

Algorithme d'Hasting Metropolis - Problème du voyageur de commerce

PEROTTINO Tony, VAILLANT Corentin,
LE BER Tom, BERNARD Léo

21 février 2025

Table des matières

1	Introduction	3
1.1	Informations générales et description historique	3
1.2	Pourquoi HM permet-il de résoudre le problème du voyageur de commerce ?	4
1.3	Objectif de ce rapport	5
2	Chaines de Markov à états finis	6
2.1	Définitions	6
2.2	Propriétés fondamentales	6
2.2.1	Représentation sous forme de Graphe Orienté	7
2.2.2	Matrice de Transition en k -pas	8
2.3	Classes de communication et d'équivalence	9
2.3.1	Les différents types d'états	9
2.3.2	Classes de communication	10
2.3.3	Classes d'équivalence	11
2.3.4	Exemples	12
2.4	Loi stationnaire	12
2.5	Comportement en temps long	12
3	Fonctionnement de l'Algorithme de Hasting-Metropolis	14
3.1	Qu'est-ce qu'une méthode dite MCMC ?	14
3.2	Définitions	15
3.3	Description des étapes de l'algorithme	15
3.4	Preuve de l'algorithme	15
3.5	Exemple d'application	16
3.6	Comparaison avec d'autres méthodes d'échantillonnage	17
4	Conclusion	18
5	Sources	19

Préambule

Ce rapport s'inscrit dans le cadre de la matière "Projet de Mathématiques" de l'université de Toulouse (Paul Sabatier).

L'objectif de ce rapport est de présenter et de prouver l'algorithme d'Hastings-Metropolis. À la fois d'expliquer en détail son fonctionnement, mais aussi de comprendre l'intérêt qu'il présente dans les sciences et en quoi il prend sa valeur. Pour illustrer l'algorithme, nous nous intéresserons au problème du voyageur de commerce et étudierons sa complexité.

1 Introduction

Afin de comprendre en profondeur ce sujet, il est nécessaire de s'en faire une idée globale, même naïve, pour pouvoir suivre convenablement la ligne directrice de notre discours.

1.1 Informations générales et description historique

L'algorithme d'Hastings-Metropolis consiste en une méthode d'échantillonnage stochastique permettant, à partir d'une distribution de probabilité donnée, de pouvoir décrire son comportement et d'obtenir des statistiques dessus. Cet algorithme prend sa valeur quand la distribution est difficile à analyser (systèmes multidimensionnels, par exemple). Cette méthode est marquante, car elle a été conceptualisée tôt dans l'histoire de l'informatique et a mis des décennies avant d'être prouvée et expliquée entièrement.

C'est en 1949 que l'écriture de l'algorithme a été publiée dans un article de Nicholas Metropolis et Stanislaw Ulam. La paternité de l'algorithme est soumise à débat, car l'algorithme s'inscrit sous le nom de son chef de projet (Metropolis), alors que l'équipe composée de Nicholas Metropolis, Arianna et Marshall Rosenbluth, Augusta et Edward Teller a contribué à cette méthode. Ils étudiaient alors plus particulièrement le cas de la distribution de Boltzmann, une des distributions les plus utilisées en physique statistique, dans des travaux datant de 1953.

Cela illustre une dynamique fréquente dans les sciences, où le mérite est disproportionnellement attribué à une personne alors qu'il s'agit des efforts de toute une équipe. En particulier, Arianna Rosenbluth était considérée comme brillante par le monde scientifique.

https://fr.wikipedia.org/wiki/Algorithme_de_Metropolis-Hastings#cite_note-2
<https://www.radcliffe.harvard.edu/news-and-ideas/flash-of-genius>

En 1970, W. K. Hastings (1930-2016) a étendu l'algorithme au cas d'une distribution quelconque, et c'est cette version généralisée qui est connue sous le

nom d'algorithme de Metropolis-Hastings. Cette extension a eu de nombreuses applications dans divers domaines scientifiques, comme en statistique bayésienne (espaces complexes multidimensionnels), en biologie computationnelle (analyse des séquences génétiques), en économie et en finance (modèles stochastiques MCMC en général), etc.

