IFT-3655, Modèles Stochastiques

Simulation: Méthode de Monte Carlo

Prof. Pierre L'Ecuyer
DIRO, Université de Montréal

Ces "diapos" sont un support pour les présentations en classe.
Un traitement beaucoup plus détaillé de ce sujet se trouve ici:
https://www-labs.iro.umontreal.ca/~lecuyer/ift6561/book.pdf

Simulation d'un chaîne de Markov

Supposons que l'on veut simuler sur ordinateur l'évolution d'une chaîne de Markov $\{X_n,\ n \geq 0\}$ avec espace d'états $\mathcal{X} = \{1, \ldots, k\}$ et probabilités de transition $P_{i,j}$.

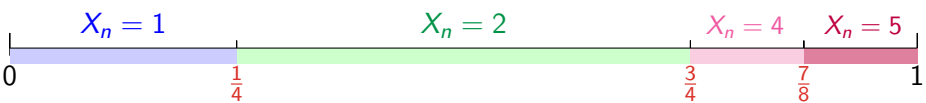
À chaque étape n, on est dans un état $X_{n-1} = i$, et on veut générer le prochain état X_n selon les probabilités $P_{i,j} = \mathbb{P}(X_n = j \mid X_{n-1} = i)$. Comment faire cela?

Simulation d'un chaîne de Markov

Supposons que l'on veut simuler sur ordinateur l'évolution d'une chaîne de Markov $\{X_n,\ n \geq 0\}$ avec espace d'états $\mathcal{X} = \{1, \ldots, k\}$ et probabilités de transition $P_{i,j}$.

À chaque étape n, on est dans un état $X_{n-1} = i$, et on veut générer le prochain état X_n selon les probabilités $P_{i,j} = \mathbb{P}(X_n = j \mid X_{n-1} = i)$. Comment faire cela?

Supposons par exemple que $P_{i,1} = 1/4$, $P_{i,2} = 1/2$, $P_{i,3} = 0$, et $P_{i,4} = P_{i,5} = 1/8$.
Les logiciels pour la simulation possèdent des fonctions pour générer $U \sim \mathcal{U}(0,1)$.
Pour générer X_n, on peut générer $U \sim \mathcal{U}(0,1)$, et retourner $X_n = j$ pour le j qui correspond à l'intervalle où se trouve U:



Ces probabilités sont en général différentes selon l'état courant i.

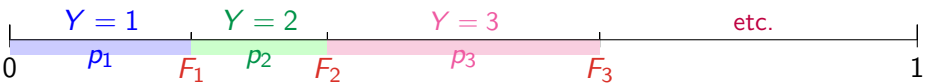
Problème du collectionneur: simulation

Il y a k types d'items et on tire un item à la fois, chacun étant du type j avec probabilité p_j, pour $j = 1, \ldots, k$.

On peut vouloir simuler ce système n fois pour estimer par exemple la distribution du nombre N de tirages requis pour obtenir une collection complète.

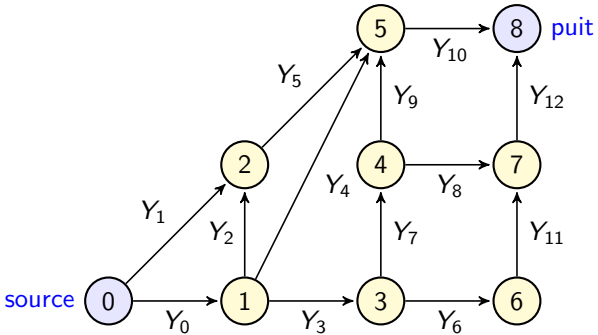
Pour simuler les tirages, on doit simuler une suite de réalisations indépendantes d'une variable aléatoire Y telle que $\mathbb{P}(Y = j) = p_j$. Ce Y represente le prochain type d'item obtenu.

Pour tirer Y, on divise l'intervalle $(0, 1)$ en sous-intervalles de longueurs $p_1, p_2, \ldots, p_k$, on tire $U \sim \mathcal{U}(0, 1)$, et on retourne le numéro j de l'intervalle correspondant.
Posons $F_j = \mathbb{P}(Y \leq j) = p_1 + \cdots + p_j$.



Un réseau d'activités stochastique.

Le graphe donne des relations de précédence entre les activités.
L'activité j a une durée aléatoire Y_j (la longueur de l'arc j) de fonction de répartition (cdf) F_j, i.e., $\mathbb{P}[Y_j \leq y] = F_j(y)$. Supposons pour simplifier que les Y_j sont indépendants.
La durée T du projet est la longueur (aléatoire) du plus long chemin de la source au puit.



On peut vouloir estimer $\mathbb{P}[T > x]$ pour x fixé, ou encore toute la loi de probabilité de T.

Calcul exact? Trop difficile en général.

On peut vouloir estimer $\mathbb{P}[T > x]$ pour x fixé, ou encore toute la loi de probabilité de T.

Calcul exact? Trop difficile en général.

Simulation: répéter n fois:
générer les Y_j selon F_j pour chaque j, puis calculer T.

