(i)}$ indipendenti e identicamente distribuite secondo la funzione densità $w(\cdot)$ e fissare poi $\hat{I} = \frac{1}{N} \sum_{i=1}^N \Psi(\mathbf{X}^{(i)})$.

Abbiamo quindi un metodo per calcolare gli integrali con una velocità di convergenza dell'ordine $N^{-1/2}$. Non è una buona velocità, tuttavia è indipendente dalla dimensione n dello spazio in cui ci troviamo. L'errore di approssimazione dei metodi di quadratura, ad esempio, dipende dalla dimensione dello spazio.

■ Consideriamo un dominio $D \subset [0, 1]^d$ e supponiamo di voler calcolare il volume di D , quindi di voler calcolare $\int_D \mathbf{1}_D(\mathbf{x}) d\mathbf{x}$. L'approssimazione data dal metodo di Monte Carlo comporta un errore dell'ordine di $N^{-1/2}$ mentre l'approssimazione data dal metodo di quadratura comporta un errore dell'ordine di $N^{-2/d}$. Quindi, per $d > 4$ la velocità di convergenza del metodo di quadratura sarà peggiore di quella di Monte Carlo.

■ Esempio 2. Come calcolare $\int_a^b h(x) dx$

(Smooth) Functions of Expectations

Consideriamo un modello stocastico con molteplici variabili output $Z =$

(Z_1, \dots, Z_m) e supponiamo che l'obiettivo sia calcolare/stimare una funzione dei valori attesi di Z , cioè $\zeta = f(\mathbb{E}[Z_1], \dots, \mathbb{E}[Z_m])$ con $f : \mathbb{R}^n \rightarrow \mathbb{R}$ funzione regolare (dobbiamo essere in grado di calcolarne le derivate).

Per comodità indichiamo $\mu = (\mu_1, \dots, \mu_m) = (\mathbb{E}[Z_1], \dots, \mathbb{E}[Z_m])$.

Possiamo approssimare ζ con il metodo Monte Carlo: è sufficiente campionare N vettori aleatori $Z^{(1)}, \dots, Z^{(N)}$ indipendenti e identicamente distribuiti secondo la stessa distribuzione di Z e fissare $\hat{\zeta} = f(\hat{\mu}_1, \dots, \hat{\mu}_m)$, dove $\hat{\mu}_j = \frac{1}{N} \sum_{i=1}^N Z_j^{(i)}$.

Delta Method

Se $f(\cdot)$ è continua in μ abbiamo che $\hat{\zeta} \rightarrow \zeta$ quasi certamente. Tuttavia questo non è sufficiente: come possiamo determinare una stima dell'errore di ζ e un intervallo di confidenza? La soluzione è fornita dal *Delta Method*, basato sull'espansione di Taylor di primo ordine centrata in $\mu = \mathbb{E}[Z]$:

$$\hat{\zeta} - \zeta = f(\hat{\mu}) - f(\mu) = \nabla f(\mu) \cdot (\hat{\mu} - \mu) + o(\|\hat{\mu} - \mu\|)$$

Definendo $C = Cov(Z) = \mathbb{E}[(Z - \mu)(Z - \mu)^T]$ otteniamo una seconda versione del teorema centrale del limite:

$$\sqrt{N}(\hat{\zeta} - \zeta) \rightarrow N(0, \nabla f(\mu) \cdot C \nabla^T f(\mu)) \text{ in distribuzione}$$

È quindi possibile ottenere un intervallo di confidenza sostituendo C con:

$$\hat{C} = \frac{1}{N-1} \sum_{i=1}^N (Z^{(i)} - \hat{\mu})(Z^{(i)} - \hat{\mu})^T$$

e $\nabla f(\mu)$ con $\nabla f(\hat{\mu})$. Osserviamo però che lo stimatore $\hat{\zeta}$ è *biased*.

