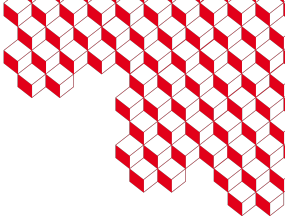

Méthodes de Monte-Carlo

M2 Radiophysique médicale, INSTN, 2024

Clément GAUCHY (clement.gauchy@cea.fr) Blog: clgch.github.io
CEA SACLAY

Sommaire

Les origines

- Le principe des méthodes Monte-Carlo est apparu au laboratoire de Los Alamos, à la fin des années 40
- Idée: Simuler la diffusion des neutrons dans un matériau fissile en utilisant de la simulation aléatoire
- Les méthodes MC sont désormais présentes dans tout les domaines impliquant de la simulation numérique: physique, finance, statistique,...

Figure 1: Stanislaw Ulam, mathématicien et fondateur des méthodes Monte-Carlo

Pourquoi Monte-Carlo ?

Figure 2: Casino de Monte-Carlo, Monaco

L'oncle de Stanislaw Ulam jouait beaucoup au casino de Monte-Carlo, où les jeux de hasard sont rois !

Les premières simulations Monte-Carlo étaient faites "à la main"...

Génération de nombres aléatoires

Comment simuler l'aléatoire avec un ordinateur ?

Génération de nombres aléatoires

Comment simuler l'aléatoire avec un ordinateur ?

On détermine une suite de nombres dans $[0, 1]$ dit pseudo-aléatoires.

Génération de nombres aléatoires

Comment simuler l'aléatoire avec un ordinateur ?

On détermine une suite de nombres dans $[0, 1]$ dit pseudo-aléatoires.

Exemple: Générateur congruentiel linéaire

$$z_{k+1} \equiv (az_k + c) \mod m \qquad x_{k+1} = \frac{z_{k+1}}{m-1}$$

On choisit des bons paramètres a, c, m pour "tromper" les test statistiques et génerer une loi uniforme $\mathcal{U}([0, 1])$. Le premier terme x_0 de la suite est appellé *seed*.

Génération de nombres aléatoires

Comment simuler l'aléatoire avec un ordinateur ?

On détermine une suite de nombres dans $[0, 1]$ dit pseudo-aléatoires.

Exemple: Générateur congruentiel linéaire

$$z_{k+1} \equiv (az_k + c) \mod m \qquad x_{k+1} = \frac{z_{k+1}}{m-1}$$

On choisit des bons paramètres a, c, m pour "tromper" les test statistiques et génerer une loi uniforme $\mathcal{U}([0, 1])$. Le premier terme x_0 de la suite est appellé *seed*.

La plupart des languages informatique utilisent des algorithmes plus sophistiqués comme Mersenne-Twister

Génération de réalisations d'une loi de probabilité arbitraire

Soit $X \sim P$, comment générer des réalisations de X à partir d'échantillon de $U \sim \mathcal{U}([0,1])$?

Génération de réalisations d'une loi de probabilité arbitraire

Soit $X \sim P$, comment générer des réalisations de X à partir d'échantillon de $U \sim \mathcal{U}([0,1])$?

Soit $F_X = \mathbb{P}(X \leq x)$ la fonction de répartition de X.

$$F_X(X) \sim \mathcal{U}([0,1])$$

On utilise alors la propriété $F_X^{-1}(U) \sim P$.

Méthode très efficace si on a une expression simple de F_X^{-1}

Génération de réalisations d'une loi de probabilité arbitraire

Méthode d'acceptation rejet de Von Neumann

Génération de réalisations d'une loi de probabilité arbitraire

Méthode d'acceptation rejet de Von Neumann

On veut échantilloner X de densité de proba f et on sait échantilloner Y de loi g tel que $f \leq M \times g$.

Génération de réalisations d'une loi de probabilité arbitraire

Méthode d'acceptation rejet de Von Neumann

On veut échantilloner X de densité de proba f et on sait échantilloner Y de loi g tel que $f \leq M \times g$.

- On simule $U \sim \mathcal{U}([0, 1])$
- On simule $Y \sim g$.
- Si $U < f(Y)/Mg(Y)$, alors on accepte le Y simulé comme un tirage selon f

Exemple: échantillonage du libre parcours d'une particule

Une particule se déplace dans un matériau, sa probabilité d'intéragir entre une distance x et $x + dx$ est

$$\Sigma dx$$

avec Σ la section efficace macroscopique (en m^{-1}).

Exemple: échantillonage du libre parcours d'une particule

Une particule se déplace dans un matériau, sa probabilité d'intéragir entre une distance x et $x + dx$ est

$$\Sigma dx$$

avec Σ la section efficace macroscopique (en m^{-1}).