On va voir plus loin comment générer les Y_j.

On obtient ainsi n réalisations indépendantes $T_1, \ldots, T_n$ de T.

On peut estimer la densité de T par un histogramme des T_i,
puis $\mathbb{E}[T]$ et $\mathbb{P}[T > x]$ par les moyennes empiriques, etc.

On peut vouloir estimer $\mathbb{P}[T > x]$ pour x fixé, ou encore toute la loi de probabilité de T.

Calcul exact? Trop difficile en général.

Simulation: répéter n fois:
générer les Y_j selon F_j pour chaque j, puis calculer T.

On va voir plus loin comment générer les Y_j.

On obtient ainsi n réalisations indépendantes $T_1, \ldots, T_n$ de T.

On peut estimer la densité de T par un histogramme des T_i,
puis $\mathbb{E}[T]$ et $\mathbb{P}[T > x]$ par les moyennes empiriques, etc.

Illustration numerique:
$Y_j \sim \max(0, N(\mu_j, \sigma_j^2))$ pour $j = 0, 1, 3, 10, 11$, et $Y_j \sim \text{Expon}(1/\mu_j)$ sinon.

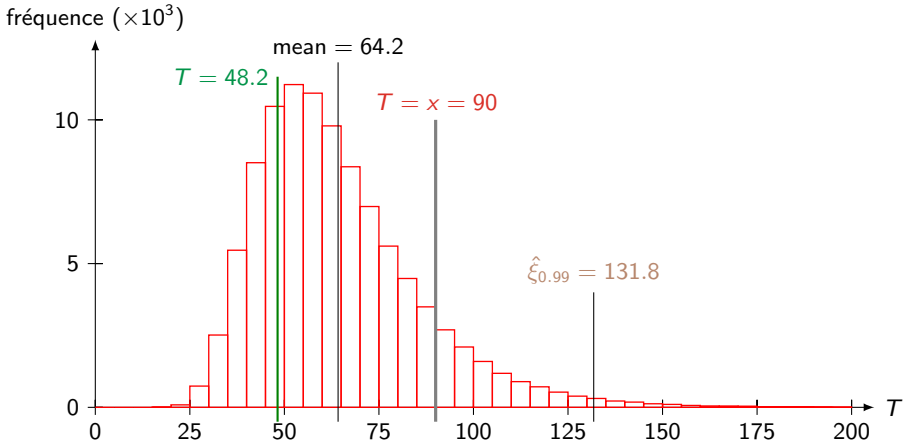
$\mu_0, \ldots, \mu_{12}$: 13.0, 5.5, 7.0, 5.2, 16.5, 14.7, 10.3, 6.0, 4.0, 20.0, 3.2, 3.2, 16.5.

On peut vouloir estimer $\mathbb{P}[T > 90]$ parce qu'on paye une pénalité si ça arrive.

Idée naive: remplacer chaque Y_j par son espérance. Donne $T = 48.2$.

Idée naive: remplacer chaque Y_j par son espérance. Donne $T = 48.2$.

J'ai simulé de modèle $n = 100\,000$ fois, en utilisant la librairie Java SSJ.
L'histogramme des n réalisations de T donne pas mal d'information.
Les valeurs de T vont de 14.4 à 268.6; 11.57% dépassent $x = 90$.



Simulation de variables aléatoires uniformes indépendantes

Objectif: On veut produire des suites de valeurs qui ont l'air d'être tirées au hasard.

Simulation de variables aléatoires uniformes indépendantes

Objectif: On veut produire des suites de valeurs qui ont l'air d'être tirées au hasard.

Exemple: Suites de bits (pile ou face):

$$01111010011011010100110110010100011 1?...$$

Loi uniforme: chaque bit est 1 avec probabilité $1/2$.

Simulation de variables aléatoires uniformes indépendantes

Objectif: On veut produire des suites de valeurs qui ont l'air d'être tirées au hasard.

Exemple: Suites de bits (pile ou face):

$$01111?100110?1?10100110110010100011 1 \ldots$$

Loi uniforme: chaque bit est 1 avec probabilité $1/2$.

Uniformité et indépendance:
Exemple: on 8 possibilités pour les 3 bits ? ? ?:

$$000,\ 001,\ 010,\ 011,\ 100,\ 101,\ 110,\ 111$$

On veut une proba. de $1/8$ pour chacune, peu importe les autres bits.

Simulation de variables aléatoires uniformes indépendantes

Objectif: On veut produire des suites de valeurs qui ont l'air d'être tirées au hasard.

Exemple: Suites de bits (pile ou face):

$$01111?100110?1?10100110110010100011 1 \ldots$$

Loi uniforme: chaque bit est 1 avec probabilité 1/2.

Uniformité et indépendance:
Exemple: on 8 possibilités pour les 3 bits ? ? ?:

$$000,\ 001,\ 010,\ 011,\ 100,\ 101,\ 110,\ 111$$

On veut une proba. de 1/8 pour chacune, peu importe les autres bits.