Recap and Next Steps

- Impariamo a campionare da $\mathcal{U}([0, 1])$.
→ *Uniform Pseudo RNG* (Chp. 1)
- Campioniamo da qualsiasi distribuzione (univariata, multivariata, processo gaussiano) partendo da una sequenza iid $\mathcal{U}([0, 1])$.
→ *Random Variable Generation* (Chp. 2)
- Impariamo a parametrizzare l'aleatorietà di un sistema stocastico (caso finito-dim. e caso infinito-dim. (random fields)).
→ *Random Inputs Parametrization* (Chp. 3)

Da ora diamo per scontato di poter campionare da qualsiasi distribuzione (univariata, multivariata, infinito-dimensionale) e di saper parametrizzare gli input di qualsiasi sistema stocastico.

- Impariamo a stimare il valore atteso di una variabile (o vettore) aleatoria con il metodo Monte Carlo.
→ *Monte Carlo Methods* (Chp. 4)

Lo stimatore $\hat{\mu}$ del metodo Monte Carlo è tale che:

$$|\mu - \hat{\mu}| \leq z_{1-\frac{\alpha}{2}} \frac{\hat{\sigma}}{\sqrt{N}}$$

- Non possiamo ridurre $1/\sqrt{N}$, questa è la velocità di convergenza.
- Come possiamo ridurre $\hat{\sigma}$?
→ *Variance Reduction Techniques* (Chp. 5)
- Come possiamo migliorare la costruzione di $\hat{\mu}$?
È possibile scegliere dei punti secondo regole adeguate in modo da ottenere una velocità di convergenza pari a $\log(N)/\sqrt{N}$. Tuttavia, i punti non saranno più random/aleatori, bensì scelti con schemi/criteri deterministicici.
→ *Quasi-Monte Carlo Formulas* (Chp. 6)
- Infine, è possibile ottimizzare ulteriormente la stima combinando diverse soluzioni di Monte Carlo.
→ *Multi-Fidelity/Multi-Level Monte Carlo* (Chp. 7)

5 Variance Reduction Techniques

L'obiettivo delle tecniche per la riduzione della varianza è di aumentare la precisione delle stime generate con il metodo Monte Carlo sfruttando informazioni sul modello stocastico in questione. Più informazioni si hanno sul modello e più si riesce a ridurre la varianza.

Consideriamo la variabile aleatoria Z , output di un modello stocastico, e assumiamo che Z sia una funzione $Z = \Psi(\mathbf{X})$ di un vettore aleatorio $\mathbf{X} = (X_1, \dots, X_d)$ la cui distribuzione è caratterizzata dalla funzione densità congiunta $f(\cdot)$. L'obiettivo è quello di calcolare il valore atteso di Z , cioè $\mu = \mathbb{E}[Z] = \int_{\mathbb{R}^d} \Psi(\mathbf{x}) f(\mathbf{x}) d\mathbf{x}$.

Abbiamo precedentemente introdotto la versione cruda del metodo Monte Carlo per approssimare il valore di μ : è sufficiente generare N campioni $Z^{(1)}, \dots, Z^{(N)}$, con $Z^{(i)} = \Psi(\mathbf{X}^{(i)})$ e $\mathbf{X}^{(i)}$ indipendenti e identicamente distribuiti secondo la legge di \mathbf{X} , e fissare $\hat{\mu}_{CMC} = \frac{1}{N} \sum_{i=1}^N Z^{(i)}$. Sfruttando il teorema del limite centrale abbiamo ottenuto:

$$|\mu - \hat{\mu}_{CMC}| \leq z_{1-\frac{\alpha}{2}} \frac{\sqrt{\text{Var}(Z)}}{\sqrt{N}}$$

con probabilità $1 - \alpha$, asintoticamente. Per ottenere dei limiti più stringenti per l'errore è necessario lavorare su $\text{Var}(Z)$, siccome non possiamo migliorare \sqrt{N} (non lavorando su $\text{Var}(Z)$, l'unico modo per avere dei limiti più stringenti è quello di aumentare N).

