

Part I: Simulating Statistical Models

1. Uniform (Pseudo) Random Number Generation

L'obiettivo principale è campionare una sequenza di variabili aleatorie da una data distribuzione. Nella pratica questo è impossibile, il meglio a cui possiamo ambire è di generare una sequenza di valori che è indistinguibile da una sequenza aleatoria (in termini di proprietà statistiche). Ci concentriamo inizialmente sui *Pseudo Random Number Generators* (RNG) per campionare da $\mathcal{U}([0, 1])$. I più famosi RNG sono:

- 1. linear congruential generator (LNG)
- 2. matrix congruential generator (MCG)
- 3. modulo-2 generator
- 4. combined generator

Come verifichiamo che la sequenza di numeri generati sia indistinguibile da una sequenza di numeri random? Attraverso test statisticci. Si può procedere sia teoricamente che empiricamente. Ci concentreremo sui test empirici, ovvero test non-parametrici di goodness-of-fit. L'idea è di confrontare la funzione di ripartizione *reale* $F(\cdot)$ con quella *empirica* $F_n(\cdot)$. I test più usati sono: QQ-plot, Kolmogorov-Smirnov test, χ^2 -test.

2. Random Variable Generation

Sappiamo campionare da $\mathcal{U}([0, 1])$, l'obiettivo è poter campionare da una distribuzione generica (univariata, multivariata, processo gaussiano).

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

Molte distribuzioni sono collegate all'uniforme attraverso semplici trasformazioni. Sfruttando queste relazioni possiamo definire l'*Inverse-Transform Method*: campioniamo una variabile aleatoria $U \sim \mathcal{U}([0, 1])$ e arriviamo a $X \sim$ *distribuzione generica* invertendo $F(\cdot)$, funzione di ripartizione di X . Questo metodo è valido sia nel caso discreto che nel caso continuo, ma solo per variabili aleatorie univariate.

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

Se la variabile aleatoria X viene da una combinazione di funzioni di ripartizione è possibile scrivere $F(x) = \sum_{i=1}^n p_i F_i(x)$, dove $\{p_1, \dots, p_n\}$ è una densità di probabilità su $\{F_1, \dots, F_n\}$. L'idea del *Composition Method* è di decidere da quale $F_i(\cdot)$ campionare in base alle probabilità p_i e poi campionare con un metodo già noto. E.g. Supponiamo di voler campionare $X \sim \alpha_1 N(\mu_1, \sigma_1^2) + \alpha_2 N(\mu_2, \sigma_2^2)$. Con probabilità α_i campioneremo da $N(\mu_i, \sigma_i^2)$. Generiamo la variabile aleatoria discreta Y con distribuzione $\mathbb{P}(Y = i) = \alpha_i$ e poi campioniamo $X \sim N(\mu_Y, \sigma_Y^2)$.

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

Sia $X \sim f(\cdot)$ e supponiamo di non essere in grado di campionare direttamente da $f(\cdot)$ (difficile di invertire o conosciuta a meno della costante di normalizzazione). L'*Acceptance-Rejection Method* sfrutta l'uso di una distribuzione ausiliare $g(\cdot)$ facile da campionare e tale che $f(x) \leq Cg(x)$ per qualsiasi x . Per ottenere un campione da $f(\cdot)$: generiamo $Y \sim g(\cdot)$ e in contemporanea (ma indipendentemente) generiamo $U \sim \mathcal{U}([0, 1])$, accettiamo Y come campione di $f(\cdot)$ solo se $U \leq f(Y)/Cg(Y)$. I campioni accettati saranno distribuiti secondo $f(\cdot)$. Il metodo risulta tanto più efficiente quanto più $f(\cdot)$ e $Cg(\cdot)$ sono vicine.

■ Transformation of Random Variables

Un altro metodo efficiente per il campionamento si basa sul trasformare le densità di probabilità, cioè trasformare variabili aleatorie. Si può ottenere un campione da $f(\cdot)$ partendo da $\tilde{f}(\cdot)$ appartenente alla stessa *location-scale family*. Lo schema generico è $f(x; \mu, \sigma) = \frac{1}{\sigma} \tilde{f}\left(\frac{x-\mu}{\sigma}\right)$.

→ Possiamo sfruttare questo metodo per campionare da $N(\mu, \sigma^2)$ partendo da $N(0, 1)$. Come campioniamo da $N(0, 1)$? Attraverso il metodo di *acceptance-rejection* usando $Exp(1)$ come densità ausiliare oppure con la *trasformazione di Box-Muller*.

■ Multivariate Random Variable Generation

Nel caso multivariato l'obiettivo è ottenere un campione $\mathbf{X} = (X_1, \dots, X_n)$ distribuito secondo la funzione di ripartizione congiunta $F(\cdot)$. Limitiamo i casi di studio a poche, specifiche, situazioni.

• Independent Components

Se le componenti X_i sono indipendenti tra loro è sufficiente generare ciascuna X_i indipendentemente usando uno dei metodi noti.

• Conditional Distributions

Se conosciamo le distribuzioni di $X_i | X_{i-1}, \dots, X_1$, per ottenere \mathbf{X} è sufficiente generare X_1 e a ogni iterazione generare X_i dalla sua distribuzione condizionata.

• Generation by Transformation: Copulas

Supponiamo che le componenti di \mathbf{X} siano dipendenti e con distribuzioni marginali $F_i(\cdot)$. Definiamo una *copula* per descrivere la relazione di dipendenza tra le variabili. Una copula è una funzione di ripartizione di n variabili aleatorie uniformemente distribuite e dipendenti tra loro. Per generare \mathbf{X} campioniamo $U \sim$ *copula* e fissiamo $\mathbf{X} = (F_1^{-1}(U_1), \dots, F_n^{-1}(U_n))$. In questo modo, la dipendenza delle variabili X_i è nascosta nella generazione di U .

• Multivariate Gaussian

Per campionare $\mathbf{X} \sim N(\mu, \Sigma)$ è necessario fattorizzare la matrice covarianza Σ con la *decomposizione spettrale* o la *fattorizzazione di Choleski*, ottenendo $\Sigma = A A^T$. Generiamo quindi $\mathbf{Y} \sim N(\mathbf{0}, \mathbf{I})$ (generiamo ogni componente $Y_i \sim N(0, 1)$) e otteniamo \mathbf{X} dalla trasformazione $\mathbf{X} = \mu + A \mathbf{Y}$.

• Conditional Multivariate Gaussian

Supponiamo di voler campionare $\mathbf{X} \sim N(\mu, \Sigma)$, dove solo le componenti $(X_{k+1}, \dots, X_n) = \mathbf{Z}$ sono osservabili. L'idea è di generare le rimanenti variabili $(X_1, \dots, X_k) = \mathbf{Y}$ attraverso la distribuzione condizionata $\mathbf{Y} | \mathbf{Z} = \mathbf{z}$ (che sappiamo essere gaussiana).

■ Gaussian Process (GP) Generation

I *Gaussian Processes* (GP) possono essere visti come una generalizzazione dei vettori gaussiani. Una collezione di variabili aleatorie $\{X_t\}_t$ è detta *processo stocastico*, se $X_t \in \mathbb{R}$, o *random field*, se $\mathbf{X}_t \in \mathbb{R}^d$. Un processo gaussiano è un processo stocastico/random field per cui ogni sottoinsieme finito $(X_{t_1}, \dots, X_{t_n})$ si comporta come un vettore aleatorio gaussiano. I processi gaussiani sono interamente caratterizzati dalle funzioni media e covarianza. Possiamo campionare $\mathbf{X} \sim GP(\mu_X(\cdot), C_X)$ nei punti t_1, \dots, t_n definendo $\mu = (\mu_X(X_{t_1}), \dots, \mu_X(X_{t_n}))$, $\Sigma_{ij} = C_X(t_i, t_j)$ e campionando (con un metodo già noto) $\mathbf{X} \sim N(\mu, \Sigma)$.

3. Random Inputs Parametrization

Per simulare un sistema stocastico è necessario caratterizzare adeguatamente gli input aleatori, i.e. parametrizzare il sistema. Nel caso semplice, il sistema dipende da un numero finito di parametri, $\mathbf{Y} = (Y_1, \dots, Y_n)$, e l'obiettivo è trovare un insieme finito di variabili aleatorie $\mathbf{Z} = (Z_1, \dots, Z_p)$ e una funzione $T(\cdot)$ tali che $\mathbf{Y} = T(\mathbf{Z})$. Nei casi più complessi, invece, il sistema dipende da un numero infinito di variabili aleatorie, ad esempio da un processo stocastico/random field. Abbiamo quindi bisogno di una tecnica per ridurre la dimensione di un oggetto infinito-dimensionale. L'idea è di rappresentare il processo stocastico/random field $a(x, w)$ attraverso una serie costruita sulla sua media e varianza (con il *teorema di Mercer* otteniamo una decomposizione di $a(x, w)$, da cui definiamo la *Karhunen-Loeve Expansion* $a(x, w) = \mathbb{E}[a](x) + \sum_{i=1}^{\infty} \sqrt{\lambda_i} b_i(x) Y_i(w)$). Troncando la serie dopo N elementi otteniamo un'approssimazione *finita* del processo stocastico/random field $a(x, w)$. Un esempio molto semplice è il *gaussian random field* (versione multidimensionale del GP).

4. Monte Carlo Methods

Se campioniamo un gran numero di volte un sistema stocastico, i campioni rifletteranno il comportamento statistico del modello. Molti problemi sono riducibili al calcolo del valore atteso di una variabile aleatoria. Pertanto, l'obiettivo è calcolare $\mathbb{E}[\Psi(\mathbf{X})]$, dove $\mathbf{X} \sim f(\cdot)$ è una variabile aleatoria (o vettore aleatorio \mathbf{X}) e $\Psi(\cdot)$ è una funzione di interesse. Il metodo *Monte Carlo* restituisce l'approssimazione $\mathbb{E}[\Psi(\mathbf{X})] \approx \frac{1}{N} \sum_{i=1}^N \Psi(\mathbf{X}^{(i)})$ a partire da N campioni $\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(N)}$ iid secondo la distribuzione di \mathbf{X} . Il metodo Monte Carlo si basa sulla *Legge (Forte) dei Grandi Numeri*. In aggiunta, basandosi sul *Teorema Centrale del Limite* è possibile generare un intervallo di confidenza asintotico per $\mathbb{E}[\Psi(\mathbf{X})]$. Questo è fondamentale siccome l'output di una simulazione Monte Carlo dovrebbe sempre avere, oltre ad una stima puntuale, una stima dell'errore.

■ Monte Carlo to compute Integrals

Sia $Z = \Psi(\mathbf{X})$, dove $\mathbf{X} = (X_1, \dots, X_n)$ è un vettore aleatorio distribuito secondo la densità $f(\cdot)$ e $\Psi(\cdot)$ è una funzione di interesse. L'obiettivo è calcolare $\mu = \mathbb{E}[Z] = \int_{\mathbb{R}^n} \Psi(\mathbf{x}) f(\mathbf{x}) d\mathbf{x}$. A partire da $\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(N)}$ campioni iid di \mathbf{X} , il metodo Monte Carlo produce $\hat{\mu} = \frac{1}{N} \sum_{i=1}^N \Psi(\mathbf{X}^{(i)})$.

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 (con nodi $\mathbf{X}^{(i)}$ e pesi $1/N$ da assegnare a ciascun nodo). Si può ragionare in modo inverso e sfruttare il metodo Monte Carlo per approssimare un generico integrale, senza doverlo inserire in un contesto statistico. Supponiamo di voler calcolare $I = \int_{\mathbb{R}^n} \Psi(\mathbf{x}) w(\mathbf{x}) d\mathbf{x}$, con l'integrandi $\Psi(\cdot)$ e la funzione di pesi $w(\cdot)$. A partire da $\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(N)} \sim w(\cdot)$ iid, il metodo Monte Carlo produce la stima $\hat{I} = \frac{1}{N} \sum_{i=1}^N \Psi(\mathbf{X}^{(i)})$.