Quant à lui, le problème du voyageur de commerce (dit "TSP", comme Travelling Salesman Problem en anglais) est un problème classique et bien connu pour être un problème NP-difficile et NP-complet, ce qui signifie qu'il est extrêmement difficile de trouver une solution optimale en temps polynomial, et il est peu probable que des algorithmes en temps polynomial existent pour résoudre ce problème de manière exacte.

L'origine du problème est assez incertaine : il a été formulé pour la première fois vers 1850 dans un manuel d'un commerçant voyageant en Suisse et en Allemagne. Ce n'est que dans les années 1930 que le problème fut énoncé d'abord comme un casse-tête (par William Rowan et Thomas Kirkman), puis étudié (par, entre autres, Thomas Kirkman, Jillian Beardwood, J. H. Halton et John Hammersley).

Le problème consiste à trouver le chemin le plus court entre des points donnés en connaissant la distance entre eux. Il faut alors tâcher de commencer et terminer par le même point (recherche d'un cycle hamiltonien le plus court). Les distances peuvent être dites symétriques ou asymétriques, c'est-à-dire que la distance entre eux varie en fonction de la direction du déplacement. On peut illustrer ce problème grâce à un voyageur de commerce devant vendre ses produits dans chacune des villes en un minimum de temps. Ce problème peut donc être naturellement représenté par un graphe.

https://fr.wikipedia.org/wiki/Probl%C3%A8me_du_voyageur_de_commerce

<https://www.cambridge.org/core/journals/mathematical-proceedings-of-the-cambridge-philosophical-society/article/abs/shortest-path-through-many-points/F1C28B5730B94887F4659FCBF8A1F2BB>

1.2 Pourquoi HM permet-il de résoudre le problème du voyageur de commerce ?

Le principe mathématique derrière HM repose sur la construction dynamique d'une chaîne de Markov, qui n'est pas connue au préalable mais se développe au fil des itérations. À mesure que l'algorithme progresse, le comportement de cette chaîne converge vers la distribution cible, permettant d'obtenir un échantillon fiable par rapport aux états de la distribution.

Quant au problème du voyageur de commerce, il peut être représenté par un graphe orienté ayant un sommet pour chaque ville et une arête pour chaque temps de trajet. Nous verrons par la suite qu'une chaîne de Markov peut être associée à un graphe orienté, c'est-à-dire que le problème du voyageur de

commerce est résoluble par HM.

De plus, nous avons précisé au préalable que le voyageur de commerce était NP-difficile et NP-complet. En conséquence, l'objectif de HM n'est pas d'apporter la réponse exacte au problème, mais une approximation fidèle de la solution, ce qui peut sembler contre-intuitif. Cette approximation est la caractéristique principale des méthodes MCMC que nous aborderons dans leur partie dédiée.

Plus généralement, il faut donc comprendre que HM est un outil adapté à la résolution de problèmes ayant un trop grand nombre de possibilités pour être explorées toutes dans un temps raisonnable. C'est pour cela que la résolution de ces problèmes se fait par l'approximation de la bonne solution (en étudiant donc des échantillons de toutes les possibilités).

Cette nuance est très importante, car elle révèle en quoi HM est un outil sophistiqué.

1.3 Objectif de ce rapport

Il est donc compréhensible que, pour prouver HM, nous allons devoir tout d'abord comprendre le fonctionnement des chaînes de Markov et en expliciter les définitions fondamentales, puis leur comportement sur le temps long, pour pouvoir ensuite construire la preuve mathématique derrière HM et déterminer les raisons de son fonctionnement. Nous étudierons à la fois cette preuve et ferons un détour par les méthodes MCMC (Monte-Carlo), dont HM est issu. Nous finirons par conclure ce rapport avec une implémentation personnelle de HM pour illustrer nos propos.

2 Chaines de Markov à états finis

2.1 Définitions

Chaîne de Markov

Soit E un ensemble fini ou dénombrable.