On note $P(x)$ la probabilité que le particule ait atteint la distance x sans interactions.

$$\begin{aligned} P(x+dx) &= P(x)\mathbb{P}(\text{aucune interactions entre } [x, x+dx]) \quad \texttt{Hypothèse d'indépendance} \\ P(x+dx) &= P(x)(1-\Sigma dx) \\ \frac{dP}{dx} &= -P(x)\Sigma \end{aligned}$$

On a donc $\boxed{P(x) = \exp(-\Sigma x)}$

Exemple: échantillonage du libre parcours d'une particule

Probabilité de ne pas intéragir jusqu'à la distance puis d'intéragir en $x + dx$:

$$P(x)\Sigma dx = \underbrace{\Sigma \exp(-\Sigma x)}_{\texttt{densité de probabilité}} \quad dx$$

La fonction de répartition $F(x) = \int_0^x \Sigma \exp(-\Sigma s) ds = 1 - \exp(-\Sigma x)$ est facile à inverser !

$$1 - \exp(-\Sigma x) = u \iff x = \frac{-\ln(u)}{\Sigma}$$

Exemple: échantillonage du libre parcours d'une particule

Soit X la variable aléatoire du libre parcours d'une particule dans le matériau. Elle a pour densité $\Sigma \exp(-\Sigma x)$.

Exemple: échantillonage du libre parcours d'une particule

Soit X la variable aléatoire du libre parcours d'une particule dans le matériau. Elle a pour densité $\Sigma \exp(-\Sigma x)$.

Le libre parcours moyen ℓ est $\ell = \mathbb{E}[X] = \int_0^{+\infty} x\Sigma \exp(-\Sigma x) dx$.

Exemple: échantillonage du libre parcours d'une particule

Soit X la variable aléatoire du libre parcours d'une particule dans le matériau. Elle a pour densité $\Sigma \exp(-\Sigma x)$.

Le libre parcours moyen ℓ est $\ell = \mathbb{E}[X] = \int_0^{+\infty} x\Sigma \exp(-\Sigma x) dx$.

Á partir d'un échantillon $(X_i)_{1 \leq i \leq N}$ i.i.d génerés selon la loi de X, on peut utiliser la loi des grands nombres pour faire l'approximation suivante:

$$\ell \approx \frac{1}{N} \sum_{i=1}^{N} X_i$$

Exemple: échantillonage du libre parcours d'une particule

Soit X la variable aléatoire du libre parcours d'une particule dans le matériau. Elle a pour densité $\Sigma \exp(-\Sigma x)$.

Le libre parcours moyen ℓ est $\ell = \mathbb{E}[X] = \int_0^{+\infty} x\Sigma \exp(-\Sigma x) dx$.

Á partir d'un échantillon $(X_i)_{1 \leq i \leq N}$ i.i.d génerés selon la loi de X, on peut utiliser la loi des grands nombres pour faire l'approximation suivante:

$$\ell \approx \frac{1}{N} \sum_{i=1}^{N} X_i$$

Les méthodes Monte-Carlo peuvent servir a calculer des intégrales

Monte-Carlo pour la quadrature numérique

On cherche à calculer $I = \mathbb{E}[g(X)] = \int_{\mathbb{R}^d} g(x)f(x)dx$ avec f la densité de proba. de X

Monte-Carlo pour la quadrature numérique

On cherche à calculer $I = \mathbb{E}[g(X)] = \int_{\mathbb{R}^d} g(x)f(x)dx$ avec f la densité de proba. de X

Estimateur Monte-Carlo: la moyenne empirique à partir de N simulations $(X_i)_{1 \leq i \leq N}$ i.i.d. de même loi que X.

$$\widehat{I}_N = \frac{1}{N} \sum_{i=1}^{N} g(X_i)$$

Monte-Carlo pour la quadrature numérique

On cherche à calculer $I = \mathbb{E}[g(X)] = \int_{\mathbb{R}^d} g(x)f(x)dx$ avec f la densité de proba. de X

Estimateur Monte-Carlo: la moyenne empirique à partir de N simulations $(X_i)_{1 \leq i \leq N}$ i.i.d. de même loi que X.

$$\widehat{I}_N = \frac{1}{N} \sum_{i=1}^{N} g(X_i)$$

Propriétés:

- Estimateur sans biais $\mathbb{E}[\widehat{I}_N] = I$
- Convergence (dite "forte") asymptotique grâce à la lois des grands nombres: $\widehat{I}_N \xrightarrow[N \to +\infty]{} I$
- Variance de l'estimateur MC:

$$\mathrm{Var}(\widehat{I}_N) = \frac{1}{N} \mathrm{Var}(g(X))$$

Inconvénient: Convergence lente en $1/\sqrt{N}$
Avantage: Vitesse de convergence indépendante de la dimension d de X

Contrôle de l'erreur d'estimation

On utilise la notion d'intervalle de confiance pour contrôler l'erreur sur $\widehat{I_N}$.

Contrôle de l'erreur d'estimation

On utilise la notion d'intervalle de confiance pour contrôler l'erreur sur $\widehat{I_N}$.