Le *Variance Reduction Techniques* si basano su un concetto semplice: invece di applicare lo stimatore della media su Z , cioè $\hat{\mu} = \hat{\mu}(Z)$, si applichi lo stimatore della media su una versione modificata di Z , diciamo \tilde{Z} , tale che $\mathbb{E}[\tilde{Z}] = \mathbb{E}[Z] = \mu$ e $\text{Var}(\tilde{Z}) \ll \text{Var}(Z)$. L'approssimazione di Monte Carlo con la riduzione della varianza sarà quindi:

$$\hat{\mu}_{VR} = \frac{1}{N} \sum_{i=1}^N \tilde{Z}^{(i)}$$

con $\tilde{Z}^{(i)}$ indipendenti, identicamente distribuite e tali che $\mathbb{E}[\tilde{Z}] = \mu$. Come costruire \tilde{Z} ?

Antithetic Variables

Supponiamo che N sia pari. Al posto di generare N campioni indipendenti e identicamente distribuiti, l'idea delle *Antithetic Variables* è di generare $N/2$ coppie di variabili aleatorie negativamente correlate

$$(Z^{(1)}, Z^{(2)}), (Z^{(3)}, Z^{(4)}), \dots, (Z^{(n-1)}, Z^{(n)})$$

Le coppie sono indipendenti e identicamente distribuite, all'interno della coppia le variabili sono negativamente correlate. Ogni $Z^{(i)}$ ha la stessa distribuzione di Z ma $\text{Cov}(Z^{(2i-1)}, Z^{(2i)}) < 0$ per $i = 1, \dots, N/2$.

Lo stimatore di μ diventa quindi:

$$\hat{\mu}_{AV} = \frac{1}{N} \sum_{i=1}^N \tilde{Z}^{(i)} = \frac{1}{N/2} \sum_{i=1}^{N/2} \frac{Z^{(2i-1)} + Z^{(2i)}}{2}$$

e ha le seguenti proprietà:

- $\mathbb{E}[\hat{\mu}_{AV}] = \mathbb{E}[Z] = \mu$
- $\text{Var}(\hat{\mu}_{AV}) = (\text{Var}(Z) + \text{Cov}(Z^{(1)}, Z^{(2)}))/N < \text{Var}(Z)/N = \text{Var}(\hat{\mu}_{CMC})$

Come generiamo coppie di variabili aleatorie negativamente correlate?

La variabile aleatoria Z è generata come $Z = \Psi(X_1, \dots, X_d)$, a partire da una sequenza di variabili aleatorie X_1, \dots, X_d indipendenti e identicamente distribuite. Se la funzione densità di X_i è simmetrica rispetto alla sua media $\mathbb{E}[X_i]$ e $\Psi : \mathbb{R}^d \rightarrow \mathbb{R}$ è una funzione monotona in ciascuno dei suoi argomenti, allora $\Psi(\mathbf{X})$ e $\Psi(2\mathbb{E}[\mathbf{X}] - \mathbf{X})$ sono negativamente correlate. Questo risultato segue dalla *Chebyshev Covariance Inequality*.

Chebyshev Covariance Inequality.

Sia $X \in \mathbb{R}$ una variabile aleatoria distribuita secondo una funzione densità $f(\cdot)$. Siano $g, h : \mathbb{R} \rightarrow \mathbb{R}$ due funzioni non-decrescenti tali che $\mathbb{E}[g(X)]$, $\mathbb{E}[h(X)]$, $\mathbb{E}[g(X)h(X)] < \infty$.

Allora $\text{Cov}(g(X), h(X)) \geq 0$.

La diseguaglianza si estende anche al caso multivariato, dove si considera $\mathbf{X} \in \mathbb{R}^d$ con una funzione densità congiunta $f(\cdot)$ e $g(\cdot)$, $h(\cdot)$ due funzioni non-negative in ciascun argomento.