Soit $(X_n)_{n \in \mathbb{N}}$ une suite de variables aléatoires à valeurs dans E .

$(X_n)_{n \in \mathbb{N}}$ est appelée **chaîne de Markov** si et seulement si :

(1) Sa loi de probabilité initiale X_0 est bien définie.

(2) Et si elle respecte la **propriété de Markov**, définie ci-dessous :

$$\forall n \geq 0, \quad \exists x_1, x_2, \dots, x_{n+1} \in E,$$

$$\mathbb{P}(X_{n+1} = x_{n+1} \mid X_1 = x_1, \dots, X_n = x_n) = \mathbb{P}(X_{n+1} = x_{n+1} \mid X_n = x_n)$$

E est alors appelé l'espace d'états de $(X_n)_{n \in \mathbb{N}}$.

En d'autres termes, la loi de probabilité \mathbb{P} régissant la transition de l'état X_n vers l'état X_{n+1} dépend uniquement du dernier terme X_n , et reste totalement indépendante des tous ses états antérieurs $\{X_0, X_1, \dots, X_{n-1}\}$.

Cette propriété, que l'on peut qualifier de « sans mémoire » ou de propriété de Markov, constitue la caractéristique fondamentale de ces processus stochastiques.

2.2 Propriétés fondamentales

Matrice de transition

Soit $(X_n)_{n \in \mathbb{N}}$ une chaîne de Markov à valeurs dans un ensemble fini ou dénombrable E .

La **matrice de transition** associée à $(X_n)_{n \in \mathbb{N}}$ est la famille de nombres réels $\{P_{ij}\}_{i,j \in E}$ définie par :

$$\forall n \geq 0, \forall i, j \in E, \quad P_{ij} = \mathbb{P}(X_{n+1} = j \mid X_n = i).$$

On regroupe cette famille de probabilités sous forme d'une matrice de transition P où chaque élément P_{ij} représente la probabilité de passer de l'état i à l'état j en une seule étape. Lorsque l'ensemble des états E est fini, par exemple $E = \{1, 2, \dots, N\}$, la matrice s'exprime sous la forme suivante :

$$P = \begin{bmatrix} P_{11} & P_{12} & \cdots & P_{1N} \\ P_{21} & P_{22} & \cdots & P_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ P_{N1} & P_{N2} & \cdots & P_{NN} \end{bmatrix}$$

Chaque ligne de P représente les probabilités de transition depuis un état fixé i vers tous les autres états du système. Ces probabilités sont normalisées, c'est à dire :

$$\forall i \in \{1, \dots, N\}, \quad \sum_{j=1}^N P_{ij} = 1.$$

Autrement dit, la somme des probabilités de transition depuis chaque état i fixé vers tous les autres états vaut 1.

De manière analogue, si l'espace des états est infini dénombrable (par exemple $E = \{1, 2, 3, \dots\}$), on indexe les états de la même façon et la condition

$$\forall i \in E, \quad \sum_{j \in E} P_{ij} = 1,$$

reste valable.

La matrice P est ainsi dite *stochastique (par lignes)* car chacun de ses éléments satisfait $0 \leq P_{ij} \leq 1$ et la somme des éléments de chaque ligne vaut 1.

2.2.1 Représentation sous forme de Graphe Orienté

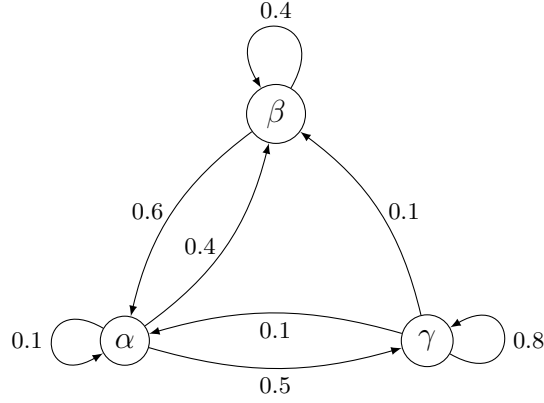
Il est relativement naturel de représenter la matrice de transition d'une chaîne de Markov par un graphe orienté. Ceci qui permet de visualiser les probabilités de transition entre les états.