Estimateur de la variance

$$S_N^2 = \frac{1}{N}\sum_{i=1}^{N}(g(X_i) - \widehat{I_N})^2$$

Contrôle de l'erreur d'estimation

On utilise la notion d'intervalle de confiance pour contrôler l'erreur sur $\widehat{I_N}$.

Estimateur de la variance

$$S_N^2 = \frac{1}{N}\sum_{i=1}^{N}(g(X_i) - \widehat{I_N})^2$$

Pour N petit:

$$\mathbb{P}(I \in [\widehat{I_N} - t_{N-1,\frac{1+\alpha}{2}}\frac{S_N}{\sqrt{N-1}}, \widehat{I_N} + t_{N-1,\frac{1+\alpha}{2}}\frac{S_N}{\sqrt{N-1}}]) \approx \alpha$$

Contrôle de l'erreur d'estimation

On utilise la notion d'intervalle de confiance pour contrôler l'erreur sur $\widehat{I_N}$.

Estimateur de la variance

$$S_N^2 = \frac{1}{N}\sum_{i=1}^{N}(g(X_i) - \widehat{I_N})^2$$

Pour N petit:

$$\mathbb{P}(I \in [\widehat{I_N} - t_{N-1,\frac{1+\alpha}{2}}\frac{S_N}{\sqrt{N-1}}, \widehat{I_N} + t_{N-1,\frac{1+\alpha}{2}}\frac{S_N}{\sqrt{N-1}}]) \approx \alpha$$

Pour N grand:

$$\mathbb{P}(I \in [\widehat{I_N} - u_{\frac{1+\alpha}{2}}\frac{S_N}{\sqrt{N-1}}, \widehat{I_N} + u_{\frac{1+\alpha}{2}}\frac{S_N}{\sqrt{N-1}}]) \approx \alpha$$

Contrôle de l'erreur d'estimation

On utilise la notion d'intervalle de confiance pour contrôler l'erreur sur $\widehat{I_N}$.

Estimateur de la variance

$$S_N^2 = \frac{1}{N}\sum_{i=1}^{N}(g(X_i) - \widehat{I_N})^2$$

Pour N petit:

$$\mathbb{P}(I \in [\widehat{I_N} - t_{N-1,\frac{1+\alpha}{2}}\frac{S_N}{\sqrt{N-1}}, \widehat{I_N} + t_{N-1,\frac{1+\alpha}{2}}\frac{S_N}{\sqrt{N-1}}]) \approx \alpha$$

Pour N grand:

$$\mathbb{P}(I \in [\widehat{I_N} - u_{\frac{1+\alpha}{2}}\frac{S_N}{\sqrt{N-1}}, \widehat{I_N} + u_{\frac{1+\alpha}{2}}\frac{S_N}{\sqrt{N-1}}]) \approx \alpha$$

C'est $\mathrm{Var}(\widehat{I_N})$ qui pilote la largeur de l'intervalle de confiance !

Réduction de variance

On rappelle que $\mathrm{Var}(\widehat{I_N}) = \frac{1}{N}\mathrm{Var}(g(X))$.

Réduction de variance

On rappelle que $\mathrm{Var}(\widehat{I_N}) = \frac{1}{N}\mathrm{Var}(g(X))$.

Contrôler la variance de l'intégrande $\iff$ Contrôler la précision de la méthode Monte-Carlo

Réduction de variance

On rappelle que $\mathrm{Var}(\widehat{I_N}) = \frac{1}{N}\mathrm{Var}(g(X))$.

Contrôler la variance de l'intégrande $\iff$ Contrôler la précision de la méthode Monte-Carlo

Il existe toute une variété de méthodes de réduction de variance:

- Échantillonage d'importance
- Stratification
- Variable de contrôle
- Conditionnement
- ...

Réduction de variance par variable de contrôle (*control variates*)

- Soit une fonction $h(X)$ appelée variable de contrôle dont on connait l'espérance $\mu = \mathbb{E}(h(X))$
- On définit la variable aléatoire en fonction d'une constante α :

$$W_\alpha(X) = g(X) + \alpha(h(X) - \mu) \rightarrow \mathbb{E}(W_\alpha(X)) = \mathbb{E}(g(X))$$

$$\widehat{I}_\alpha = \frac{1}{N}\sum_{i=1}^{N} W_\alpha(X_i)$$

- Le calcul de l'intégrale peut donc se faire sur la fonction $W_\alpha(X)$. Sa variance est :

$$\mathrm{Var}(W_\alpha(X)) \quad = \quad \mathrm{Var}(g(X)) + \alpha^2\mathrm{Var}(h(X)) + 2\alpha\mathrm{Cov}(g(X), h(X))$$