Nel contesto di *Antithetic Variables* abbiamo che se $\Psi(\cdot)$ è non-decrescente in ciascuno dei suoi argomenti allora $\Psi(2\mathbb{E}[\mathbf{X}] - \mathbf{X})$ è crescente, e quindi $\text{Cov}(\Psi(\cdot), \Psi(2\mathbb{E}[\mathbf{X}] - \mathbf{X})) \geq 0$. Analogamente se $\Psi(\cdot)$ è non-crescente.

Antithetic Variables.

Sia $Z = \Psi(\mathbf{X})$ un modello stocastico, con $\mathbf{X} = (X_1, \dots, X_d)$ distribuita secondo la funzione densità congiunta $f(\cdot)$ e $\Psi(\cdot)$ una funzione non-decrescente. È possibile stimare $\mu = \mathbb{E}[Z]$ con il metodo delle *Antithetic Variables* come segue:

1. Genera $N/2$ campioni $\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(N/2)}$ indipendenti e identicamente distribuiti secondo la distribuzione di \mathbf{X}
2. Calcola $\hat{\mu}_{AV} = \frac{1}{N/2} \sum_{i=1}^{N/2} \frac{\Psi(\mathbf{X}^{(i)}) + \Psi(2\mathbb{E}[\mathbf{X}^{(i)}] - \mathbf{X}^{(i)})}{2}$

Notiamo, in particolare, che è necessario conoscere $\mathbb{E}[\mathbf{X}]$. Questo non è un problema perché ciò che non conosciamo è $\mathbb{E}[Z]$.

■ Consideriamo $Z \sim \text{Exp}(\lambda)$, per cui $Z = -1/\lambda \log(X) = \Psi(X)$ con $X \sim \mathcal{U}([0, 1])$. In particolare, $\Psi(\cdot)$ è una funzione monotonamente crescente e quindi $\Psi(X)$ e $\Psi(2\mathbb{E}[X] - X) = \Psi(1 - X)$ sono negativamente correlate. Lo stimatore dato dal metodo Monte Carlo con antithetic variables per stimare $\mu = \mathbb{E}[Z]$ è $\hat{\mu}_{AV} = \frac{1}{N/2} \sum_{i=1}^{N/2} \frac{1}{2} (\Psi(X^{(i)}) - \Psi(1 - X^{(i)}))$, con $X^{(i)} \sim \mathcal{U}([0, 1])$.

Importance Sampling

Utile soprattutto per la stima delle probabilità di eventi rari, l'*Importance Sampling* è un metodo che si basa su qualcosa di simile al cambio di variabili nell'integrale. Viene introdotta una nuova funzione densità e questa pesa in modo diverso gli eventi (i punti del dominio su cui è definita).

Sia $\mathbf{X} = (X_1, \dots, X_d)$ un vettore aleatorio con funzione densità congiunta $f : \mathbb{R}^d \rightarrow \mathbb{R}_+$ e sia $Z = \Psi(\mathbf{X})$, con $\Psi : \mathbb{R}^d \rightarrow \mathbb{R}$. L'obiettivo è quello di calcolare il valore atteso di Z , dato da $\mu = \mathbb{E}[Z] = \int_{\mathbb{R}^d} \Psi(\mathbf{x}) f(\mathbf{x}) d\mathbf{x}$. Il metodo Monte Carlo non risulta efficiente se la funzione integranda ha valori alti concentrati in una piccola regione del dominio.

Il metodo *Importance Sampling* introduce una funzione densità ausiliare $g : \mathbb{R}^d \rightarrow \mathbb{R}_+$ tale che $g(\mathbf{x}) = 0$ solo se $\Psi(\mathbf{x}) f(\mathbf{x}) = 0$ e $\int_{\mathbb{R}^d} g(\mathbf{x}) d\mathbf{x} = 1$. L'integrale può quindi essere riscritto come:

$$\mu = \mathbb{E}[Z] = \mathbb{E}_f[\Psi(\mathbf{X})] = \int_{\mathbb{R}^d} \left(\frac{\Psi(\mathbf{x}) f(\mathbf{x})}{g(\mathbf{x})} \right) g(\mathbf{x}) d\mathbf{x} = \mathbb{E}_g \left[\frac{\Psi(\mathbf{X}) f(\mathbf{X})}{g(\mathbf{X})} \right]$$