Par exemple, voici le cas d'une chaîne de Markov disposant de 3 états ; α, β et γ , ordonnés respectivement, dans la matrice de transition suivante :

$$P = \begin{bmatrix} 0,1 & 0,4 & 0,5 \\ 0,6 & 0,4 & 0 \\ 0,1 & 0,1 & 0,8 \end{bmatrix}$$

Nous avons donc par exemple $P_{\alpha,\beta} = P_{1,2} = \mathcal{P}_{\alpha}(\beta) = 0,4$.

Nous obtenons ainsi le graphe suivant :



2.2.2 Matrice de Transition en k -pas

Dans une chaîne de Markov, la matrice de transition en un pas, que nous avons noté P , décrit pour tout couple d'états $i, j \in \{1, 2, \dots, N\}$ la probabilité de passer du premier état au deuxième via la valeur de l'élément $P_{i,j}$. Chaque élément $P_{i,j}$ représente donc la probabilité de passer de l'état i à l'état j en une seule étape, ce qui se traduit mathématiquement par :

$$\mathbb{P}(X_1 = j \mid X_0 = i)$$

Où X_t désigne l'état de la chaîne à l'instant t .

Cependant, il peut être nécessaire de chercher à déterminer la probabilité de passer de l'état i à l'état j après exactement k étapes. Pour cela, on utilise la matrice de transition en k -pas, notée P^k , pour laquelle chaque élément $P_{i,j}^{(k)}$ représente la probabilité de passer de l'état i à l'état j au bout d'exactly k étapes. Plus formellement :

$$P_{i,j}^{(k)} = \mathbb{P}(X_k = j \mid X_0 = i).$$

Cette matrice P^k est obtenue en élevant la matrice de transition P à la puissance k , ce qui revient à effectuer $k - 1$ produits matriciels successifs :

$$P^k = \underbrace{P \cdot P \cdot \dots \cdot P}_{(k \text{ fois})}.$$

Nous allons démontrer ce résultat en utilisant la propriété de Markov ainsi que le produit matriciel. La relation ci-dessus repose sur la propriété fondamentale des chaînes de Markov ; pour chaque étape, la probabilité de transition d'un état i à un état j dépend uniquement du dernier pas réalisé, et non de tous les précédents. En d'autres termes :

$$\mathbb{P}(X_{k+1} = j \mid X_0 = i) = \sum_{n=1}^N \mathbb{P}(X_{k+1} = j \mid X_k = n) \cdot \mathbb{P}(X_k = n \mid X_0 = i).$$

En termes matriciels, cette équation se traduit donc par :

$$P_{i,j}^{(k+1)} = \sum_{n=1}^N P_{i,n}^{(k)} \cdot P_{n,j}.$$

Cela correspond précisément à la règle du produit matriciel, ce qui montre en conséquence que $P^{n+1} = P \cdot P^n$. Par récurrence, on en déduit donc que :

$$P^k = \underbrace{P \cdot P \cdot \dots \cdot P}_{(k \text{ fois})}.$$

Par exemple, si nous reprenons la matrice P donnée dans la sous-section 2.2.1 ci-dessus, la matrice de transition en 5 étapes, nommée P^5 , représente l'ensemble des probabilités permettant de passer de chaque état i à un état j au bout d'exactly 5 étapes :

$$P^5 = \begin{bmatrix} 0,22350 & 0,24385 & 0,53265 \\ 0,24150 & 0,26140 & 0,49710 \\ 0,20595 & 0,22252 & 0,57153 \end{bmatrix}$$

Toujours en reprenant notre exemple, la probabilité de passer de l'état α à l'état β au bout d'exactly 5 étapes est donc de :

$$P_{\alpha,\beta}^5 = \mathcal{P}_{\alpha}(\beta)^5 = 24,385\%$$

2.3 Classes de communication et d'équivalence

2.3.1 Les différents types d'états

Dans une chaîne de Markov, chaque état peut être classifié en différente(s) catégorie(s) en fonction de ses liaisons avec les autres états. Ces différentes classifications nous permettent d'analyser le comportement de la chaîne en temps long.