■ (Smooth) Functions of Expectations

Sia $\mathbf{Z} = (Z_1, \dots, Z_m)$ l'output di un sistema stocastico. L'obiettivo è calcolare una funzione dei valori attesi di \mathbf{Z} , i.e. $\zeta = f(\mathbb{E}[Z_1], \dots, \mathbb{E}[Z_m])$, con $f(\cdot)$ funzione regolare (dobbiamo essere in grado di calcolarne le derivate).

È possibile stimare ζ con il metodo Monte Carlo: è sufficiente campionare $\mathbf{Z}^{(1)}, \dots, \mathbf{Z}^{(N)}$ iid secondo la distribuzione di \mathbf{Z} e fissare $\hat{\zeta} = f(\hat{\mu}_1, \dots, \hat{\mu}_m)$, con $\hat{\mu}_j = \frac{1}{N} \sum_{i=1}^N Z_j^{(i)}$. Oltre allo stimatore puntuale è necessario fornire una stima dell'errore. Questo è possibile con il *Delta Method*, una seconda versione del teorema centrale del limite basata sull'espansione di Taylor di primo ordine centrata in $\mu = \mathbb{E}[Z]$.

5. Variance Reduction Techniques

Sia $Z = \Psi(\mathbf{X})$, dove $\mathbf{X} = (X_1, \dots, X_d)$ è un vettore aleatorio distribuito secondo la densità $f(\cdot)$ e $\Psi(\cdot)$ è una funzione di interesse. L'obiettivo è di calcolare $\mu = \mathbb{E}[Z]$. La versione *cruda* del metodo Monte Carlo produce lo stimatore $\hat{\mu}_{CMC}$ con un errore:

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

Non è possibile ridurre $1/\sqrt{N}$, che rappresenta la velocità di convergenza. Per ridurre l'errore è quindi necessario lavorare su $\sqrt{Var(Z)}$. Le *Variance Reduction Techniques* prevedono di applicare il metodo Monte Carlo su una versione modificata di Z , diciamo \tilde{Z} , tale che $\mathbb{E}[\tilde{Z}] = \mathbb{E}[Z] = \mu$ e $Var(\tilde{Z}) \ll Var(Z)$. Come definiamo \tilde{Z} ?

- **Antithetic Variables**

Al posto di generare N campioni *iid*, l'idea è di generare $N/2$ coppie $(Z^{(2i-1)}, Z^{(2i)})$ di variabili aleatorie negativamente correlate. Ogni $Z^{(i)}$ ha la stessa distribuzione di Z e $Cov(Z^{(2i-1)}, Z^{(2i)}) < 0$. Come generiamo coppie di variabili aleatorie negativamente correlate? Come conseguenza della *Chebyshev Covariance Inequality*, se $f(\cdot)$ e $\Psi(\cdot)$ sono non-decrescenti e $f(\cdot)$ è simmetrica rispetto alla sua media allora $\Psi(\mathbf{X})$ e $\Psi(2\mathbb{E}[\mathbf{X}] - \mathbf{X})$ sono negativamente correlate. Il risultato è adattabile al caso in cui $\Psi(\cdot)$ è monotona. Lo stimatore $\hat{\mu}_{AV}$ è tale che $Var(\hat{\mu}_{AV}) = (Var(Z) + Cov(Z^{(1)}, Z^{(2)}))/N$, per cui concludiamo che $Var(\hat{\mu}_{AV})$ è tanto più piccola di $Var(\hat{\mu}_{CMC})$ quanto più le variabili aleatorie sono negativamente correlate. Cool, ma ipotesi troppo stringenti ($\Psi(\cdot)$ deve essere monotona).

- **Importance Sampling**

Utile soprattutto per la stima delle probabilità di eventi rari, l'*Importance Sampling* è un metodo che si basa su una sorta di cambio di variabili nell'integrale che definisce $\mathbb{E}[Z]$. L'idea è di introdurre una distribuzione auxiliare $g(\cdot)$, trasformare $\mathbb{E}_f[Z] = \mathbb{E}_f[\Psi(\mathbf{X})]$ in $\mathbb{E}_g[\Psi(\mathbf{X}) f(\mathbf{X})/g(\mathbf{X})]$ e procedere con Monte Carlo campionando da $g(\cdot)$. L'obiettivo di $g(\cdot)$ è di rendere più probabili le \mathbf{x} che sono poco probabili secondo $f(\cdot)$ ma per cui $\Psi(\cdot)$ assume valori alti. Per trovare la densità ottimale $g^*(\cdot)$ risolviamo un problema di ottimizzazione e concludiamo che il metodo è tanto più efficiente quanto più $g(\cdot)$ è simile a $|\Psi(\cdot)|f(\cdot)$.

- **Control Variates**

L'idea è applicare Monte Carlo a $\tilde{Z}_\alpha = Z + \alpha(Y - \mathbb{E}[Y])$, dove Y è una variabile aleatoria correlata con Z e tale che $\mathbb{E}[Y]$ è nota. Il valore ottimale di α si ottiene minimizzando la varianza di \tilde{Z}_α ed è dato da $\alpha_{opt} = -Cov(Z, Y)/Var(Y)$. Ne segue che $Var(\tilde{Z}_{\alpha_{opt}})$ è tanto più piccola di $Var(Z)$ quanto più Z e Y sono correlate. Il caso estremo (in cui la varianza si azzera) è dato da $Y = \gamma Z$, dove però $\mathbb{E}[Y] = \gamma \mathbb{E}[Z]$ (non conosciamo $\mathbb{E}[Z]$).

È possibile generalizzare al caso multidimensionale definendo $\tilde{Z}_\alpha = Z + \alpha(\mathbf{Y} - \mathbb{E}[\mathbf{Y}])$, con \mathbf{Y} vettore aleatorio di cui conosciamo $\mathbb{E}[\mathbf{Y}]$ e tale che ogni Y_i è correlata con Z .

- **Stratification/Stratified Sampling**

Un altro modo per stimare $\mathbb{E}[Z]$ è introdurre una partizione $\{\Omega_j\}_j$ del dominio Ω e lavorare su ogni strata Ω_j . In ogni Ω_j generiamo N_j campioni *iid* $Z^{(i)}$ e applichiamo Monte Carlo per ottenere lo stimatore $\hat{\mu}_j$. Lo stimatore finale $\hat{\mu}_{str}$ sarà dato dalla somma pesata di $\{\hat{\mu}_j\}_j$. Il peso associato a $\hat{\mu}_j$ è basato sulla probabilità p_j di finire in Ω_j . È possibile stratificare il campionamento di un'uniforme (immediato) o di qualsiasi altra distribuzione (passando per la funzione di ripartizione). Il vantaggio di questo approccio è la garanzia di avere N_j elementi nello strata Ω_j . Come determiniamo N_j ? Due possibili metodi sono proportional allocation e optimal allocation. In entrambi i casi si arriva a una riduzione della varianza. Un aspetto negativo è che il metodo funziona bene solo se la dimensione dello spazio è ridotta (*curse of dimensionality*).

- **Latin Hypercube Sampling**

Vogliamo stimare $\mathbb{E}[Z] = \mathbb{E}[\Psi(\mathbf{X})]$, con $\mathbf{X} = (X_1, \dots, X_d)$. Se usassimo il metodo della stratificazione dovremmo stratificare ciascuna variabile X_j in s strata, arrivando così a un totale di s^d strata. Non è computazionalmente ammissibile per d grande. Il *Latin Hypercube Sampling* risolve i problemi legati alla dimensione dello spazio rinunciando all'indipendenza dei campioni. In particolare, *LHS* stratifica ciascuna variabile X_j ma non tutto il dominio. Vengono generati N campioni $\mathbf{X}^{(i)}$ correlati tra loro, tali che ciascuna componente X_j è stratificata in N strata e vi è un campione per ciascuno strata.

Il metodo *LHS* campiona da $\mathcal{U}([0, 1]^d)$ (da cui è possibile adattarsi ad altre distribuzioni) basandosi su semplici permutazioni e genera lo stimatore $\hat{\mu}_{LHS}$. La varianza dello stimatore è tanto più ridotta (e quindi il metodo risulta tanto più efficiente) quanto più la struttura di $\Psi(\cdot)$ si avvicina a una struttura additiva.

Siccome i campioni non sono *iid*, per sfruttare il teorema centrale del limite e fornire una stima dell'errore ci occorre creare lo stimatore $\hat{\mu}_{LHS}$ (media di K ripetizioni indipendenti di $\hat{\mu}_{LHS}$).

6. Quasi Monte Carlo Methods

Sia $Z = \Psi(\mathbf{X})$, dove $\mathbf{X} = (X_1, \dots, X_d) \sim \mathcal{U}([0, 1]^d)$ e $\Psi(\cdot)$ è una funzione di interesse. L'obiettivo è quello di calcolare $\mu = \mathbb{E}[Z]$. A partire dalla stima dell'errore del metodo Monte Carlo, ci siamo concentrati sulla riduzione di $\sqrt{Var(Z)}$. È possibile, invece, ridurre \sqrt{N} ? Monte Carlo non è in grado. Introduciamo dunque i *Quasi Monte Carlo Methods*, una classe di metodi che preserva la struttura dello stimatore Monte Carlo (media dei campioni) partendo però da un campionamento puramente deterministico, non più *random*. I metodi Quasi Monte Carlo scelgono come punti (campioni) gli elementi di sequenze a bassa discrepanza. Una sequenza di punti è detta a bassa discrepanza se può essere usata per approssimare efficientemente un dato volume. In particolare, una sequenza a bassa discrepanza sembra più uniformemente distribuita di N campioni casuali di un'uniforme.

- **Koksma-Hlawka Identity**

Come mai è importante parlare di *discrepancy*? Partendo dall'identità di Zaremba è possibile dedurre l'identità di Koksma-Hlawka, una relazione che limita l'errore dei metodi Quasi Monte Carlo con il prodotto di due termini: uno dipendente dall'integrandi $\Psi(\cdot)$ e l'altro dipendente da una misura della discrepanza dei punti (ci sono diverse misure di discrepanza). Dall'identità di KH capiamo che, siccome l'integrandi è fissa, la velocità di convergenza sarà tanto più buona quanto più sarà bassa la discrepanza dei punti.

- **Convergence Rate**

Esistono vari costruttori di sequenze a bassa discrepanza. In base alla loro costruzione, è possibile distinguere le sequenze annidate (sequenze) e quelle non annidate (point set). I migliori costruttori di point set (o sequenze) a bassa discrepanza raggiungono una velocità di convergenza dell'ordine di $(\log(N))^{d-1}/N$ (o $(\log(N))^d/N$). Da qui deduciamo che un drawback di questo metodo è che la velocità di convergenza dipende dalla dimensione dello spazio in cui ci troviamo (nel caso di Monte Carlo, invece, si ha una velocità di convergenza non ottima, ma indipendente da d).

- **Randomized QMC**

A partire da una sequenza a bassa discrepanza, i metodi Quasi Monte Carlo generano lo stimatore $\hat{\mu}_{QMC}$. Tuttavia, siccome i punti sono deterministici e non *iid*, per sfruttare il teorema centrale del limite e fornire una stima dell'errore ci occorre randomizzare l'algoritmo. Una versione randomizzata del metodo Quasi Monte Carlo è la *Randomly Shifted QMC*: è sufficiente generare K campioni da un'uniforme d -dimensionale, usarli per distorcere K volte la sequenza a bassa discrepanza, generare K stimatori indipendenti e restituire lo stimatore finale $\hat{\mu}_{QMC}$.