- Comme fonction de α, la variance de $W_\alpha(X)$ atteint son minimun pour la valeur :

$$\begin{aligned}
\alpha_{\mathrm{opt}} &= -\frac{\mathrm{Cov}(g(X), h(X))}{\mathrm{Var}(h(X))} \\
\mathrm{Var}(W_{\alpha_{\mathrm{opt}}}(X)) &= \mathrm{Var}(g(X)) - \underbrace{\frac{[\mathrm{Cov}(g(X), h(X))]^2}{\mathrm{Var}(h(X))}}_{\texttt{réduction de la variance}} \\
&= \mathrm{Var}(g(X))(1 - \rho^2_{g(X),h(X)})
\end{aligned}$$

  en notant $\rho_{g(X),h(X)}$ le coefficient de corrélation entre les variables $g(X)$ et $h(X)$

- Intérêt de choisir une variable de controle la plus corrélée à $g(X)$ (pas toujours évident)

Exemple d'utilisation d'une variable de contrôle

- Calcul de $I = \int_0^1 g(x)dx$
- $g(x) = 1/(1+x) \rightarrow I = \ln(2)$
- Par tirages MC d'une loi uniforme $X \sim \mathcal{U}(0,1)$:

$$\widehat{I}_N = (1/n) \sum_{i=1}^{N} \frac{1}{1+X_i}$$

- On prend comme variable de contrôle $h(X) = 1 + X$, $\mu = 3/2$
- On peut calculer $\rho_{g(X),h(X)} \approx 0.6$

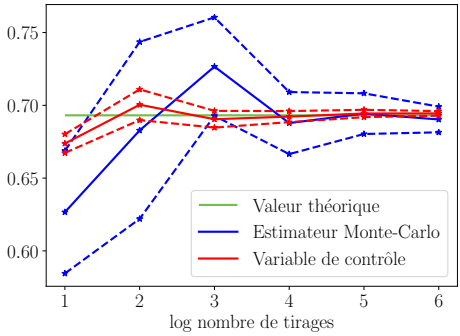

Figure 3: Les courbes en pointillés correspondent à l'intervalle de confiance à 95%

- Dans la figure, les intervalles de confiance asymptotique ont été estimés (TCL : $\bar{G}_n \rightarrow$ loi normale)

Échantillonage d'importance (*Importance sampling*)

- Calcul de l'intégrale $I = \int_D g(x)f(x)dx$ où $x \in \mathbb{R}^d$ et $g(x)$ une fonction de $D \subset \mathbb{R}^d$ dans $\mathbb{R}$ et f une certaine densité de probabilité
- La représentation de I comme une espérance n'est pas unique:

$$I = \int_D g(x)f(x)dx = \int_D \frac{g(x)f(x)}{h(x)} h(x)dx = \mathbb{E}_{X\sim h}\left[\frac{g(x)f(x)}{h(x)}\right]$$

- Idée: On peut biaiser l'échantillonage en simulant X selon g pour rendre plus probable les réalisations "importantes".
- On propose l'estimateur suivant:

$$\widehat{I_n} = \frac{1}{N}\sum_{i=1}^{N} g(X_i)\frac{f(X_i)}{h(X_i)}$$

avec $f(x)/h(x)$ appellé le rapport de vraisemblance

Echantillonage d'importance (2)

- L'estimateur est non biaisé:

$$\mathbb{E}[\widehat{I_N}] = \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}_h\left[g(X_i)\frac{f(X_i)}{h(X_i)}\right] = \mathbb{E}_h\left[g(X)\frac{f(X)}{h(X)}\right] = \int_D \frac{g(x)f(x)}{h(x)}h(x)dx = \mathrm{I}$$

- Convergence de l'estimateur (par la loi forte des grand nombres):

$$\widehat{I_N} \xrightarrow[N\to+\infty]{} I$$

- La variance de l'estimateur s'écrit:

$$\mathrm{Var}(\widehat{I_N}) = \frac{1}{N}\mathrm{Var}_h\left(g(X)\frac{f(X)}{h(X)}\right) = \frac{1}{N}\left(\mathbb{E}_{X\sim f}\left[g(X)^2\frac{f(X)}{h(X)}\right] - I^2\right)$$

Le choix astucieux de h peut réduire drastiquement la variance !

Échantillonage d'importance optimal

La meilleur distribution de probabilité h est celle minimisant $\mathrm{Var}(\widehat{I_N}) \hookrightarrow$ Soit h^* la meilleur distribution, alors

$$h^* \in \underset{h}{\operatorname{argmin}}\, \mathbb{E}_{X\sim f}\left[g(X)^2 \frac{f(X)}{h(X)}\right] = \int_D g(x)^2 \frac{f^2(x)}{h(x)} dx$$

Échantillonage d'importance optimal

La meilleur distribution de probabilité h est celle minimisant $\mathrm{Var}(\widehat{I_N}) \hookrightarrow$ Soit h^* la meilleur distribution, alors

$$h^* \in \underset{h}{\operatorname{argmin}}\, \mathbb{E}_{X\sim f}\left[g(X)^2 \frac{f(X)}{h(X)}\right] = \int_D g(x)^2 \frac{f^2(x)}{h(x)} dx$$