Dans cette sous-section, nous définirons les différents types d'états que chaque état peut prendre dans une chaîne de Markov.

Un premier type que peut prendre un état est l'état récurrent. Un état i est dit **récurrent** si, en partant de cet état, la probabilité de le revisiter à un moment donné est égale à 1. Autrement dit, l'état i sera obligatoirement revisité / visité une infinité de fois lorsque l'on réalise une infinité de transitions. Plus rigoureusement, cela signifie que :

$$\mathbb{P}_i(\text{Num}_i(X) = \infty) = 1.$$

où $\text{Num}_i(X) = |\{n \geq 0, X_n = i\}|$.

Si cette probabilité est strictement inférieure à 1, l'état i est à l'inverse qualifié de **transitoire**. Le cas échéant, la chaîne quittera de manière systématique et définitive cet état lorsque l'on réalisera une infinité de

transitions.

Un deuxième type que peut prendre un état est l'état absorbant. Un état i est appelé **absorbant** si, une fois atteint, la chaîne reste dans cet état avec probabilité 1. Formellement, cet état satisfait :

$$P_{i,i} = 1 \quad \text{et} \quad \forall j \neq i, \quad P_{i,j} = 0.$$

A l'inverse, tout état ne respectant pas cette condition est appelé état **non-absorbant** ou encore état **évanescent**.

Un troisième type que peut prendre un état est le type périodique. Un état i est dit **périodique** si la chaîne ne peut revenir à cet état qu'après un nombre d'étapes multiple (un certain entier $d > 1$), appelé la période de i . La période $d(i)$ est définie comme :

$$d(i) = \text{PGCD}\{n \geq 1 \mid P_{i,i}^{(n)} > 0\}.$$

Si $d(i) = 1$, l'état i est dit **apériodique**, ce qui signifie qu'il est possible de revenir à cet état à chaque nouvelle transition / sans contrainte de périodicité.

Un quatrième et dernier type d'état que nous étudierons est le type accessible. Un état j est dit **accessible** depuis un état i si la probabilité de passer de i à j est non-nulle après un nombre fixé n de pas. Autrement dit :

$$\exists n \geq 1, \quad P_{i,j}^{(n)} > 0.$$

Réciproquement, un état i est dit **co-accessible** depuis j si j est accessible depuis i .

2.3.2 Classes de communication

Soit $\alpha, \beta \in E$ et i, j leurs indices respectifs dans E . On note $\alpha \rightarrow \beta$ si, et seulement si :

$$\exists n \geq 0, \quad P_{i,j}^{(n)} > 0.$$

On dit que les deux états α et β communiquent si $\alpha \rightarrow \beta$ et $\beta \rightarrow \alpha$. On note alors $\alpha \leftrightarrow \beta$. Nous avons ainsi :

$$\alpha \leftrightarrow \beta \quad \Longleftrightarrow \quad \exists n \geq 0, \quad P_{i,j}^{(n)} > 0 \quad \text{et} \quad \exists m \geq 0, \quad P_{j,i}^{(m)} > 0.$$

\leftrightarrow est une relation d'équivalence dont les classes sont appelée **classes de communications** de la chaîne de Markov. Autrement dit, dans une chaîne de Markov, une classe de communication est un ensemble d'états qui sont accessibles les uns depuis les autres.

L'ensemble des classes de communication, engendrées par \leftrightarrow , d'une chaîne de Markov sont **disjointes** et forment une **partition** de l'ensemble des état de

cette chaîne. Autrement dit, la relation d'équivalence \leftrightarrow partitionne l'ensemble des états d'une chaîne de Markov en classes de communication disjointes.