La densità $f(\cdot)$ viene detta *distribuzione nominale*, la densità $g(\cdot)$ viene detta *importance distribution*. L'obiettivo di $g(\cdot)$ è di rendere più probabili le zone in cui $\Psi(\mathbf{x}) f(\mathbf{x})$ assume valori alti. L'*Importance Sampling* è quindi molto utile quando $\Psi(\mathbf{x})$ assume valori alti per le \mathbf{x} che sono poco probabili secondo $f(\cdot)$.

Fissata la densità $g(\cdot)$ si procede con l'approccio Monte Carlo. Al posto di generare N campioni di $\mathbf{X} \sim f(\cdot)$ e calcolare $\mu = \mathbb{E}[\Psi(\mathbf{X})]$, generiamo N campioni di $\tilde{\mathbf{X}} \sim g(\cdot)$ e calcoliamo $\mu = \mathbb{E}_g[\Psi(\tilde{\mathbf{X}}) f(\tilde{\mathbf{X}})/g(\tilde{\mathbf{X}})]$.

Importance Sampling.

Sia $Z = \Psi(\mathbf{X})$ un modello stocastico, con $\mathbf{X} = (X_1, \dots, X_d)$ distribuita secondo la funzione densità congiunta $f(\cdot)$ e $\Psi(\cdot)$ una funzione di interesse. Introduciamo una funzione densità ausiliare $g(\cdot)$ che favorisce le \mathbf{x} per cui $\Psi(\cdot)$ assume valori alti. È possibile stimare $\mu = \mathbb{E}[Z]$ con il metodo *Importance Sampling* come segue:

1. Genera N campioni $\tilde{\mathbf{X}}^{(1)}, \dots, \tilde{\mathbf{X}}^{(N)}$ indipendenti e identicamente distribuiti secondo $g(\cdot)$
2. Calcola $\hat{\mu}_{IS} = \frac{1}{N} \sum_{i=1}^N \frac{\Psi(\tilde{\mathbf{X}}^{(i)}) f(\tilde{\mathbf{X}}^{(i)})}{g(\tilde{\mathbf{X}}^{(i)})}$
3. Calcola $\hat{\sigma}_{IS}^2 = \frac{1}{N-1} \sum_{i=1}^N \left(\frac{\Psi(\tilde{\mathbf{X}}^{(i)}) f(\tilde{\mathbf{X}}^{(i)})}{g(\tilde{\mathbf{X}}^{(i)})} - \hat{\mu}_{IS} \right)^2$
4. Restituisce $\hat{\mu}_{IS}$ e l'intervallo di confidenza di livello $1 - \alpha$:

$$\hat{I}_\alpha = [\hat{\mu}_{IS} - z_{1-\frac{\alpha}{2}} \frac{\hat{\sigma}_{IS}}{\sqrt{N}}, \hat{\mu}_{IS} + z_{1-\frac{\alpha}{2}} \frac{\hat{\sigma}_{IS}}{\sqrt{N}}]$$

Lo stimatore $\hat{\mu}_{IS}$ è *ubaised* e tale che:

$$\text{Var}(\hat{\mu}_{IS}) = \frac{1}{N} \left(\mathbb{E}_g \left[\left(\frac{\Psi(\mathbf{X}) f(\mathbf{X})}{g(\mathbf{X})} \right)^2 \right] - \mu^2 \right)$$

L'*Importance Sampling* e l'*Acceptance-Rejection Method* sono idee molto simili: entrambi si basano sul campionamento da una distribuzione per ottenere un campione da un'altra.