Part II: Forward Uncertainty Quantification and Sensitivity Analysis

7. Sensitivity Analysis

La *Sensitivity Analysis* quantifica gli effetti delle variazioni degli input sugli output, creando un criterio con cui poter classificare gli input dal più influente al meno influente. In particolare, la SA è lo studio di come l'incertezza dell'output può essere attribuita a diverse fonti di incertezza degli input. Dato un modello, i *fattori in input* possono essere suddivisi in quattro gruppi: equazioni usate nel modello, iperparametri (*time step, mesh size*), parametri e dati. Ci concentreremo sugli input dati dai parametri (e qualche volta anche sui dati).

La *Sensitivity Analysis* investiga l'importanza dei fattori input e può essere suddivisa in cinque *task*: *factor prioritization* (identificazione degli input più influenti), *factor fixing* (assegnazione di valori non aleatori agli input meno influenti), *factor mapping* (mappare il comportamento dell'output con una funzione definita su uno specifico dominio degli input), *variance cutting* (riduzione della varianza dell'output sotto a una certa tolleranza), calibrazione del modello sfruttando informazioni aggiunte.

La *Sensitivity Analysis* è suddivisa (principalmente) in due gruppi: metodi *locali*, che studiano gli effetti di piccole perturbazioni dell'input, e metodi *globali*, che invece prevedono di considerare l'intero spazio degli input.

■ Local Methods

Consideriamo un modello della forma $Y = f(\mathbf{X})$, dove $\mathbf{X} = (X_1, \dots, X_k)$ è un vettore aleatorio.

1. Partial Derivatives

I *Local Methods* sono basati sulla valutazione delle derivate parziali dell'output Y rispetto a ogni variabile input X_i . Questo può diventare complicato nei casi in cui il modello che genera l'output è complesso, se la dimensione dello spazio degli input è grande, oppure se diversi input interagiscono tra di loro. In particolare, eseguendo un metodo locale rende difficile esplorare tutto lo spazio degli input, cosa che invece vorremmo fare. Lasciando inesplorate zone dello spazio, rischiamo di perdere informazioni essenziali. È possibile mostrare (con un controesempio) quanto sia limitato il metodo delle derivate parziali come indicatori di sensitività. Una possibile modifica del metodo è considerare come indicatore di sensitività di X_i la derivata parziale di Y rispetto a X_i moltiplicata per il rapporto tra la deviazione standard della variabile X_i e la deviazione standard di Y , ottenendo la sigma-normalized output derivative.

2. Scatterplots and Linear Regression

Un metodo semplice per capire in che relazione è Y con ciascuna delle sue variabili input X_i è lo scatterplot. Per riassumere i risultati di uno scatterplot possiamo applicare una regressione. Se supponiamo che una regressione lineare possa essere una buona struttura (quindi assumiamo linearità e normalità) allora gli Standardized Regression Coefficients (SRC) diventano una misura per la sensitività dell'output a ciascun input. In particolare, i SRC² rappresentano il contributo frazionario alla varianza dell'output.

Per capire l'importanza delle SRC, introduciamo nello stesso framework un'altra misura di sensitività: il coefficiente di correlazione R^2 . Se gli input sono indipendenti tra di loro allora R^2 è semplicemente la somma degli SRC². Se, in aggiunta, il modello è lineare, le SRC risultano essere le sigma-normalized derivatives (se il modello non è lineare le SRC sono misure di sensitività più robuste).

3. Screening Methods

Il problema delle derivate parziali è l'elevato costo che richiedono e calcolarle per un insieme fitto di punti diventa un problema. Se però, al posto delle derivate parziali, in ciascun punto analizziamo il rapporto incrementale otteniamo un metodo efficiente, il *Screening Method*. L'idea è di performare una variazione one-at-time dei parametri input (tutti i parametri meno uno sono fissi). Ci sono diversi modi di procedere, uno di questi è l'*Elementary Effects* (EE) o *Morris Method*. L'EE è molto efficiente per identificare gli inputs non influenti anche nei casi in cui il modello matematico è computazionalmente costoso o nel caso in cui si hanno molti inputs.

Il metodo EE prevede di discretizzare l'input space per ciascuna variabile e poi di performare un certo numero di *one-at-time design experiments* per calcolare l'*elementary effect* EE_j per ciascuna variabile X_j . Negli experiments, i punti e le direzioni di variazione sono scelti casualmente. Il procedimento viene ripetuto diverse volte, portando a μ_j^* (che rappresenta l'importanza complessiva della variabile X_j) e σ_j (che rappresenta una misura della non-linearietà/degli effetti di interazione tra X_j e le altre variabili).

Il metodo è buono, tuttavia è soggetto al *curse of dimensionality*. Questo succede perché, al crescere della dimensione dello spazio degli input diventa poco esplorativo, rischiando di portare a conclusioni sbagliate sull'importanza delle variabili input.

■ Global Methods (Variance-Based Methods)

L'approccio alternativo per la *Sensitivity Analysis* sono i *Global Methods*, ovvero *Variance-Based Methods*. Vengono detti globali perché considerano tutto il range di variazione per ciascun input. È risultato interessante l'approccio della regressione lineare, tuttavia, è possibile introdurre degli indicatori di sensitività che spiegano la totale varianza dell'output piuttosto che spiegare solo la frazione di varianza associata al modello lineare surrogato? Si, l'idea è di introdurre gli *indici di Sobol*, indicatori che spiegano la varianza in termini di *First Order Effects* e *Total Effects*.

• First Order Effects

Cosa succede alla varianza di Y se teniamo fisso il fattore X_i ? Calcoliamo la varianza di Y condizionata a $X_i = x_i$ per un certo valore di x_i , $Var(Y|X_i = x_i)$. Questa è sicuramente minore di $Var(Y)$, in particolare, è tanto più piccola di $Var(Y)$ quanto più X_i ha influenza sulla varianza totale di Y . Però, per far sì che questo indicatore non dipenda da una scelta (s)fortunata di x_i , è necessario fare una media su tutti i possibili valori di x_i . Partendo da questo valore è possibile definire il *First Order Effect* di X_i su Y e il *First Order Sensitivity Index* S_i dell'input X_i su Y . Più S_i è alto e più X_i è importante (se S_i è alto significa che è tanta la varianza dell'output che possiamo spiegare con X_i).

Questo approccio è ottimo per il *factor prioritization* (è sufficiente ordinare per importanza le variabili secondo S_1, \dots, S_k) e per il *factor fixing* (decidiamo di fissare tutte le variabili per cui $S_i < \epsilon$).

Nel caso degli *additive models* (dove l'effetto totale degli input può essere espresso come la somma dei singoli effetti di ciascun input) la somma degli S_i è pari a 1. Nel caso di modelli generici, invece, la somma degli S_i è minore di 1. Questo perché nei *non-additive models* vi è delle varianza spiegata dall'interazione degli input: mancano le *higher order interactions* S_{ij}, S_{ijk}, \dots . Se consideriamo anche le interazioni degli input allora arriviamo a giustificare il 100% della varianza di Y anche nel caso di *non-additive models*. Il problema è che, se consideriamo tutte le possibili interazioni di k input, arriviamo a dover studiare $2^k - 1$ termini. Troppi.

• Total Effects

Per evitare di calcolare tutte le interazioni, introduciamo l'indicatore *Total Effects/Total Sensitivity Index* S_{T_i} di un input X_i sull'output Y . S_{T_i} rappresenta una misura del contributo complessivo di X_i sull'output Y , tenendo conto di tutte le possibili interazioni di X_i . Maggiore è S_{T_i} e maggiore è l'influenza di X_i sull'output.

Questo indicatore è ottimo nel caso di *factor fixing*. Una feature interessante è che questo metodo permette di calcolare il *total effect* di un input X_i o di un set di inputs $\{X_i, \dots, X_j\}$.

Come mai questa decomposizione ha senso? Se supponiamo che \mathbf{X} è distribuita in modo uniforme con le variabili X_i indipendenti allora $f(\cdot)$ può essere decomposta secondo la *High-Dimensional Model Representation* (HDMR), ovvero una *functional analysis of variance* (FANOVA). Si tratta di una decomposizione finita, ma non unica. Sotto ipotesi aggiuntive si può arrivare a una decomposizione unica di $f(\cdot)$, con tutti i termini ortogonali tra di loro. Da qui poi segue la decomposizione della varianza di Y (che giustifica gli indici di Sobol).

Per stimare questi indici dobbiamo stimare molte varianze e valori attesi. L'idea sarebbe quella di usare Monte Carlo, tuttavia risulta troppo computazionalmente costosa. Sfruttiamo, come alternativa, i metodi Quasi Monte Carlo.

È possibile estendere la *Variance-Based Sensitivity Analysis* anche al caso dinamico (output dipendente dal tempo t): è sufficiente introdurre la versione generalizzata degli indici di Sobol, che tiene conto dell'intervallo di tempo considerato. Gli indici generalizzati misurano l'importanza dell'input tenendo conto della loro storia passata. Dunque, è possibile ottenere l'evoluzione dell'importanza degli input.

■ Moment-Independent Importance Measures

I metodi Quasi Monte Carlo sono molto più efficienti dei metodi Monte Carlo, soprattutto se usati insieme alle *Sobol sequences*. Tuttavia, calcolare così tanti valori attesi rimane dispendioso. I *Moment-Independent Importance Measures* sono delle misure di sensitività che considerano direttamente la distribuzione dell'output, senza considerare particolari momenti (come la media o la varianza).

Due diversi *moment-independent methods* sono:

• δ-Sensitivity Measure (basato sulla densità)

Fissiamo la densità (incondizionata) dell'output Y , dopodiché riceviamo informazioni sulla variabile aleatoria X_i e decidiamo di fissarla a un valore x_i . Calcoliamo quanto è diversa la densità di $Y|X_i = x_i$ rispetto alla densità di Y per ogni possibile x_i e facciamo una media, ottenendo il valore δ_i . Le δ_i rappresentano la separazione media tra la densità condizionata e incondizionata di Y e vengono definite *δ-Sensitivity Measures*. Più alta è δ_i e più è alta la sensitività di Y rispetto a X_i .

• PAWN Sensitivity Measure (basato sulla funzione di ripartizione)

L'idea è di confrontare la funzione di ripartizione di Y e la funzione di ripartizione di $Y|X_i = x_i$. Calcoliamo una statistica T_i della distanza tra $F_Y(\cdot)$ e $F_{Y|X_i=x_i}(\cdot)$ (ad esempio calcoliamo la distanza

per ogni x_i e facciamo una media, oppure calcoliamo la distanza per ogni x_i e consideriamo la distanza massima). È possibile considerare diverse distanze tra le due funzioni di ripartizioni (la più usata è la distanza di Kolmogorov-Smirnov) e anche diverse statistiche (massimo, media). In ogni caso, X_i ha tanta più influenza su Y quanto più T_i è alta. Questo approccio può essere usato per il *factor prioritization* e per il *factor fixing*. In particolare, nel caso di *factor fixing* è possibile fare, ad esempio, un *KS test a due variabili*, dove $H_0 : F_Y = F_{Y|X_i}$. Un vantaggio di questo metodo è che può essere usato per qualsiasi input, anche per le *time series* (questo è l'unico *global SA* che possiamo usare per le *time series*).