La solution de ce problème de minimisation est:

$$h^*(x) = \frac{g(x)f(x)}{\int_D g(u)f(u)du}$$

Échantillonage d'importance optimal

La meilleur distribution de probabilité h est celle minimisant $\mathrm{Var}(\widehat{I_N}) \hookrightarrow$ Soit h^* la meilleur distribution, alors

$$h^* \in \underset{h}{\operatorname{argmin}}\, \mathbb{E}_{X\sim f}\left[g(X)^2 \frac{f(X)}{h(X)}\right] = \int_D g(x)^2 \frac{f^2(x)}{h(x)} dx$$

La solution de ce problème de minimisation est:

$$h^*(x) = \frac{g(x)f(x)}{\int_D g(u)f(u)du}$$

On peut remarquer que $\mathrm{Var}_{h^*}\left(g(X)\frac{f(X)}{h^*(X)}\right) = 0$!

Échantillonage d'importance optimal

La meilleur distribution de probabilité h est celle minimisant $\mathrm{Var}(\widehat{I_N}) \hookrightarrow$ Soit h^* la meilleur distribution, alors

$$h^* \in \underset{h}{\mathrm{argmin}}\, \mathbb{E}_{X\sim f}\left[g(X)^2 \frac{f(X)}{h(X)}\right] = \int_D g(x)^2 \frac{f^2(x)}{h(x)} dx$$

La solution de ce problème de minimisation est:

$$h^*(x) = \frac{g(x)f(x)}{\int_D g(u)f(u)du}$$

On peut remarquer que $\mathrm{Var}_{h^*}\left(g(X)\frac{f(X)}{h^*(X)}\right) = 0$!

⚠ Le dénominateur de h^* est... $I = \int_D g(x)f(x)dx$ la quantité que l'on cherche à estimer ! Cette loi n'est pas utile en pratique, mais on peut chercher à l'approcher par une famille paramétrique de lois $\{h_\theta, \theta \in \Theta\}$.

$$\theta_* \in \underset{\theta\in\Theta}{\mathrm{argmin}}\, D(h^*, h_\theta)$$

Exemple de réduction de variance par échantillonnage d'importance

- $g(x) = 3x^2$ et intégrale $I = \int_0^1 g(x)dx = 1$
- Choix uniforme $U(0,1)$, $f(x) = 1_{[0,1]}(x)$

$$\mathrm{Var}(g(X)) \quad = \quad \int_0^1 (3x^2 - 1)^2 dx = \frac{4}{5}$$

- Choix plus astucieux par échantillonnage d'importance : $h(x) = 2x\, 1_{[0,1]}(x)$.

$$\frac{g(x)f(x)}{h(x)} \quad = \quad \frac{3x}{2}$$

$$\mathrm{Var}_f\left(\frac{g(X)f(X)}{h(X)}\right) \quad = \quad \int_0^1 (\frac{3x}{2} - 1)^2 2x dx = \text{。}$$

- La variance a été divisé d'un facteur 6

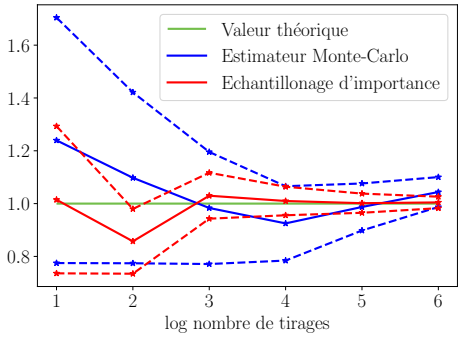

Figure 4: Les courbes en pointillés correspondent à l'intervalle de confiance à 95%

Sommaire

Définition d'une chaîne de Markov

Définition

Une chaîne de Markov est de suite de variables aléatoires $X_0, X_1, \ldots, X_t$ toutes définis sur le même espace possédant la **propriété de Markov**

$$P(X_t = x_t | X_0 = x_0, \ldots, X_{t-1} = x_{t-1}) = P(X_t = x_t | X_{t-1} = x_{t-1})$$

Elle est determinée par deux paramètres:

- sa distribution initiale $P(X_0)$
- son noyau de transition $K(x, A) = P(X_t \in A | X_{t-1} = x)$

On considère dans ce cours des chaînes de Markov **homogène**

$$P(X_{t+1} = x | X_t = y) = P(X_t = x | X_{t-1} = y)$$

Propriétés des chaînes de Markov

Propriété:

Une chaîne de Markov est dite **irréductible** si tout les ensembles de probabilité non nulles peuvent être atteint à partir de tout point de départ (tout état est accessible à partir de n'importe quelle autre).

Propriété:

Une chaîne de Markov est dite **récurrente** si tout les trajectoires X_t passent une infinité de fois dans tout ensemble de probabilité non nulle de l'espace d'état

Propriété:

Une chaîne de Markov est dite **apériodique** s'il n'y a pas de comportement périodique.