Optimal Choice of $g(\cdot)$

L'efficacia del metodo è basata sulla scelta di $g(\cdot)$. La scelta ottimale di $g(\cdot)$ è quella che minimizza $\int (\Psi(\mathbf{x}) f(\mathbf{x}))^2 / g(\mathbf{x}) d\mathbf{x}$ rispettando la costri-

zione che $g(\cdot) \geq 0$ e $\int g(\mathbf{x}) d\mathbf{x} = 1$. Sfruttando il metodo di Lagrange si può trovare la soluzione:

$$g^*(\mathbf{x}) = \frac{|\Psi(\mathbf{x})| f(\mathbf{x})}{\mathbb{E}_f[|\Psi(\mathbf{X})|]}$$

Lavorare con $g^*(\cdot)$ non è pratico, siccome la costante di normalizzazione $\mathbb{E}[|\Psi(\mathbf{X})|]$ è, tipicamente, difficile da calcolare tanto quanto è difficile da calcolare $\mu = \mathbb{E}[\Psi(\mathbf{X})]$. Una candidata $g(\cdot)$ è tanto più buona quanto più vicina a $g^*(\cdot)$, cioè $g(\cdot)$ dovrebbe essere simile a $|\Psi(\cdot)| f(\cdot)$ ma anche essere facile da calcolare.

Una delle considerazioni più importanti per scegliere una buona $g(\cdot)$ è che $\hat{\mu}_{IS}$ dovrebbe avere varianza finita. Questo è equivalente a chiedere:

$$\mathbb{E}_g \left[\Psi^2(\mathbf{X}) \frac{f^2(\mathbf{X})}{g^2(\mathbf{X})} \right] = \mathbb{E}_f \left[\Psi^2(\mathbf{X}) \frac{f(\mathbf{X})}{g(\mathbf{X})} \right] < \infty$$

Control Variates

Il metodo delle *Control Variates* applica la versione cruda di Monte Carlo a $Z_\alpha = Z + \alpha(Y - \mathbb{E}[Y])$, piuttosto che alla variabile aleatoria originale Z , dove Y è una variabile aleatoria fortemente correlata con Z e tale che $\mathbb{E}[Y]$ è noto. La variabile aleatoria Z_α è tale che $\mathbb{E}[Z_\alpha] = \mathbb{E}[Z] = \mu$ e $\text{Var}(Z_\alpha) = \text{Var}(Z) + \alpha^2 \text{Var}(Y) + 2\alpha \text{Cov}(Z, Y)$. La varianza di Z_α è una forma quadratica di α ed è quindi minimizzata con:

$$\alpha_{opt} = -\frac{\text{Cov}(Z, Y)}{\text{Var}(Y)}$$

Si ottiene dunque $\text{Var}(Z_\alpha) = \text{Var}(Z)(1 - \text{Corr}(Z, Y)^2) < \text{Var}(Z)$.

La varianza è tanto più ridotta quanto più Z e Y sono correlate. Il caso ideale è $Y = \gamma Z$, $\gamma \in \mathbb{R}$, che porta a $Var(\tilde{Z}_\alpha) = 0$. Tuttavia, in questa situazione si ha $\mathbb{E}[Y] = \gamma \mathbb{E}[Z]$, che è il problema di partenza.

Control Variates.

Sia Z un modello stocastico. È possibile stimare $\mu = \mathbb{E}[Z]$ con il metodo *Control Variates* applicando la versione *cruda* di Monte Carlo alla variabile aleatoria $\tilde{Z}_\alpha = Z + \alpha(Y - \mathbb{E}[Y])$, dove Y è una variabile aleatoria correlata con Z e tale che $\mathbb{E}[Y]$ è conosciuto.