■ Surrogate Models/Emulators

Abbiamo visto che gli indicatori globali di sensitività, come gli indici di Sobol, sono efficaci per individuare gli input per i quali l'output risulta maggiormente sensibile. Tuttavia, sono troppo dispendiosi computazionalmente. Dunque, questo è un ottimo scenario per l'uso dei modelli surrogati/emulatori. Un modello surrogato è un'approssimazione del modello originale, una semplificazione su cui è più facile lavorare. I due più famosi metodi con cui possiamo ricavare dei surrogati sono *Polynomial Chaos Expansion* (PCE), da cui possiamo ricavare gli indici di Sobol analiticamente a partire dai coefficienti, e i *Gaussian Process* (GP), da cui possiamo ricavare facilmente intervalli di confidenza per gli indici di sensitività.

• Polynomial Chaos Expansion

Consideriamo il vero modello $Y = G(\mathbf{X})$, dove $\mathbf{X} \sim f_{\mathbf{X}}(\cdot)$. Assumendo che Y ha varianza finita, è possibile usare la decomposizione spettrale $Y = \sum_{j=0}^{\infty} y_j Z_j$, dove $\{Z_j\}_j$ è un insieme numerabile di polinomi ortonormali (rispetto alla densità di $f_{\mathbf{X}}(\cdot)$) che genera una base tale che $Z_j = \phi_j(\mathbf{X})$ e $\{y_j\}_j$ sono i coefficienti (coordinate di Y nella nuova base). Troncando la serie dopo N termini, otteniamo una rappresentazione finito-dimensionale di qualsiasi funzione $G(\mathbf{X})$. Notiamo però che si tratta di un procedimento dispendioso, siccome per definire le basi è necessario calcolare integrali. Inoltre, un altro svantaggio è che è necessario conoscere la forma analitica di $G(\cdot)$. Una volta determinate le basi e deciso a quale valore di N vogliamo troncare la serie, è possibile applicare un *least-square minimization approach* per trovare i coefficienti $\{y_j\}_j$. La versione troncata della *PC Expansion* contiene tutte le informazioni sulle proprietà statistiche della variabile aleatoria $Y = G(\mathbf{X})$. Inoltre, siccome entrambe la *PC Expansion* e la *HDMR Sobol decomposition* sono somme di funzioni ortogonali, concludiamo che gli indici di Sobol di qualsiasi ordine possono essere ottenuti dalla combinazione dei coefficienti della *PC Expansion* (al quadrato).

• Gaussian Process Regression (Kriging)

La *GP Regression* si basa sulla distribuzione condizionale gaussiana multivariata. L'idea è di considerare che la *prior knowledge* del modello $G(\mathbf{X})$ può essere modellata con un processo gaussiano (*gaussian random field*) $Z(\mathbf{X})$ con media $\mu_Z(\cdot)$ e funzione covarianza (o *kernel*) C_Z . In questo framework, consideriamo che la vera risposta sia una realizzazione di $Z(\mathbf{X})$. Partendo dalla *prior distribution*, l'obiettivo è di determinare la *predictive distribution* per qualsiasi possibile input.

Definiamo un training set \mathcal{X} e le sue corrispondenti risposte $\mathcal{Y} = G(\mathcal{X})$. L'obiettivo è di performare una predizione *out-of-sample*, cioè di determinare $Z(\mathcal{X}_{new})$. Per farlo, aggiorniamo la *prior* $Z(\cdot) \sim GP(\mu_Z(\cdot), C_Z)$ incorporando la conoscenza aquisita dal fatto che modelliamo $\mathcal{Y} = Z(\mathcal{X})$, ottenendo una nuova funzione media e varianza (che definiscono la *posterior*). La nuova funzione media è una combinazione lineare delle osservazioni \mathcal{Y} più la media *prior*. Partendo dalla *posterior*, è possibile fare predizione su qualsiasi input. Notiamo, in particolare, che la *posterior* definisce una distribuzione degli output. Da qui, oltre ad una stima puntuale, è possibile ottenere degli intervalli di confidenza e stime dell'errore.

Nel caso in cui sia presente del *gaussian noise*, è possibile modificare il modello per renderlo più *loose* nei punti del *training set*. In particolare, senza rumore abbiamo la certezza di predizione nei punti già osservati, ora invece aggiungiamo un margine/intervallo di varianza. La funzione media e covarianza della *prior* sono iperparametri. Esistono diverse possibilità sia per la media (simple kriging, ordinary kriging, universal kriging) che per la covarianza (Matern, lienare, esponenziale). Diverse scelte portano a comportamenti diversi (in termini di *smoothness* o altro).

Una volta che il modello surrogato è costruito, per performare la *Sensitivity Analysis* è possibile procedere in due modi: si può sostituire il vero modello $G(\mathbf{X})$ con la media della *posterior* e fare *Sensitivity Analysis* sulla media stessa (è molto efficiente, tuttavia produce del *bias* e non permette di quantificare l'errore), oppure si può sostituire il vero modello $G(\mathbf{X})$ con il surrogato $Z_N(\mathbf{X})$ avente come distribuzione la *posterior* (con questo approccio è possibile quantificare l'errore).

8. Forward Uncertainty Quantification

Consideriamo il problema di diffusione $-div(a \nabla u) = f$ su un certo dominio D e con condizioni al contorno. La soluzione $u(x)$ può rappresentare la temperatura in una stanza, la concentrazione di inquinante in un fiume o la posizione di una membrana caricata da $f(\cdot)$. Di solito $a(x)$ e $f(x)$ sono fisse, tuttavia è possibile considerarle aleatorie, così che anche la soluzione $u(x)$ è aleatoria. Supponiamo che l'obiettivo sia di calcolare una certa *Quantity of Interest* (QoI) $Q(u(x))$, una funzione della soluzione $u(x)$ (ad esempio la media in una certa zona o anche $u(x_0)$ per un dato x_0). Se $a(x)$ e $f(x)$ fossero state fisse, questo sarebbe stato un problema risolvibile con metodi di calcolo. In questo framework, invece, dobbiamo gestire anche l'*uncertainty*. La stima di $Q(u(x))$ deve tener conto che per ogni realizzazione di $a(x)$ e $f(x)$ abbiamo una soluzione diversa.

■ Crude Monte Carlo Approach

Supponiamo di avere un input aleatorio Y di dimensione d a una PDE/ODE, la cui soluzione (approssimata) è data da U_h (dove h è il parametro di discretizzazione usato nei metodi numerici per la risoluzione dell'equazione differenziale, la vera soluzione sarebbe U). Il nostro obiettivo è calcolare una QoI $Q_{h,d}$, funzione (non) lineare della soluzione U_h . La vera QoI Q non è accessibile, siccome non è possibile arrivare alla vera soluzione U . Tuttavia, possiamo assumere che $\mathbb{E}[Q_{h,d}] \rightarrow \mathbb{E}[Q]$ e che $|\mathbb{E}[Q_{h,d} - Q]| = O(h^\alpha) + O(d^{-\alpha'})$, per due valori α e α' .

Possiamo procedere con la versione *cruda* del metodo Monte Carlo, ottenendo lo stimatore $\hat{Q}_{CMC} := \hat{Q}_h$. Richiamando la *Bias-Variance decomposition* arriviamo a concludere che l'errore totale (MSE) è composto da due fattori:

$$\begin{aligned} MSE &:= \mathbb{E}[(\hat{Q}_h - \mathbb{E}[Q])^2] \\ &= (\mathbb{E}[\hat{Q}_h] - \mathbb{E}[Q])^2 + Var(\hat{Q}_h) \\ &= \underbrace{(\mathbb{E}[\hat{Q}_h] - \mathbb{E}[Q])^2}_{\text{discretization error}} + \underbrace{Var(Q_{h,d})/N}_{\text{sampling error}} \end{aligned}$$

Un secondo *bound*, in cui sottolineiamo per comodità la dipendenza della QoI dalla soluzione e la dipendenza dalla variabile aleatoria Y della soluzione, è dato da $\mathbb{E}[Q(u(y))] - \hat{Q}_h = \varepsilon^Q(h) + \varepsilon_h^Q(N)$, dove il primo termine rappresenta il *discretization error* e il secondo lo *statistical error*. L'obiettivo è di studiare i due contributi separatamente.

• Discretization error $\varepsilon^Q(h)$

Per riuscire a capire i limiti di questo termine, dobbiamo sfruttare le conoscenze che abbiamo sulla stima dell'errore dato un metodo numerico. Se consideriamo, ad esempio, lo schema di Eulero (*Explicit Euler*) e supponiamo, in aggiunta, che Q sia Lipschitz, allora otteniamo che $\varepsilon^Q(h) \leq c_Q c_u h^\alpha$, dove c_Q è la costante di Lipschitz, c_u e α sono costanti del metodo numerico.

• Statistical error $\varepsilon_h^Q(N)$

Consideriamo lo stesso framework del *discretization error*. È possibile dimostrare che se Q è Lipschitz allora $Var(Q(u_h(y)))$ è finita. Possiamo quindi sfruttare il teorema centrale del limite e concludere che $|\varepsilon_h^Q(N)| \leq z_{1-\alpha/2} \sqrt{Var(Q(u_h(y)))}/\sqrt{N}$.

Tornando ora alla stima dell'errore, siamo in grado di dire:

$$\mathbb{E}[Q(u(y))] - \hat{Q}_h = \varepsilon^Q(h) + \varepsilon_h^Q(N) \leq C_{discr} h^\alpha + C_{stat} / \sqrt{N}$$

In particolare, evidenziamo solo la struttura di h e N , siccome sono gli unici due componenti di cui abbiamo bisogno per quantificare quanto sarà dispendioso lo stimatore Monte Carlo. L'*Abstract Complexity Theorem for Crude Monte Carlo* ci fornisce il risultato: per determinati valori α e γ (dipendenti dal problema), si ha che, per ottenere $MSE < \varepsilon^2$ il costo dello stimatore Monte Carlo sarà dato da $\tilde{\varepsilon}^{(2+d\gamma/\alpha)}$.

■ Multi-Level Monte Carlo

L'idea del *Multi-Level Monte Carlo* è di considerare più modelli e di fare una media dei risultati ottenuti su ciascuno. In particolare, l'idea è di stimare il valore atteso della quantità di interesse Q , cioè $\mathbb{E}[Q(u(y))]$, dividendo il calcolo in una sequenza di livelli (sequenza di modelli caratterizzati da diversi valori del parametro di discretizzazione (h_0, \dots, h_L) , dove h_L è la discretizzazione più fine e più dispendiosa, mentre h_0 è la discretizzazione più coarsa)).

Formalizziamo. Vogliamo applicare l'idea delle *Control Variates* per il problema della stima di $\mathbb{E}[Q(u_h(y))]$, dove $u_h(y)$ è l'approssimazione numerica della soluzione di un dato problema. Per facilitare, introduciamo $P = Q(u(y))$ e $P_h = Q(u_h(y))$. Per applicare il metodo delle *Control Variates* è necessario introdurre una variabile aleatoria Y , correlata con P_h , di cui conosciamo la media. Dopodiché consideriamo $P_{h,Y} = P_h - \beta(Y - \mathbb{E}[Y])$ e ricaviamo lo stimatore \hat{P}_h^{CV} (con l'approccio standard Monte Carlo). Ci sono due modi per scegliere Y (e β).

• Multi-Fidelity Monte Carlo

La Y è un modello surrogato e β è ottimizzato numericamente.

• Multi-Level Monte Carlo

La Y è una versione più coarsa di P_h , ad esempio P_{2h} , con $\beta = 1$.