Loi stationnaire et théorème ergodique

Définition

Une distribution de probabilités P_* est appellé **loi invariante** (ou stationnaire) pour une chaîne de Markov si elle vérifie

$$X_t \sim P_* \implies \forall j > t,\ X_{t+j} \sim P_* \ .$$

Remarque: Une châine de Markov peut admettre plusieurs lois stationnaires.

Théorème ergodique

Une chaîne de Markov irréductible et récurrente positive (le temps de retour moyen est fini) admet une unique loi de probabilité invariante P_*. Si cette chaîne de Markov est de plus apériodique, alors elle converge en loi vers P_*.

Introduction aux méthodes MCMC

- Les MCMC, Méthodes de Monte-Carlo par Chaînes de Markov permettent de simuler numériquement un grand nombre de distributions pour lequelles la densité de probabilité est connue à une constante près.
- Les cas où la loi d'échantillonnage n'est connue qu'au facteur de normalisation près :
  - Physique statistique: la densité de probabilité de trouver le systême dans l'état $\mathbf{x}$ d'énergie $E(\mathbf{x})$ décrit par la distribution de Boltzmann: température T et k_B la constante de Boltzmann :

$$f(\mathbf{x}) \propto \exp\left[-E(\mathbf{x})/k_B T\right]$$

    où le facteur de normalisation est la fonction de partition $Z = \int \exp\left[-E(\mathbf{x})/k_B T\right] d\mathbf{x}$
  - Inférence Bayésienne : loi *a posteriori*

$$\underbrace{\pi(\mathbf{x}|\texttt{observations})}_{\texttt{loi } a \textit{ posteriori}} \propto \underbrace{f(\texttt{observations}|\mathbf{x})}_{\texttt{vraisemblance}} \times \underbrace{\pi(\mathbf{x})}_{\texttt{loi } a \textit{ priori}}$$

- Les MCMC permettent donc d'échantillonner une loi afin d'estimer des grandeurs d'intérêt comme l'espérance, la variance ou un taux d'évènements

Algorithmes MCMC: principe général

Objectif: Approximer une intégrale d'une loi de probabilité cible. $\Longrightarrow$ Générer une chaîne de Markov dont la loi stationnaire est la loi cible (distribution de Boltzmann en Φ statistique, loi *a posteriori* en statistique Bayésienne) puis d'appliquer une estimation Monte Carlo.

Nécessite une **double convergence**:

1) Convergence de la chaîne de Markov vers sa loi stationnaire: $X_n \xrightarrow[n\to+\infty]{\mathcal{L}} P_*$
2) Convergence de la méthode de Monte Carlo $1/N \sum_{i=1}^{N} f(X_i) \to \mathbb{E}_{X\sim P_*}[f(X)]$

Schéma général des algorithmes MCMC

Les algorithmes MCMC s'appuient sur une approche acceptation-rejet

Algorithm 1 Algorithme MCMC

Require: $x_0,\ N$
Ensure: $t \leftarrow 0$
while $t < N$ **do**
  Proposer un nouveau candidat $y_t \sim q(y_t|x_{t-1})$
  $U \sim \mathcal{U}([0,1])$
  if $U < \alpha(x_{t-1}, y_t)$ **then**
    $x_t \leftarrow y_t$
  end if
  $t \leftarrow t+1$
end while

Choix de la loi instrumentale

Pas de choix optimal pour la loi q.

Afin de garantir la convergence vers la loi cible P_* :

- Le support de q soit contenir le support de P_*
- q ne doit pas générer de valeurs périodiques

Algorithme de Metropolis-Hastings

Objectif : échantillonner selon une loi de proba de densité $f(\mathbf{x})$.

Algorithm 2 Algorithme de Metropolis-Hastings

Require: Condition initiale X_0, loi instrumentale $q(\mathbf{x} \to \mathbf{y})$. On pose $t = 0$ et N la taille de la chaîne

1: **while** $t < N$ **do**
2: Tirage aléatoire y_t avec la loi $q(y_t|x_{t-1})$
3:

$$\alpha(x_{t-1}, y_t) = \min\left[1, \frac{f(y_t)q(y_t|x_{t-1})}{f(x_{t-1})q(x_{t-1}|y_t)}\right]$$

4: $U \sim \mathcal{U}([0,1])$
5: **if** $U < \alpha(x_{t-1}, y_t)$ **then**
6: $X_t = y_t$ (toujours vrai si $\alpha(x_{t-1}, y_t) = 1$)
7: **else**
8: $X_t = x_t$
9: **end if**
10: $t = t + 1$
11: **end while**

Algorithme de Metropolis Hastings

$$\alpha(x_{t-1}, y_t) = \min\left[1, \frac{f(y_t)q(y_t|x_{t-1})}{f(x_{t-1})q(x_{t-1}|y_t)}\right]$$