1. Genera \tilde{N} campioni $(Z^{(i)}, Y^{(i)})$ di (Z, Y)
2. Calcola $\hat{\mu}_Z = \frac{1}{\tilde{N}} \sum_{i=1}^{\tilde{N}} Z^{(i)}$
3. Calcola $\hat{\sigma}_{ZY}^2 = \frac{1}{\tilde{N}-1} \sum_{i=1}^{\tilde{N}} (Z^{(i)} - \hat{\mu}_Z)(Y^{(i)} - \mathbb{E}[Y])$
4. Stima $\hat{\alpha}_{opt} = -\hat{\sigma}_{ZY}^2 / \sigma_Y^2$
5. Applica Monte Carlo a $\tilde{Z}_{\alpha_{opt}} = Z + \hat{\alpha}_{opt}(Y - \mathbb{E}[Y])$:
 - Genera N campioni $(\tilde{Z}_{\alpha_{opt}}^{(i)}, Y^{(i)})$ di (Z, Y)
 - Calcola $\hat{\mu}_{CV} = \frac{1}{N} \sum_{i=1}^N (\tilde{Z}_{\alpha_{opt}}^{(i)} + \hat{\alpha}_{opt}(Y^{(i)} - \mathbb{E}[Y]))$

Lo stimatore $\hat{\mu}_{CV}$ è *unbiased* e la sua varianza è data da:

$$Var(\hat{\mu}_{CV}) = \frac{Var(\tilde{Z}_\alpha) + Var(\hat{\alpha}_{opt}) \sigma_Y^2}{N} \approx \hat{\sigma}_Z^2 - \frac{\hat{\sigma}_{ZY}^2}{\sigma_Y^2} := \hat{\sigma}^2(\tilde{Z}_{\alpha_{opt}})$$

L'algoritmo è eseguibile anche in un'altra versione, dove non è necessario campionare \tilde{N} elementi iniziali. Questo ha un costo: lo stimatore $\hat{\mu}_{CV}$ diventa *biased* (stiamo rompendo l'indipendenza tra $\hat{\alpha}_{opt}$ e lo stimatore).

Control Variates - One Shot.

1. Genera N campioni $(Z^{(i)}, Y^{(i)})$ di (Z, Y)
2. Calcola $\hat{\mu}_Z = \frac{1}{N} \sum_{i=1}^N Z^{(i)}$
3. Calcola $\hat{\sigma}_{ZY}^2 = \frac{1}{N-1} \sum_{i=1}^N (Z^{(i)} - \hat{\mu}_Z)(Y^{(i)} - \mathbb{E}[Y])$
4. Stima $\hat{\alpha}_{opt} = -\hat{\sigma}_{ZY}^2 / \sigma_Y^2$
5. Calcola $\hat{\mu}_{CV} = \frac{1}{N} \sum_{i=1}^N (Z^{(i)} + \hat{\alpha}_{opt}(Y^{(i)} - \mathbb{E}[Y]))$

In entrambe le versioni dell'algoritmo otteniamo:

$$\sqrt{N} \frac{\hat{\mu}_{CV} - \mu}{\hat{\sigma}^2(\tilde{Z}_{\alpha_{opt}})} \rightarrow N(0, 1) \text{ in distribuzione}$$

da cui poi è possibile ottenere un intervallo di confidenza asintotico.

Multiple Control Variates

È possibile generalizzare a una versione multidimensionale. Definiamo la variabile aleatoria $\tilde{Z}_\alpha = Z + \sum_{j=1}^p \alpha_j (Y_j - \mathbb{E}[Y_j]) = Z + \alpha(Y - \mathbb{E}[Y])$, dove $Y = (Y_1, \dots, Y_p)$ è un vettore aleatorio multivariato di cui conosciamo $\mathbb{E}[Y]$. Otteniamo dunque:

$$Var(\tilde{Z}_\alpha) = Var(Z) + 2 Cov(Z, Y) \alpha + \alpha^T Cov(Y, Y) \alpha$$

dove $Cov(Z, Y) = (Cov(Z, Y_1), \dots, Cov(Z, Y_p))$ e $Cov(Y, Y) \in \mathbb{R}^{p \times p}$, con $(Cov(Y, Y))_{ij} = Cov(Y_i, Y_j)$. Si tratta di una forma quadratica di α , minimizzata quindi da:

$$\alpha_{opt} = -Cov(Y, Y)^{-1} Cov(Z, Y)$$

Stratification/Stratified Sampling

Latin Hypercube Sampling