In entrambi i casi $\mathbb{E}[Y]$ non è conosciuta e va quindi stimata con uno stimatore Monte Carlo indipendente (per questo stimatore è possibile usare molti più campioni siccome (in entrambi i casi) la Y costa meno che P_h). Per convenzione, denotiamo P_h con P_1 e Y con P_0 e supponiamo che la Y

sia semplicemente una versione più coarsa di P_h (con $h_1 < h_0$). Applichiamo Monte Carlo e otteniamo $\hat{\mu}_h^{CV}$, contenente però la quantità $\mathbb{E}[P_0]$ sconosciuta. Lo stimatore $\hat{\mu}_h^{CV}$ può essere approssimato da uno stimatore a due livelli $\hat{\mu}_1^{MLMC}$, il cui costo è di $N_0 C_0 + N_1 C_1$, dove N_0 e C_0 sono il numero di campioni e il costo per campione dal modello P_0 e N_1 e C_1 sono il numero di campioni e il costo per campione dal modello $P_1 - P_0$. Se i costi C_i sono fissi, è possibile scegliere i valori N_i in modo da ottenere una riduzione massima della varianza.

La generalizzazione di questa idea è detta metodo *Multi Level Monte Carlo* (MLMC), dove vengono generati diversi modelli, ciascuno con una discretizzazione diversa. La generalizzazione al caso *multi-level* consiste nel considerare una sequenza di modelli P_0, \dots, P_L , dove i modelli P_0, \dots, P_{L-1} approssimano P_L con un'accuratezza crescente ma anche ad un prezzo crescente. Questi modelli corrispondono a una sequenza di mesh sempre più fitte: se definiamo $P_\ell = P_{h_\ell} = Q(u_{h_\ell}(\mathbf{y}))$ allora $h_0 > h_1 > \dots > h_L$. Nel caso di infinito costante ($h_{\ell-1} = m \cdot h_\ell$) il metodo prende il nome di *Geometric MLMC*.

Lo stimatore *Multi-Level Monte Carlo* $\hat{\mu}_L^{MLMC}$ prevede un campionamento di N_ℓ elementi dal modello $P_\ell - P_{\ell-1}$. Come determinare N_j ottimale per ciascun livello? Con un problema di ottimizzazione, tenendo conto il costo per-campione C_ℓ per un campione dal modello $P_\ell - P_{\ell-1}$ e la costrizione che la varianza totale sia sotto una data tolleranza.

Using a MLMC method, the MC complexity is always improved for optimal choice of N_ℓ .

Part III: Inverse Uncertainty Quantification

9. Statistical Inverse Problems and Parameter Estimation

- Frequentist approach
- Bayesian vs. frequentist
- Prior, likelihood, posterior
- Cosa fare a partire dalla posterior?
 $\hat{\theta}_{MAP}, \hat{\theta}_{CM}, Cov(\theta|\mathbf{z}), CR_\alpha, \pi_{pred}(z_{new}|\mathbf{z})$
- Nella pratica nessuno calcola la posterior
(l'integrale al denominatore è un problema)
 - MCMC con target distribution $\pi(\theta|\mathbf{z})$
 - Metropolis
 - Metropolis, gaussian proposal, uniform prior
 - Adaptive Metropolis, gaussian proposal, uniform prior
 - Metropolis, gaussian proposal, generic prior
 - Metropolis-Hastings
 - Gibbs-sampling
- Come mai funzionano le MCMC?
 - Markov Chain theory
 - Markov Chain: definizione, omogenea, stazionaria, periodica/aperiodica, irriducibile
 - Basic limit theorem, reversibility
 - MCMC definition
 - Ergodic theorem
 - CLT for Markov Chains
 - MCMC - Metropolis-Hastings revisited

COMPUTATIONAL STATISTICS

Simulating Statistical Models

- 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 μ 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. (Monte Carlo non è in grado di fare meglio)
- Come possiamo ridurre $\hat{\sigma}$?
→ *Variance Reduction Techniques* (Chp. 5)
- Come possiamo migliorare la costruzione di $\hat{\mu}$? (Cambiare $1/\sqrt{N}$?)
È possibile scegliere dei punti secondo regole adeguate in modo da ottenere una velocità di convergenza pari a $\log(N)^{d-1}/\sqrt{N}$. Tuttavia, i punti non saranno più random/aleatori, bensì scelti con schemi/criteri deterministici.
→ *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. 8)

Forward Uncertainty Quantification and Sensitivity Analysis

Parliamo ora di Sensitivity Analysis. Dato un modello a più input, vogliamo poter dire quanta incertezza genera nell'output ciascun input.

→ *Sensitivity Analysis* (Chp. 7)

- Partiamo da *Local Methods* (derivate, scatterplot, screening (Morris)), che però non esplorano adeguatamente lo spazio degli input.
- Passiamo allora ai *Global Methods*, ovvero *Variance-Based Sensitivity Analysis*, introducendo i *Sobol' indeces* (first order/total effects).
- I *Sobol' indeces* prevedono il calcolo di momenti (medie/varianze), usiamo i metodi Quasi Monte Carlo su *Sobol sequences* per risparmiare a livello computazionale, tuttavia è comunque troppo dispendioso. Introduciamo due possibili alternative:

1. *Moment Independent importance measure*: studiamo direttamente la variazione in distribuzione, senza calcolare momenti particolari (media/varianza): possiamo analizzare la densità (δ -sensitivity measure) o la funzione di ripartizione (PAWN)
2. sostituiamo il modello originale con un surrogato/emulatore più semplice (*Polynomial Chaos Expansion*, *Gaussian Process Regression*)

ACCEPTANCE-REJECTION:

$$\begin{aligned} \mathbb{P}(X \leq x) &= \mathbb{P}(Y \leq x | U \leq \frac{\tilde{F}(Y)}{c \cdot g(Y)}) = \frac{\mathbb{P}(Y \leq x, U \leq \frac{\tilde{F}(Y)}{c \cdot g(Y)})}{\mathbb{P}(U \leq \frac{\tilde{F}(Y)}{c \cdot g(Y)})} = \mathbb{E}[\mathbb{1}_{[U \leq \frac{\tilde{F}(Y)}{c \cdot g(Y)}]}] = \mathbb{E}\left[\frac{\tilde{f}(Y)}{c \cdot g(Y)}\right] \\ &= \frac{\int_{-\infty}^x \left(\int_0^{\frac{\tilde{F}(y)}{c \cdot g(y)}} du \right) g(y) dy}{\int_R \frac{\tilde{f}(y)}{c \cdot g(y)} g(y) dy} = \frac{\frac{x}{c} \int_{-\infty}^x \tilde{f}(y) dy}{\frac{x}{c} \int_R \tilde{f}(y) dy} = \int_{-\infty}^x f(y) dy \end{aligned}$$

STRATIFICATION / STRATIFIED SAMPLING

$$\mathbb{P}(\vec{x} \in \Omega_j) = \int_{\Omega_j} \mathbb{1}_{\Omega_j}(\vec{x}) f(\vec{x}) d\vec{x} = p_j \quad (\sum_{j=1}^s p_j = 1)$$

$$f_j(\vec{x}) = \frac{1}{p_j} f(\vec{x}) \mathbb{1}_{\Omega_j}(\vec{x})$$

→

STRATIFICATION

for $j = 1, \dots, s$:

genera N_j iid repliche $\vec{z}_j^{(i)} \sim z_j$

$$\text{calcola: } \hat{\mu}_j = \frac{1}{N_j} \sum_{i=1}^{N_j} \vec{z}_j^{(i)}$$

$$\text{calcola: } \hat{\mu}_{\text{str}} = \sum_{j=1}^s p_j \hat{\mu}_j$$

$$z_j = \gamma(\vec{x}_j) = \gamma(\vec{x}|_{\Omega_j})$$

$$\vec{x}|_{\Omega_j} = \vec{x}_j \sim f_j(\cdot)$$

• PROPORTIONAL ALLOCATION

$$N_j = N \cdot p_j \implies \text{Var}(\hat{\mu}_{\text{str}}) = \frac{\text{Var}(z) - \text{Var}(\mathbb{E}[z|\vec{x}])}{N} \leq \frac{\text{Var}(z)}{N} = \text{Var}(\hat{\mu}_{\text{mc}})$$

from the variance formula:
 $\text{Var}(z) = \mathbb{E}[\text{Var}(z|\vec{x})] + \text{Var}(\mathbb{E}[z|\vec{x}])$

• OPTIMAL ALLOCATION

(best choice of (N_j) that minimize $\text{Var}(\hat{\mu}_{\text{str}})$ (Lagrangean with $\sum_j N_j = N$))

$$N_j = \frac{N \cdot p_j \cdot \sigma_j}{\sum_{k=1}^s p_k \sigma_k}, \quad \sigma_j = \sqrt{\text{Var}(z_j)}$$

$$\implies \text{Var}(\hat{\mu}_{\text{str}})|_{\text{optimal}} \leq \text{Var}(\hat{\mu}_{\text{str}})|_{\text{proportional}} \leq \text{Var}(\hat{\mu}_{\text{mc}})$$

→

STRATIFICATION WITH OPTIMAL ALLOCATION

* for $j = 1, \dots, s$:

genera \bar{N}_j iid $\vec{z}_j^{(i)} \sim z_j$

$$\text{stima } \hat{\sigma}_j^2 = \sum_{i=1}^{\bar{N}_j} (\vec{z}_j^{(i)} - \hat{\mu}_j)^2 / (\bar{N}_j - 1)$$

$$\text{stima } \hat{\sigma}^2 = \sum_{j=1}^s p_j \hat{\sigma}_j^2$$

$$\text{scopri } N = \left(\frac{z_1 - z_2}{\hat{\sigma}} \right)^2$$

$$\boxed{\text{STRATIFICATION}} \text{ con } N_j = N \cdot \frac{p_j \sigma_j}{\sum_{k=1}^s p_k \sigma_k}$$

LATIN HYPERCUBE SAMPLING



LHS

- genera N iid $\tilde{U}^{(i)} \sim U([0, 1]^d)$
- genera $\tilde{u}_1, \dots, \tilde{u}_d$ permutazioni di $\{1, \dots, N\}$
- setta $\tilde{V}^{(i)} = (u_{1i}, \dots, u_{di})$
- setta $\tilde{X}^{(i)} = \frac{1}{N} (\tilde{V}^{(i)} - 1 + \tilde{U}^{(i)})$
- calcola $\hat{\mu}_{LHS} = \frac{1}{N} \sum_{i=1}^N \gamma(\tilde{X}^{(i)})$

$$d=2, N=4$$

$$\tilde{u}_1 = [4 \ 3 \ 1 \ 2]$$

$$\tilde{u}_2 = [2 \ 1 \ 3 \ 4]$$

$$\tilde{V}^{(1)} \ \tilde{V}^{(2)} \ \tilde{V}^{(3)} \ \tilde{V}^{(4)}$$

$$\begin{matrix} 1 & 2 & 3 & 4 \\ 3 & 1 & 2 & 4 \\ 2 & 4 & 1 & 3 \end{matrix}$$

$$\begin{matrix} 1 & 2 & 3 & 4 \\ 3 & 1 & 2 & 4 \\ 2 & 4 & 1 & 3 \end{matrix}$$



LHS ESTIMATOR

- for $k=1, \dots, K$
 $\boxed{LHS} \rightarrow \hat{\mu}_{LHS}^{(k)}$
- calcola: $\hat{\mu}_{LHS} = \frac{1}{K} \sum_{k=1}^K \hat{\mu}_{LHS}^{(k)}$
- calcola $\hat{\sigma}_{LHS}^2 = \frac{1}{K-1} \sum_{k=1}^K [\hat{\mu}_{LHS}^{(k)} - \hat{\mu}_{LHS}]^2$

1	2
3	4

For $Z = \gamma(\tilde{X})$, $\tilde{X} \sim U([0, 1]^d)$, $E[Z] < \infty$, $\text{Var}(Z) < \infty$:

$$\text{Var}(\hat{\mu}_{LHS}) = \frac{\text{Var}(\gamma - \gamma^{\text{add}})}{N} + O\left(\frac{1}{N}\right)$$

dove γ^{add} è la miglior approssimazione di γ in termini di modello additivo.