Calculable en connaissant f à une constante multiplicative près ! $\Longrightarrow$ Fort intérêt pour la statistique Bayésienne. On peut parfois simplifier $\alpha(x_{t-1}, y_t)$

- Metropolis-Hastings indépendant: $q(y_t|x_{t-1}) = q(y_t)$
- Metropolis-Hastings à marche aléatoire: $q(y_t|x_{t-1}) = g(y_t - x_{t-1})$. Si g est symétrique alors

Algorithme de Metropolis Hastings

$$\alpha(x_{t-1}, y_t) = \min\left[1, \frac{f(y_t)q(y_t|x_{t-1})}{f(x_{t-1})q(x_{t-1}|y_t)}\right]$$

Calculable en connaissant f à une constante multiplicative près ! $\Longrightarrow$ Fort intérêt pour la statistique Bayésienne. On peut parfois simplifier $\alpha(x_{t-1}, y_t)$

- Metropolis-Hastings indépendant: $q(y_t|x_{t-1}) = q(y_t)$
- Metropolis-Hastings à marche aléatoire: $q(y_t|x_{t-1}) = g(y_t - x_{t-1})$. Si g est symétrique alors

$$\frac{f(y_t)q(y_t|x_{t-1})}{f(x_{t-1})q(x_{t-1}|y_t)} = \frac{f(y_t)g(y_t - x_{t-1})}{f(x_{t-1})g(y_t - x_{t-1})} = \frac{f(y_t)}{f(x_{t-1})}$$

Metropolis-Hastings, pour ou contre ?

Avantages:

- Très simple & très général
- Permet l'échantillonnage selon une grande variété de distributions de probabilité

Metropolis-Hastings, pour ou contre ?

Avantages:

- Très simple & très général
- Permet l'échantillonnage selon une grande variété de distributions de probabilité

Inconvénients:

- Le choix du *proposal* est crucial, c'est le degré de liberté principal de l'algorithme
- Fléau de la dimension
- Seulement des heuristiques pour vérifier la convergence de la chaîne de Markov vers sa distribution stationnaire

Échantilloneur de Gibbs

Dimension $\nearrow \implies$ Effondrement de la probabilité d'acceptation

Échantilloneur de Gibbs: réactualisation coordonée par coordonée en conditionnant sur les dernières valeurs obtenus (⚠ pas d'acceptation-rejet !)

Algorithm 3 Algorithme de l'échantilloneur de Gibbs

Require: $x^{(0)} = (x_1^{(0)}, \dots, x_d^{(0)}),\ N$

while $t < N$ **do**
  Simuler $x_1^{(t)} \sim P(x_1 | x_2^{(t-1)}, \dots, x_d^{(t-1)})$
  Simuler $x_2^{(t)} \sim P(x_2 | x_1^{(t-1)}, x_3^{(t-1)} \dots, x_d^{(t-1)})$
  ...
  Simuler $x_i^{(t)} \sim P(x_i | x_1^{(t-1)}, \dots, x_{i-1}^{(t-1)}, x_{i+1}^{(t-1)}, \dots, x_d^{(t-1)})$
  ...
  $x_d^{(t)} \sim P(x_d | x_2^{(t-1)}, \dots, x_{d-1}^{(t-1)})$
end while

Échantilloneur de Gibbs

Dimension $\nearrow \implies$ Effondrement de la probabilité d'acceptation

Échantilloneur de Gibbs: réactualisation coordonée par coordonée en conditionnant sur les dernières valeurs obtenus (⚠ pas d'acceptation-rejet !)

Algorithm 4 Algorithme de l'échantilloneur de Gibbs

Require: $x^{(0)} = (x_1^{(0)}, \dots, x_d^{(0)}),\ N$

1: **while** $t < N$ **do**
2: Simuler $x_1^{(t)} \sim P(x_1 | x_2^{(t-1)}, \dots, x_d^{(t-1)})$
3: Simuler $x_2^{(t)} \sim P(x_2 | x_1^{(t-1)}, x_3^{(t-1)} \dots, x_d^{(t-1)})$
4: ...
5: Simuler $x_i^{(t)} \sim P(x_i | x_1^{(t-1)}, \dots, x_{i-1}^{(t-1)}, x_{i+1}^{(t-1)}, \dots, x_d^{(t-1)})$
6: ...
7: $x_d^{(t)} \sim P(x_d | x_2^{(t-1)}, \dots, x_{d-1}^{(t-1)})$
8: **end while**

Nota Bene: Si les lois conditionelles sont inconnues, on peut introduire un échantillonage par Metropolis-Hastings pour chaque lois (*Metropolis within Gibbs*)

Convergence des MCMC

L'objectif du MCMC est d'échantillonner selon f connue à une constante multiplicative près

Convergence des MCMC

L'objectif du MCMC est d'échantillonner selon f connue à une constante multiplicative près

Aucune garantie de la convergence de la chaîne en temps fini !