Il est important de noter qu'une classe de communication appartient systématiquement à l'une des deux catégories suivantes :

- **Fermée** : Une fois qu'un état de cette classe est atteint, il est impossible de quitter cette classe. Formellement, pour une classe C , cela signifie que :

$$\forall i \in C, \quad \forall j \notin C, \quad \forall n \geq 1, \quad P_{i,j}^{(n)} = 0.$$

- **Ouverte** : L'opposée de fermée ; il existe au moins un état de la classe qui permet de sortir de celle-ci. Plus formellement, cela signifie que :

$$\exists i \in C, \quad \exists j \notin C, \quad \exists n \geq 1, \quad P_{i,j}^{(n)} > 0.$$

2.3.3 Classes d'équivalence

Les classes d'équivalence sont des sous-ensembles d'états dans une chaîne de Markov qui partagent des propriétés spécifiques en plus d'être dans une même classe de communication. Une classe d'équivalence est donc une classe de communication qui regroupe des états voisins présentant une ou plusieurs des propriétés suivantes :

1. **Récurrence** : Tous les états de la classe sont récurrents. Cela signifie qu'une fois entré dans cette classe, il est certain que la chaîne revisitera un état de cette classe une infinité de fois au cours des transitions suivantes. Cette classe d'équivalence est systématiquement fermée.
2. **Transience (transitivité)** : Dans certaines classes de communication, les états peuvent être transitoires, ce qui signifie qu'il est possible de quitter définitivement la classe par ces états. Cette classe d'équivalence, à l'opposé de la première est toujours ouverte.
3. **Absorption** : Une classe d'équivalence récurrence (et fermée) peut également être absorbante. Cela signifie que, si la chaîne entre dans cette classe, elle ne pourra plus jamais en sortir.
4. **Apériodicité** : Tous les états d'une classe peuvent également être apériodiques, ce qui garantit l'absence de contraintes sur le moment où la chaîne peut revisiter un état donné. La classe d'équivalence ainsi formée peut être ouverte ou fermée.

Ainsi, une classe d'équivalence est une classe de communication fermée qui regroupe des états partageant une même propriété structurelle (tous les états de cette classe d'équivalence possède le même type d'état), telle que la récurrence, l'absorption, ou l'apériodicité.

Les classes d'équivalence jouent en conséquence un rôle crucial dans l'étude des chaînes de Markov. En effet, elles permettent de décomposer la chaîne en sous-ensembles analytiquement indépendants.

Par exemple ; les propriétés asymptotiques d'une chaîne dépendent fortement des classes d'équivalence fermées et récurrentes.

2.3.4 Exemples

TODO

2.4 Loi stationnaire

Soit une chaîne de Markov $(X_n)_{n \in \mathbb{N}}$ à espace d'état E , et ayant pour matrice de transition P .

Une mesure $\pi = (\pi_i)_{i \in E}$ sur l'espace d'état E est dite stationnaire si elle satisfait la relation $\pi = \pi P$.

Cela signifie que π est un vecteur propre à droite de la transposée P^\top , associé à la valeur propre 1 :

$$\pi = \pi P \iff \pi^\top = P^\top \pi^\top.$$

De plus, pour que π soit une mesure de probabilité, elle doit respecter les conditions suivantes :

$$\forall i \in E, \quad \pi_i \geq 0 \quad \text{et} \quad \sum_{i \in E} \pi_i = 1.$$

2.5 Comportement en temps long

Une chaîne de Markov est dite **irréductible** si le graphe qui lui est associé est fortement connexe. Autrement dit, tous les couples (α, β) de sommets du graphe, avec $\alpha \neq \beta$, communiquent mutuellement ; $\alpha \leftrightarrow \beta$.

Une chaîne de Markov est dite **récurrente** si tous ses états sont récurrents (voir sous-section 2.3.1).

Soit une chaîne de Markov $(X_n)_{n \in \mathbb{N}}$ à \mathbb{N} états contenus dans E et une matrice de transition P associée à cette chaîne. Un état α est alors dit **récurrent positif** s'il existe une mesure stationnaire $\pi = (\pi_i)_{i \in E}$ telle que $\pi_\alpha > 0$.

Cela signifie donc qu'en moyenne le temps entre deux visites consécutives de l'état α est fini.