$$(\gamma(\tilde{X}) = \gamma_0 + \sum_{j=1}^d \gamma_j(x_j) + r(\tilde{X}) = \gamma^{\text{add}}(\tilde{X}) + r(\tilde{X}))$$

QUASI-MONTE CARLO METHODS

Discrepancy: $\vec{y} \in [0, 1]^d \rightarrow [\vec{0}, \vec{y}] = \prod_{i=1}^d [0, y_i]$, $\text{Vol}([\vec{0}, \vec{y}]) = \prod_{i=1}^d y_i$

$$\hat{\text{Vol}}([\vec{0}, \vec{y}]) = \frac{1}{N} \sum_{i=1}^N \mathbb{1}_{[\vec{0}, \vec{y}]}(\tilde{x}^{(i)}) = \frac{\#\tilde{x}^{(i)} \in [\vec{0}, \vec{y}]}{N}$$

$$\Delta(\vec{y}) = \hat{\text{Vol}}([\vec{0}, \vec{y}]) - \text{Vol}([\vec{0}, \vec{y}])$$

(discrepancy function)

Discrepancy measures: $D_N^* = \sup_{\vec{y} \in [0, 1]^d} |\Delta(\vec{y})|$, $D_{N,q} = \left(\int_{[0, 1]^d} |\Delta(\vec{y})|^q d\vec{y} \right)^{1/q}$

Zaremba's identity:

$$\int_0^1 \gamma(x) dx - \frac{1}{N} \sum_{i=1}^N \gamma(x^{(i)}) = \int_0^1 \gamma'(y) \Delta(y) dy$$

Koksma's inequality:

$$\left| \int_0^1 \gamma(x) dx - \frac{1}{N} \sum_{i=1}^N \gamma(x^{(i)}) \right| \leq \|\gamma'\|_{L^p} \|\Delta\|_{L^q} D_{N,q}$$

$$\leq \|\gamma'\|_{L^2} \|\Delta\|_{L^\infty} D_N^*$$

ricorre $V(\gamma)$ e immutabile, dobbiamo trovare sequenze a basse discrepanze



RANDOMLY SHIFTED QMC

- genera $\tilde{U}^{(1)}, \dots, \tilde{U}^{(K)} \sim U([0, 1]^d)$
- for $k=1, \dots, K$:
 - calcola $\hat{\mu}_{QMC}^{(k)} = \frac{1}{N} \sum_{i=1}^N \gamma(\{\tilde{x}^{(i)} + \tilde{U}^{(k)}\})$
 - calcola $\hat{\mu}_{QMC} = \frac{1}{K} \sum_{k=1}^K \hat{\mu}_{QMC}^{(k)}$
 - calcola $\hat{\sigma}_{QMC}^2 = \frac{1}{K-1} \sum_{k=1}^K [\hat{\mu}_{QMC}^{(k)} - \hat{\mu}_{QMC}]^2$

$\{x\} = \text{fractional part of } x$

LOCAL METHODS

Consideriamo un modello $Y = Y(z_1, \dots, z_d)$

- PARTIAL DERIVATIVES:

$$S_{z_i}^P = \frac{\partial Y}{\partial z_i} \quad S_{z_i}^S = \left(\frac{\partial Y}{\partial z_i} \right) \cdot \frac{\sigma_{z_i}}{\sigma_Y}$$

output derivative sigma-normalized output derivative

Why?

Consideriamo $Y = \sum_{j=1}^d z_j$, $z_j \sim N(0, \sigma_j^2)$
con $\sigma_1 < \dots < \sigma_d$.

Riconosciamo dati e osserviamo uno scatterplot per $d=4$:



Y è più sensibile per z_4 che per z_1 ,
eppure $S_{z_1}^P = S_{z_4}^P = 1$.

- LINEAR REGRESSION

Modelliamo come: $\hat{Y} = b_0 + \sum_{j=1}^d b_{z_j} z_j$

Standardized regression coefficients (SRC): $\hat{\beta}_{z_j} = b_{z_j} \frac{\sigma_{z_j}}{\sigma_Y}$

- se il modello è valido e le z_i sono indipendenti:

$$\sigma_Y^2 = \sum_{j=1}^d b_{z_j}^2 \sigma_{z_j}^2 \Rightarrow \hat{\beta}_{z_j}^2 = \frac{b_{z_j}^2 \sigma_{z_j}^2}{\sum_{j=1}^d b_{z_j}^2 \sigma_{z_j}^2} = \text{contributo frazionario alle varianze di } Y \text{ dato dalle var. } z_j$$

- se le z_i sono indipendenti:

$$R_Y^2 = \sum_{j=1}^d (\hat{\beta}_{z_j})^2 = \text{coefficiente di correlazione}$$

(la validità di $\hat{\beta}_{z_j}$'s come misura di sensibilità dipende da quanto il modello di regressione "filtra" i dati)

- SCREENING METHODS

Consideriamo più esplicitamente: $Y = f(x_1, \dots, x_d)$.

Discretizziamo ciascuna x_j in p livelli, creando una griglia.

Scegliamo $r = \# \text{ repetitions}$



ELEMENTARY EFFECTS

for $j = 1, \dots, d$:

- for $i = 1, \dots, r$:

• campioniamo un $\tilde{x}^{(i)}$ dalle griglie

• calcoliamo $EE_j^{(i)} = \frac{1}{\Delta} (f(\tilde{x}^{(i)}) + \Delta \tilde{e}_j) - f(\tilde{x}^{(i)})$

end

• calcoliamo $\mu_j^* = \frac{1}{r} \sum_{i=1}^r |EE_j^{(i)}|$

• calcoliamo $\sigma_j = \sqrt{\frac{1}{r} \sum_{i=1}^r (EE_j^{(i)} - \frac{1}{r} \sum_{i=1}^r |EE_j^{(i)}|)^2}$

GLOBAL METHODS

Consideriamo un modello $Y = f(x_1, \dots, x_d)$

- FIRST ORDER EFFECTS

$$\text{Var}_{\tilde{x}_{-i}}(Y | X_i = x_i)$$

$$\Rightarrow E_{X_i} [\text{Var}_{\tilde{x}_{-i}}(Y | X_i)] = \text{Var}(Y) - \text{Var}_{x_i} (E_{\tilde{x}_{-i}} [Y | X_i]) \leq \text{Var}(Y)$$

$$\Rightarrow s_i = \frac{\text{Var}_{x_i} (E_{\tilde{x}_{-i}} [Y | X_i])}{\text{Var}(Y)} \quad \uparrow \text{if } x_i \text{ è importante}$$

first order sensitivity index

- modelli additivi: $\sum_{i=1}^d s_i = 1$

- modelli non-additivi: $\sum_{i=1}^d s_i < 1$

$$\left(\frac{\text{Var}_{x_i} (E_{\tilde{x}_{-i}} [Y | X_i])}{\text{Var}(Y)} \right)$$

• TOTAL EFFECT

$$S_{T_i} = 1 - \frac{\text{Var}_{\tilde{x}_{-i}}(\mathbb{E}_{X_i}[Y|\tilde{x}_{-i}])}{\text{Var}(Y)} = \frac{\mathbb{E}_{\tilde{x}_{-i}}[\text{Var}_{X_i}(Y|\tilde{x}_{-i})]}{\text{Var}(Y)}$$

varianza totale - varianza di tutte le variabili / sisteme se consideriamo X_i fisse

• HDMR EXPANSION

Consideriamo $Y = f(X_1, \dots, X_d)$, $X_j \in [0,1]$.

La HDMR (High-Dimensional Model Representation) è data da:
(HDMR expansion / FANOVA decomposition)

$$Y = f_0 + \sum_{i=1}^d f_i(x_i) + \sum_{i < j} f_{ij}(x_i, x_j) + \dots + f_{1,\dots,d}(X_1, \dots, X_d)$$

(non unica).

Perciò, se qualiasi termine $f_{i_1, \dots, i_s}(X_{i_1}, \dots, X_{i_s})$ è tale che:

$$\int_0^1 f_{i_1, \dots, i_s}(X_{i_1}, \dots, X_{i_s}) dX_{i_k} = 0 \quad k=1, \dots, s$$

allora tutti i termini sono ortogonali ($\int f_{i_1, \dots, i_s}(X_{i_1}, \dots, X_{i_s}) f_{j_1, \dots, j_t}(X_{j_1}, \dots, X_{j_t}) d\tilde{x} = 0$) e l'expansion è unica.

Se $f(\tilde{x})$ è integrabile:

$$f_0 = \int f(\tilde{x}) d\tilde{x}$$

$$f_i(x_i) = \int f(\tilde{x}) d\tilde{x}_{-i} - f_0$$

$$f_{ij}(x_i, x_j) = \int f(\tilde{x}) d\tilde{x}_{-i-j} - f_i(x_i) - f_j(x_j) - f_0$$

Se $f^2(\tilde{x})$ è integrabile:

$$\text{Var}(Y) = \sum_{i=1}^d V_i + \sum_{i < j} V_{ij} + \dots + V_{1,2,\dots,d}$$

$$V_i = \text{Var}_{X_i}(\mathbb{E}_{\tilde{x}_{-i}}[Y|\tilde{x}_i])$$

$$V_{ij} = \text{Var}_{X_i X_j}(\mathbb{E}_{\tilde{x}_{-i} \tilde{x}_{-j}}[\mathbb{E}(Y|X_i, X_j)])$$

$$\Rightarrow \begin{cases} \text{First Sobol index: } S_i = \frac{V_i}{\text{Var}(Y)} \\ \text{Total effect (Sobol) index: } S_{T_i} = \frac{\sum \text{every } V_i \text{ with } i}{\text{Var}(Y)} \end{cases}$$

• NUMERICAL EVALUATION OF SOBOL INDICES

- generiamo una Sobol sequence di N elementi e dim = $2d$
- definiamo:

$$A = \begin{bmatrix} x_1^{(1)} & \dots & x_d^{(1)} \\ \vdots & \ddots & \vdots \\ x_1^{(N)} & \dots & x_d^{(N)} \end{bmatrix} \quad B = \begin{bmatrix} x_{d+1}^{(1)} & \dots & x_{d+d}^{(1)} \\ \vdots & \ddots & \vdots \\ x_{d+1}^{(N)} & \dots & x_{d+d}^{(N)} \end{bmatrix} \quad C_i = \begin{bmatrix} x_{d+1}^{(1)} & \dots & x_i^{(1)} & \dots & x_{d+d}^{(1)} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{d+1}^{(N)} & \dots & x_i^{(N)} & \dots & x_{d+d}^{(N)} \end{bmatrix}$$

- definiamo: $\tilde{y}_A^i = f(A)$, $\tilde{y}_B^i = f(B)$, $\tilde{y}_{C_i}^i = f(C_i)$

$$\bullet \quad S_i = \frac{(\tilde{y}_A^T \tilde{y}_A^i)/N - \bar{y}_A^2}{(\tilde{y}_A^T \tilde{y}_A^i)/N - \bar{y}_A^2}, \quad S_{T_i} = \frac{(\tilde{y}_B^T \tilde{y}_{C_i}^i)/N - \bar{y}_A^2}{(\tilde{y}_A^T \tilde{y}_A^i)/N - \bar{y}_A^2}$$

B-i

A-i

/

SURROGATE MODELS

Modello reale: $Y = G(\vec{x})$, $\vec{x} = (x_1, \dots, x_d) \in \mathbb{R}^d$, $\vec{x} \sim f(\cdot)$.