Convergence des MCMC

L'objectif du MCMC est d'échantillonner selon f connue à une constante multiplicative près

Aucune garantie de la convergence de la chaîne en temps fini !

Il faut étudier la convergence effective à chaque analyse

Trace

Échantillonage par Metropolis Hastings de $f(x) \propto 0.4 \exp((x/2)^2) + 0.6 \exp(((x-2)/2)^2)$

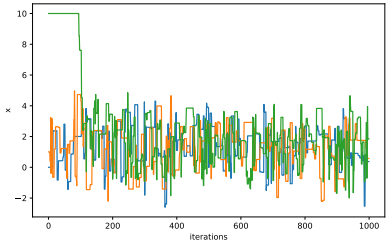

Figure 5: *Trace plot*

$\rightarrow$ Les chaînes doivent se superposer et se confondre. Ici problème de *burn in* $\Longrightarrow$ élagage de la chaîne.

Auto-corrélation

L'auto-corrélation d'un procesuss stochastique X_t est la corrélation du processus par rapport à une version décalée de lui même. Pour un processus stationnaire d'espérance μ et de variance σ^2 elle est définit par:

$$\rho(k) = \frac{1}{\sigma^2}\mathbb{E}[(X_t - \mu)(X_{t+k} - \mu)] \ ,$$

pour un décalage (*lag*) k fixé.

L'estimateur empirique de l'autocorrélation pour un échantillon $(X_t)_{1 \leq t \leq k}$ est défini par:

$$\widehat{\rho}(k) = \frac{1}{(n-k)\sigma^2}\sum_{t=1}^{n-k}(X_t - \mu)(X_{t+k} - \mu)$$

auto-corrélation

```
plt.acorr(chain, maxlag=50)
```

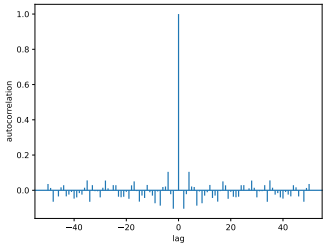

Figure 6: Bon autocorrélogramme

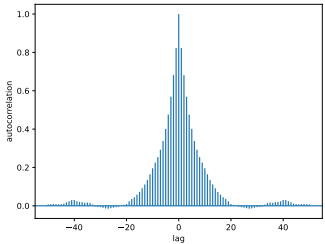

Figure 7: Mauvais autocorrélogramme

Autocorrélagramme d'une chaîne (à droite) et de la même chaîne après *thinning*.

Taille d'échantillon effective

La propriété de Markov induit de l'**auto-corrélation** entre les valeurs générées à la suite les unes des autres $\implies$ ralentit la convergence de la loi des grands nombres.

On peut quantifier cette perte via la **Taille d'échantillon effectizve**

$$ESS = \frac{N}{1 + 2\sum_{k=1}^{+\infty} \rho(k)}$$

où $\rho(k)$ désigne l'autocorrélation de *lag* k.

On peut utiliser la librairie Python `statsmodels` pour calculer ρ

Logiciels MCMC

BUGS: *Bayesian inference Using Gibbs Sampling*, logiciel permettant l'analyse de modèles statistique complexes. `JAGS` est une interface en ligne de commande interfacé avec R.
`https://mcmc-jags.sourceforge.io/`

Stan: `http://mc-stan.org` est un language de programation impératif crée en 2012 qui permet de définir une log densité de probabilité sur des paramètres conditionés à des constantes et des données. Il est basé sur des algorithmes MCMC très avancées (*Hamiltonian Monte Carlo*).

$\hookrightarrow$ Présentation du logiciel `Stan` dans la suite.

Stan

Exemple d'un fichier Stan:

bernouilli.stan

```
data {
        int<lower=0> N;
        array[N] int<lower=0,upper=1> y;
}
parameters {
        real<lower=0,upper=1> theta;
}
model {
        theta ~ beta(1,1);  // uniform prior on interval 0,1
        y ~ bernoulli(theta);
}
```

Langage fortement typé (comme le C++). Le fichier `.stan` est ensuite compilé à l'aide de l'interface en ligne de commande `CmdStan` (`https://mc-stan.org/docs/cmdstan-guide/`)

Stan

Les données sont stockées dans un fichier `.json`:

`bernouilli.data.json`

```
{
        "N" : 10,
        "y" : [0,1,0,0,0,0,0,0,0,1]
}
```

La compilation de `bernouilli.stan` produit un exécutable `bernouilli` (ou `bernouilli.exe` sur Windows). Qui peut être appellé avec le flag

- `sample` pour effectuer un échantillonage MCMC des paramètres
- `optimize` pour effectuer une optimisation permettant d'obtenir l'EMV ou le MAP

Références

- The beginning of the Monte Carlo method, N. Metropolis, *Los Alamos Science* special issue, 1987
- Exemple de simulation MCMC
  `https://chi-feng.github.io/mcmc-demo/app.html?algorithm=RandomWalkMH&target=banana`