Une chaîne de Markov est dite **récurrente positive** si tous ses états sont récurrents positifs.

Si une chaîne de Markov est irréductible et récurrente positive, alors sa matrice de transition P , élevée à une puissance k , converge lorsque $k \rightarrow +\infty$. Plus formellement :

$$L = \lim_{k \rightarrow +\infty} P^k$$

Où la matrice limite L est bien définie, et chaque ligne de cette matrice est égale à l'unique distribution stationnaire π associée à la chaîne.

3 Fonctionnement de l'Algorithme de Hasting-Metropolis

3.1 Qu'est-ce qu'une méthode dite MCMC ?

Les méthodes MC dites de Monte-Carlo ont pour but d'approcher de manière empirique des valeurs par un processus aléatoire répété un nombre suffisant de fois. Elles interviennent quand une résolution déterministe est trop difficile à obtenir, comme pour calculer des intégrales ou générer des échantillon de distributions statistiques. Les méthodes MC sont très fiables malgré l'aléatoire utilisé, pourvu que l'algorithme ait les bonnes optimisations telles que le nombre de répétitions, la précision exigée, la vitesse de convergence vers les valeurs cherchée, etc.

Les méthodes MC sont utilisées dans beaucoup de domaines des sciences (comme la physique, la chimie, la biologie, les mathématiques statistiques, l'intelligence artificielle, la finance et la cryptographie).

Les méthodes MCMC (Monte-Carlo par chaînes de Markov) sont une sous-famille des algorithmes MC qui construisent, dans leur fonctionnement, une chaîne de Markov dont la distribution stationnaire est celle que l'on souhaite estimer.

Il devient ainsi possible d'exploiter les propriétés des chaînes de Markov, notamment leur convergence vers une distribution stationnaire et leur propriété d'ergodicité, qui peuvent être plus utiles que la loi des grands nombres seule, utilisée dans les méthodes MC classiques.

La principale différence entre les méthodes MC et MCMC réside dans la dépendance entre les points générés. Les méthodes MC choisissent aléatoirement des points de l'espace, réalisant ainsi un échantillonnage direct de la distribution cible, avec des échantillons indépendants les uns des autres. Leur seul point commun est d'appartenir à l'intervalle des valeurs possibles de la distribution de départ. En revanche, les méthodes MCMC reposent sur une marche aléatoire : elles génèrent chaque nouveau point à partir du précédent selon une règle de transition déterminée. Cela crée une chaîne de points corrélés, qui possède les propriétés des chaînes de Markov.

Ainsi, lorsque la distribution est concentrée sur une certaine partie de l'espace, l'utilisation de MCMC sera généralement préférable, car elle permet d'obtenir, dans un temps raisonnable, des échantillons représentatifs de cette distribution.

On pourra citer les méthodes MCMC d'échantillonnage de Gibbs (et ses variantes comme le Blocked Gibbs Sampling ou l'Adaptive Gibbs Sampling), qui prennent leur intérêt lorsque les variables de la distribution sont conditionnelles. Il existe aussi l'échantillonnage par tranches, qui échantillonne progressivement la distribution par "tranches" de valeurs, ou bien des optimisations de HM, telles que la méthode hamiltonienne de Monte-Carlo, qui calcule le gradient pour faciliter la marche du programme et rejeter moins de valeurs, tout en

gagnant en performances.

3.2 Définitions

TODO

3.3 Description des étapes de l'algorithme

Voici une description de cet algorithme :

1. Il faut avant tout choisir un point x_0 , comme étant le premier échantillon de notre loi cible, et aussi une probabilité de transition g , en donnant les fonctions correspondant à $g_y(x)$ et $g(x)$, $\forall x, y \in E$.
2. Ensuite nous itérons sur t allant de 0 à N (notre nombre d'itérations voulu)
 - On tire x avec $g_{x_t}(x)$
 - On pose $\alpha := \frac{\pi(x)g_x(x_t)}{\pi(x_t)g_{x_t}(x)}$ (notons que si nous ne possédons qu'une densité proportionnelle à π , f nous pouvons poser $\alpha := \frac{f(x)g_x(x_t)}{f(x_t)g_{x_t}(x)}$)
 - On tire $u \in [0; 1]$, tel que $u \sim \mathcal{U}(0, 1)$
 - Si $u \leq \alpha$ alors : $x_{t+1} := x$
 - Sinon : on conserve l'état précédent $x_{t+1} := x$
3. La séquence $\{x_0, x_1, \dots, x_{N-1}\}$ constitue donc l'échantillon obtenue à partir de la chaîne de Markov associé à la loi π