Abbiamo le osservazioni: $\vec{x} = \{\vec{x}^{(1)}, \dots, \vec{x}^{(N)}\}$, $Y = \{y^{(i)} = G(\vec{x}^{(i)})\}$

- POLYNOMIAL CHAOS EXPANSION (PCE)

Spectral representation: $Y = \sum_{k=0}^{\infty} y_k z_k$

- $\{z_k\}_{k=0}^{\infty}$ insieme di polinomi ortonormali w.r.t. $f(\cdot)$ tali che $z_k = \phi_k(\vec{x})$ e tali che:

$$\mathbb{E}[\phi_m(\vec{x}) \phi_n(\vec{x})] = \int \phi_m(\vec{x}) \phi_n(\vec{x}) f(\vec{x}) d\vec{x} = a_m \delta_{mn}$$

$$a_m = \mathbb{E}[\phi_m^2(\vec{x})]$$

- $\{y_k\}_{k=0}^{\infty}$ coefficienti (coordinate di Y nelle calcolate le basi e deciso K ottieniamo:

$$Y = G(\vec{x}) = \sum_{k=0}^K y_k \phi_k(\vec{x}) + \varepsilon$$

$$\Rightarrow \{y_k\}_{k=0}^K = \vec{y} = \arg \min \mathbb{E}[(G(\vec{x}) - \sum_{k=0}^K y_k \phi_k(\vec{x}))^2]$$

$$\Rightarrow \hat{\vec{y}} = \arg \min \frac{1}{N} \sum_{i=1}^N (G(\vec{x}^{(i)}) - \sum_{k=0}^K y_k \phi_k(\vec{x}^{(i)}))^2$$

- GAUSSIAN PROCESS REGRESSION (GPR)

Ipotizziamo che $G(\vec{x})$ sia una realizzazione del processo gaussiano Z . Partiamo da $Z \sim GP(\mu_Z, C_Z)$, dove μ_Z e C_Z rappresentano le priori, e aggiorniamo con X e Y , ottenendo:

$$\mu_{Z|Y}(x_{\text{new}}), C_{Z|Y}(x_{\text{new}}, x_{\text{new}}) \quad \forall x_{\text{new}}$$

(che caratterizzano $Z|Z(Y) = Y$)

FORWARD UC - CRUDE MONTE CARLO

Situazione: $\vec{y} = \text{random input a una PDE/ODE}$

$u(\vec{y}) = \text{soltuzione del problema a PDE/ODE}$

$$Q(u(\vec{y})) = Q \circ I$$

Goal: stimare $\mathbb{E}[Q(u(\vec{y}))]$.

→ CRUDE MONTE CARLO: $\mathbb{E}[Q(u(\vec{y}))] \approx \frac{1}{N} \sum_{i=1}^N Q(u(\vec{y}^{(i)}))$

tuttavia $u(\vec{y})$ non è conoscibile,

usiamo un'approssimazione numerica $u_h(\vec{y})$

basata sulla discretizzazione h .

$$\Rightarrow \mathbb{E}[Q(u(\vec{y}))] \approx \hat{Q}_h = \frac{1}{N} \sum_{i=1}^N Q(u_h(\vec{y}^{(i)}))$$

$$\mathbb{E}[Q(u(\vec{y}))] - \hat{Q}_h = \underbrace{\mathbb{E}[Q(u(\vec{y})) - Q(u_h(\vec{y}^{(i)}))] + \mathbb{E}[Q(u_h(\vec{y}^{(i)})) - \hat{Q}_h]}$$

$\epsilon_h^Q(h)$
discretization error

$$|\epsilon_h^Q(h)| \leq C_Q C_u h^\alpha$$

(α dipende dal metodo)

$\epsilon_h^Q(N)$
statistical error

$$|\epsilon_h^Q(N)| \leq z_{1-\alpha} \sqrt{\text{Var}(Q(u_h(\vec{y})))} / \sqrt{N}$$

(grazie al TCL)

$$\Rightarrow |\mathbb{E}[Q(u(\vec{y}))] - \hat{Q}_h| \leq C_{\text{discr.}} h^\alpha + C_{\text{stat.}} \frac{1}{\sqrt{N}}$$

Abstract complexity thm. for standard MC:

$$\mathbb{E}[|\hat{Q}_h - Q|] = O(h^\alpha)$$

$$\text{cost}(\hat{Q}_h) = O(h^{-d\alpha})$$

$$\text{MSE} = \mathbb{E}[(\hat{Q}_h - \mathbb{E}[Q])^2] < \varepsilon$$

$$\Rightarrow \text{cost}(\hat{Q}_h) = O(\varepsilon^{-(2+\frac{d\alpha}{2})})$$

MULTI-LEVEL MONTE CARLO

$$P = Q(u(y)), \quad P_h = Q(u_h(y))$$

Introduciamo γ e definiamo: $P_{h,\gamma} = P_h + \beta(\gamma - \mathbb{E}[\gamma])$

$$\Rightarrow \hat{\mu}_h^{CV} = \frac{1}{N} \sum_{i=1}^N (P_h^{(i)} - \beta \gamma^{(i)}) + \beta \mathbb{E}[\gamma]$$

(Control Variates)

Multi-level MC: γ è una versione più coarta

$$\rightarrow \text{definiamo } P_1 := P_h, \quad P_0 := \gamma \stackrel{e.g.}{=} P_{2h}$$

$$\Rightarrow \hat{\mu}_h^{CV} = \frac{1}{N_1} \sum_{i=1}^{N_1} (P_1^{(i)} - P_0^{(i)}) + \mathbb{E}[P_0]$$

$$\Rightarrow \hat{\mu}_h^{MLMC} = \frac{1}{N_1} \sum_{i=1}^{N_1} (P_1^{(i)} - P_0^{(i)}) + \frac{1}{N_0} \sum_{i=1}^{N_0} P_0^{(i)}$$

Definiamo:

$$C_0 = \text{cost}(P_0)$$

$$V_0 = \text{Var}(P_0)$$

$$C_1 = \text{cost}(P_1) - \text{cost}(P_0)$$

$$V_1 = \text{Var}(P_1 - P_0)$$

$$\Rightarrow \text{total cost} = C_0 N_0 + C_1 N_1,$$

$$\text{la varianza è minimizzata per } \frac{N_1}{N_0} = \frac{\sqrt{V_1/C_1}}{\sqrt{N_0/C_0}}$$

→ MULTI-LEVEL GENERALIZATION

Consideriamo una sequenza $P_l = P_{h_l} = Q(u_{h_l}(y)) \quad l = 0, \dots, L$

(i modelli P_0, \dots, P_L approssimano P con un'accuratezza crescente ma anche con un prezzo crescente)

$h_l \downarrow \Leftrightarrow$ precisione $\uparrow \Leftrightarrow$ mesh si infittisce

$$\hat{\mu}_L^{MLMC} = \frac{1}{N_0} \sum_{i=1}^{N_0} P_0^{(0,i)} + \sum_{l=1}^L \left[\frac{1}{N_l} \sum_{i=1}^{N_l} (P_l^{(l,i)} - P_{l-1}^{(l,i)}) \right]$$

dove (l,i) indice che ad ogni livello campioniamo di nuovo.

$$\Rightarrow \mathbb{E}[\hat{\mu}_L^{MLMC}] = \mathbb{E}[P_L] = \mathbb{E}[Q_{h_L}] \quad (\text{dove } h_L \text{ è la mesh più fitta})$$

Come scegliamo N_l ?

Problema di ottimizzazione:

$$\{N_l\}_l = \arg \min \sum_{l=0}^L N_l C_l \quad \text{s.t.} \quad \text{Var}(\hat{\mu}_L^{MLMC}) = \sum_{l=0}^L \frac{V_l}{N_l} \leq \varepsilon^2$$

$$\Rightarrow N_l^* = \left\lceil \frac{1}{\varepsilon^2} \sqrt{\frac{V_l}{C_l}} \left(\sum_{j=1}^L \sqrt{V_j C_j} \right) \right\rceil - \text{(upper part)}$$

MCMC - MARKOV CHAIN MONTE CARLO

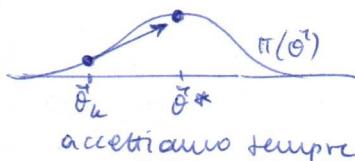
Il goal delle MCMC e' di costruire una catena di Markov (quindi θ_k dipende solo da θ_{k-1}) la cui distribuzione stazionaria e' la posteriore $\pi(\theta | z)$ non e' stata direttamente, e' solo il target (la target distribution) che vogliamo esplorare.



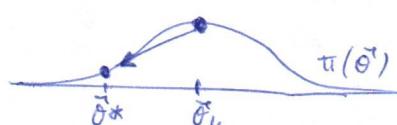
METROPOLIS con target $\pi(\theta)$

- seleziona θ_k^* e setta $k=1$
- campiona θ^* dalla proposal $q(\cdot | \theta_k^*)$
- con probabilita' $\alpha(\theta_k^*, \theta^*) = \min\{1, \frac{\pi(\theta^*)}{\pi(\theta_k^*)}\}$ { accetto }
e fissa $\theta_{k+1}^* = \theta^*$ e $k = k+1$,
altrimenti torna allo step precedente

- la costante di normalizzazione di $\pi(\theta)$ non serve
- $q(\cdot | \theta_k^*)$ deve essere simmetrica ($q(\theta_k^* | \theta^*) = q(\theta^* | \theta_k^*)$)
- quando accettiamo?



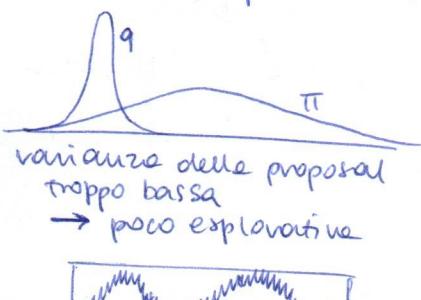
accettiamo sempre



accettiamo con prob. $\alpha = \pi(\theta^*) / \pi(\theta_k)$

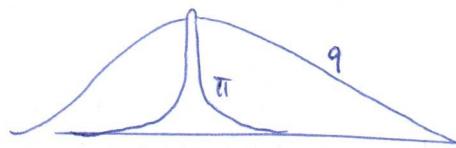
- come deve essere la proposal?

Facile da campionare e vicina a $\pi(\theta)$



varianza delle proposal
troppo bassa

→ poco esplorativa



varianza delle proposal troppo grande
→ si sta perdendo the essential support di π
(generiamo tanti sample da buttare)



Un caso famoso di prop. e' la Gaussian proposal ($N(\theta_0, C)$)

METROPOLIS con GENERIC PRIOR

- selezioniamo θ_0 tale che $\pi(\theta_0 | z) > 0$
- for $k = 1, \dots, \# \text{ samples}$
 - generiamo $\tilde{z} \sim N(0, I)$ ($\dim(\tilde{z}) = \dim(\theta)$)
 - costruiamo il candidato $\theta^* = \theta_{k-1} + R \cdot \tilde{z}$ dove R e' il fattore di Cholesky di C
 - calcoliamo il rapporto:

$$r(\theta^* | \theta_{k-1}) = \frac{\pi(\theta^* | z)}{\pi(\theta_{k-1} | z)} = \frac{\pi(z | \theta^*) \pi_{\text{prior}}(\theta^*)}{\pi(z | \theta_{k-1}) \cdot \pi_{\text{prior}}(\theta_{k-1})}$$

- Fissiamo:

$$\theta_k = \begin{cases} \theta^* & \text{con prob. } \alpha = \min\{1, r\} \\ \theta_{k-1} & \text{con prob. } 1 - \alpha \end{cases}$$

- METROPOLIS - HASTINGS** per una proposal non simmetrica:

$$\text{modifichiamo } r(\theta^* | \theta_{k-1}) = \frac{q(\theta_{k-1} | \theta^*) \pi_{\text{prior}}(\theta^*)}{q(\theta^* | \theta_{k-1}) \pi_{\text{prior}}(\theta_{k-1})}$$

- GIBBS - SAMPLING** se non riusciamo a trovare una proposal per tutti i parametri simultaneamente ($\pi(\theta_i | \theta_{-i})$)

FREQUENTIST APPROACH

Consideriamo il modello statistico $\tilde{Y} = f(\theta_0) + \varepsilon$

dove $\tilde{Y} = [Y_1, \dots, Y_n]$ è un vettore aleatorio la cui realizzazione è data da $\tilde{y} = [y_1, \dots, y_n]$ (misure di un esperimento) e $f(\theta_0) = [f_1(\theta_0), \dots, f_n(\theta_0)]$. La dipendenza fra variabili indipendenti è soppressa in $f(\theta_0)$.