3.4 Preuve de l'algorithme

TODO

3.5 Exemple d'application

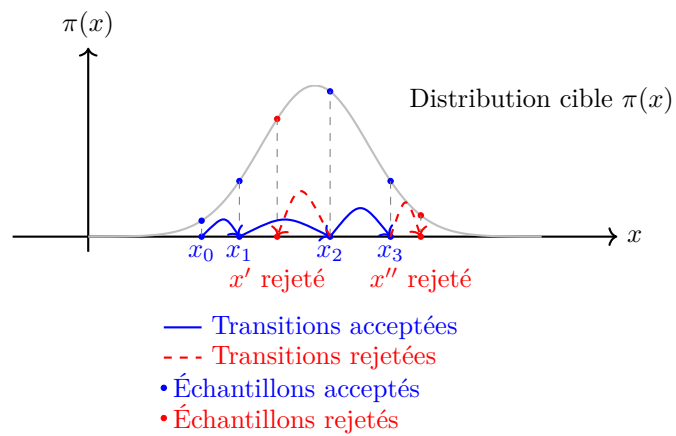
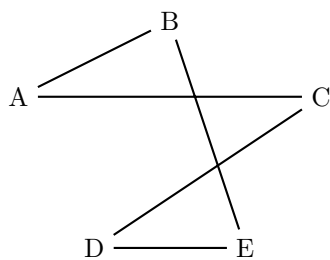
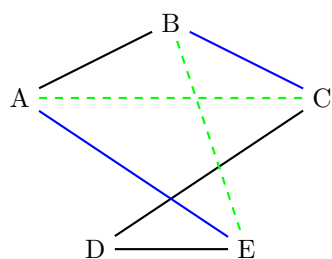


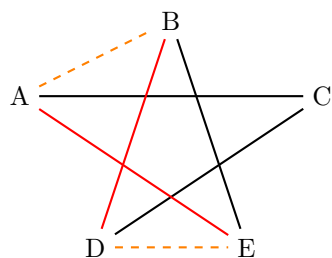
Diagramme initial à optimiser



Exemple de proposition retenue par HM



Exemple de proposition rejetée par HM



3.6 Comparaison avec d'autres méthodes d'échantillonnage

TODO

4 Conclusion

TODO

5 Sources

Documents textuels :

- https://fr.wikipedia.org/wiki/Chaîne_de_Markov
- https://dms.umontreal.ca/~bedard/BergeronL_rapport_final.pdf
- https://www.math.univ-paris13.fr/~tourner/fichiers/agreg/2014/cours_markov.pdf
- <https://www.imo.universite-paris-saclay.fr/~pierre-loic.meliot/agreg/markov.pdf>
- <https://www.college-de-france.fr/fr/agenda/cours/apprentissage-et-generation-par-echantillonnage-aleatoire/algorithmes-de-metropolis-hasting>
- <https://www.math.u-bordeaux.fr/~mibonnef/mimse-markov/recurrence-transience.pdf>
- https://fr.wikipedia.org/wiki/Méthode_de_Monte-Carlo_par_châînes_de_Markov

Documents vidéos :

- <https://www.youtube.com/watch?v=yCv2N7wGDCw>
- https://www.youtube.com/watch?v=MxI78mpq_44
- <https://www.youtube.com/watch?v=e0ZHDk4DSEI&list=PLWoShwK0FEjovcc32x9LbpDTf8pquPimV>
- <https://www.youtube.com/playlist?list=PLM8wYQRetTxBkdvBtz-gw8b9lcVkdXQKV>