Vogliamo trovare $\hat{\theta}$ in modo che $f(\hat{\theta})$ fitti i dati:

- ORDINARY LEAST SQUARES

$$\hat{\theta}_{LS} = \arg \min_{\theta} \sum_{i=1}^n (Y_i - f_i(\theta))^2$$

- MAXIMUM LIKELIHOOD

$$\hat{\theta}_{MLE} = \arg \max_{\theta} \prod_{i=1}^n f_{Y_i}(y_i; \theta)$$

(Se $\varepsilon_i \stackrel{iid}{\sim} N(0, \sigma^2)$ i due stimatori sono equivalenti)

BAYESIAN APPROACH

- le probabilità sono soggettive e si aggiornano
- i parametri non sono fissi, sono variabili aleatorie con densità proprie, la soluzione delle stime di un parametro è trovare la sua densità

$$\text{Bayes: } P(A|B) = \frac{P(B|A) P(A)}{P(B)} \rightarrow \pi(\theta|z) = \frac{\text{likelihood} \cdot \text{prior}}{\pi(z|\theta) \pi_\theta(\theta)}$$

prior → modo in cui mettiamo della knowledge nel modello (informative / non-informative)
likelihood → come i dati comunicano al modello

prior-posterior conjugate → $\pi_\theta(\theta) \stackrel{def}{=} \pi(\theta|z)$

i loro parametri vengono detti iperparametri
Come vengono "aggiornati"?

$$\begin{aligned} \text{E.g. evento } z &\sim N(\mu, \sigma^2), \quad \mu \sim N(\mu_0, \sigma_0^2) \\ &\Rightarrow \mu|z \sim N(\mu_1, \sigma_1^2) \end{aligned}$$

$$\mu_1 = w_0 \mu_0 + w_1 \bar{z}_n$$

e le medie pesate della prior mean e la media delle osservazioni. Chi ha più peso? Chi ha meno varianza.

Cosa fare a partire dalla posterior?

- stimatore puntuale $\rightarrow \hat{\theta}_{MAP} = \arg \max_{\theta} \pi(\theta|z)$
 $\rightarrow \hat{\theta}_{CM} = E[\theta|z] = \int \theta \cdot \pi(\theta|z) d\theta$
- analisi della varianza $\rightarrow \text{Cov}(\hat{\theta}|z)$ (confidence regions)
- predictive distributions $\rightarrow \pi_{\text{pred}}(z_{\text{new}}|\hat{\theta}) = \int \pi(z_{\text{new}}|\theta) \pi(\theta|z) d\theta$

Se $\dim(\theta)$ è grande \Rightarrow diventa problematico implementare un metodo per ottenere la posterior $\pi(\theta|z)$

\Rightarrow Al posto che cercare/valutare $\pi(\theta|z)$, introduciamo le tecniche MCMC, ovvero tecniche per produrre sistematicamente campioni per esplorare la distribuzione $\pi(\theta|z)$.

- Come scegliamo la matrice covarianza C ?

Se linearizziamo le forme di θ_0 possiamo arrivare a:

(nel modello $\vec{z} = f(\vec{\theta}_0) + \vec{\epsilon}$, $\sigma_{\theta_0}^2 = \text{Var}(z_i)$)

$$C = \text{cov}(\vec{\theta}_{0S}) = \sigma^2 (\mathbf{J}(\vec{\theta}_{0S})^\top \mathbf{J}(\vec{\theta}_{0S}))^{-1} \quad (\mathbf{J}(\vec{\theta}))_{ij} = \frac{\partial f_i(\vec{\theta})}{\partial \theta_j}$$

che e' un buon punto di partenza.

Una versione migliore ha anche uno scaling-factor: $C' = \frac{2.4^2}{\dim(\vec{\theta})} \cdot C$.

In ogni caso, non e' un'ottima scelta tenere fissa C .

→ ADAPTIVE MCMC

Partiamo da C' e per un periodo di burn-in teniamo C' .

Dopo di che modificiamo C ad ogni step.

$$C_n = \frac{2.4^2}{\dim(\vec{\theta})} (\underbrace{\text{cov}(\vec{\theta}_0, \dots, \vec{\theta}_{n-1})}_{\text{campioni generati fino ad allora}} + \varepsilon I)$$

$\underbrace{\varepsilon}_{\text{tende per garantire che } C \text{ sia semidefinita positiva}}$

THEORY BEHIND

- Una catena di Markov e' un processo stocastico in cui, saperndo il presente, passato e futuro sono indipendenti.

Consideriamo:

T countable set of indices

S state space

$\{\theta^{(t)}\}_{t \in T}$ collezione di variabili aleatorie

$$\rightarrow P(\theta^{(n+1)} \in A_{n+1} | \theta^{(n)} = x, \theta^{(n-1)} \in A_{n-1}, \dots, \theta^{(0)} \in A_0) = P(\theta^{(n+1)} \in A_{n+1} | \theta^{(n)} = x)$$

per qualsiasi $A_0, \dots, A_{n-1}, A_{n+1} \subseteq S$.

- Una catena di Markov e':

→ omogenea: se le probabilità di transizione sono il dal tempo
(definiamo il transition kernel di una catena come:

$$(P)_{ij} = P(X_{n+1} = x_j | X_n = x_i) = p_{ij})$$

→ irriducibile: se qualsiasi stato x_j e' raggiungibile da qualsiasi altro stato x_i in un numero finito di steps

→ periodica: se parte dei suoi stati sono visitati regolarmente

m = periodo = maggior-common-divisore $\{n > 0 : P(X_n = x_i | X_0 = x_i) > 0\}$
se $m = 1$ allora lo stato e' aperiodico.

se ogni stato e' aperiodico la catena e' aperiodica.

- Una distribuzione π tale che $\pi = \pi P$ e' detta distribuzione stazionaria delle catene di Markov (con transition kernel P).

Questo equivale a dire che la distribuzione $p^0 = [p_1^0, \dots, p_S^0]$ converge in distribuzione a π spontaneamente in $n \rightarrow \infty$.

$$(p^1 = p^0 P, \quad p^n = p^{n-1} P)$$

Thm. Qualsiasi Markov chain

- definita su S set finito
- irriducibile
- aperiodica

⇒ esiste una distribuzione stazionaria $\pi = [\pi_1, \dots, \pi_S]$

ed e' tale che a partire da qualsiasi p^0 :

$$\lim_{n \rightarrow \infty} p_{ij}^{(n)} = \pi_j$$

$$\pi = \pi P \iff$$

$$\pi_i p_{ij} = \pi_j p_{ji}$$

Detailed balance condition

(def. catene reversibile)

- MCMC consiste nella costruzione di una catena di Markov (inducibile e aperiodica, avente come distribuzione stazionaria π) la target distr. (π). Il transition kernel $p(\vec{\theta}', \vec{\theta})$ è tale che π è stazionario:

$$\pi(\vec{\theta}') p(\vec{\theta}, \vec{\theta}') = \pi(\vec{\theta}) p(\vec{\theta}', \vec{\theta}')$$

(detailed balance condition $\forall \vec{\theta}', \vec{\theta}$).

In particolare sceglieremo un kernel della forma:

$$p(\vec{\theta}', \vec{\theta}) = q(\vec{\theta}' | \vec{\theta}) \alpha(\vec{\theta}', \vec{\theta})$$

arbitrary
transition
kernel
(:= proposal
density)

acceptance
probability

$$\left. \begin{aligned} &\text{dove:} \\ &p(x, y) = P(\vec{\theta}^{n+1} \leq y | \vec{\theta}^n = x) \\ &p(x, y) = \frac{\partial P(x, y)}{\partial y} \end{aligned} \right\}$$

$$\alpha(\vec{\theta}', \vec{\theta}) = \min \left\{ 1, \frac{\pi(\vec{\theta}') q(\vec{\theta}' | \vec{\theta})}{\pi(\vec{\theta}) q(\vec{\theta} | \vec{\theta}')} \right\}$$

Da qui definiamo METROPOLIS-HASTINGS.

Ma come sappiamo che funziona?

= come sappiamo che la Markov Chain convincerà a generare valori da π ?

→ Dimostriamo che la $p(\vec{\theta}', \vec{\theta})$ definita (↑) rispetta le detailed balance condition:

$$\begin{aligned} p_{k-1, k} &= P(\vec{\theta}_k | \vec{\theta}_{k-1}) = P(\text{transitioning } \vec{\theta}_{k-1} \rightarrow \vec{\theta}_k) \\ &= q(\vec{\theta}_k | \vec{\theta}_{k-1}) \cdot \alpha(\vec{\theta}_{k-1}, \vec{\theta}_k) \\ &= q(\vec{\theta}_k | \vec{\theta}_{k-1}) \min \left\{ 1, \frac{\pi(\vec{\theta}_k) q(\vec{\theta}_{k-1} | \vec{\theta}_k)}{\pi(\vec{\theta}_{k-1}) q(\vec{\theta}_k | \vec{\theta}_{k-1})} \right\} \end{aligned}$$

det. bal. condition:

$$\pi(\vec{\theta}_{k-1}) p_{k-1, k} = \pi(\vec{\theta}_k) p_{k, k-1}$$

E considerando: $x \min \{1, \frac{y}{x}\} = \min \{x, y\} = y \min \{1, \frac{x}{y}\}$

$$\begin{aligned} \pi(\vec{\theta}_{k-1}) p_{k-1, k} &= \pi(\vec{\theta}_{k-1}) q(\vec{\theta}_k | \vec{\theta}_{k-1}) \min \left\{ 1, \frac{\pi(\vec{\theta}_k) q(\vec{\theta}_{k-1} | \vec{\theta}_k)}{\pi(\vec{\theta}_{k-1}) q(\vec{\theta}_k | \vec{\theta}_{k-1})} \right\} \\ &= \pi(\vec{\theta}_k) q(\vec{\theta}_{k-1} | \vec{\theta}_k) \min \left\{ 1, \frac{\pi(\vec{\theta}_{k-1}) q(\vec{\theta}_k | \vec{\theta}_{k-1})}{\pi(\vec{\theta}_k) q(\vec{\theta}_{k-1} | \vec{\theta}_k)} \right\} \\ &= \pi(\vec{\theta}_k) p_{k, k-1} \end{aligned}$$

Erodic theorem.

$\{X^{(i)}\}_{i \in \mathbb{N}}$ catene di Markov con distribuzione di X come distr. stazionaria π . Per qualsiasi bounded $\psi: \mathbb{R} \rightarrow \mathbb{R}$ tale che $E_\pi[\psi(X)] < \infty$ si ha:

$$\frac{1}{N} \sum_{i=1}^N \psi(X^{(i)}) \xrightarrow[N \rightarrow \infty]{} E_\pi[\psi(X)]$$