Pour s bits, on veut une probabilité de $1/2^s$ pour chacune des 2^s possibilités.

Loi uniforme sur $(0,1)$

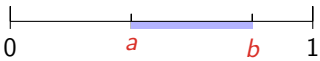
Pour la simulation en général, on veut imiter une suite $U_0, U_1, U_2, \ldots$ de variables aléatoires indépendantes de loi uniforme sur $(0,1)$.

Uniformité: On veut $\mathbb{P}[a \leq U_j \leq b] = b - a$.

Loi uniforme sur $(0, 1)$

Pour la simulation en général, on veut imiter une suite $U_0, U_1, U_2, \ldots$ de variables aléatoires indépendantes de loi uniforme sur $(0, 1)$.

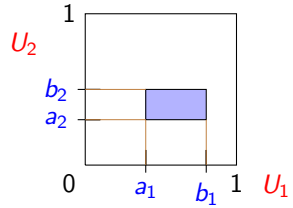
Uniformité: On veut $\mathbb{P}[a \leq U_j \leq b] = b - a$.



Indépendance: Pour $(U_1, \ldots, U_s)$ en s dimensions, on veut

$$\mathbb{P}[a_j \leq U_j \leq b_j \text{ pour } j = 1, \ldots, s] = (b_1 - a_1) \cdots (b_s - a_s).$$

Exemple pour $s = 2$:



Loi uniforme sur $(0,1)$

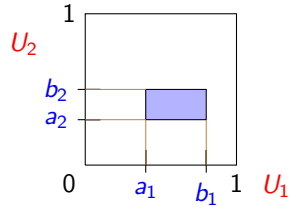
Pour la simulation en général, on veut imiter une suite $U_0, U_1, U_2, \ldots$ de variables aléatoires indépendantes de loi uniforme sur $(0,1)$.

Uniformité: On veut $\mathbb{P}[a \leq U_j \leq b] = b - a$.

Indépendance: Pour $(U_1, \ldots, U_s)$ en s dimensions, on veut

$$\mathbb{P}[a_j \leq U_j \leq b_j \text{ pour } j = 1, \ldots, s] = (b_1 - a_1) \cdots (b_s - a_s).$$

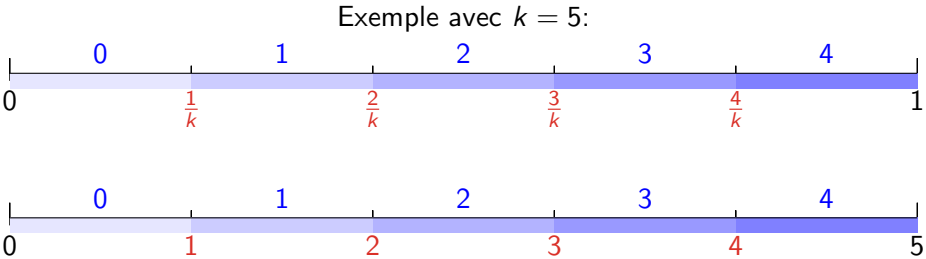
Exemple pour $s = 2$:



Cette notion de v.a. uniformes et indépendantes n'est qu'une abstraction mathématique.
N'existe peut-être même pas dans la réalité physique!

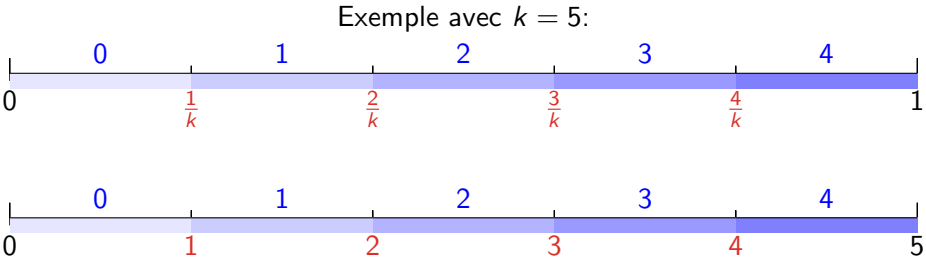
Générer un entier au hasard dans $\{0, 1, \ldots, k-1\}$

Générer $U \sim \mathcal{U}(0,1)$ et retourner $X = \lfloor kU \rfloor$.



Générer un entier au hasard dans $\{0, 1, \ldots, k-1\}$

Générer $U \sim \mathcal{U}(0,1)$ et retourner $X = \lfloor kU \rfloor$.



Si on veut X uniforme sur $\{1, \ldots, k\}$, on retourne $X = 1 + \lfloor kU \rfloor$.

Permutation aléatoire

1 2 3 4 5 6 7

Permutation aléatoire

1 2 3 4 5 6 7

1 2 3 4 6 7 5

Permutation aléatoire

1 2 3 4 5 6 7	
1 2 3 4 6 7	5
1 3 4 6 7	5 2

Permutation aléatoire

1 2 3 4 5 6 7	
1 2 3 4 6 7	5
1 3 4 6 7	5 2
3 4 6 7	5 2 1

Permutation aléatoire

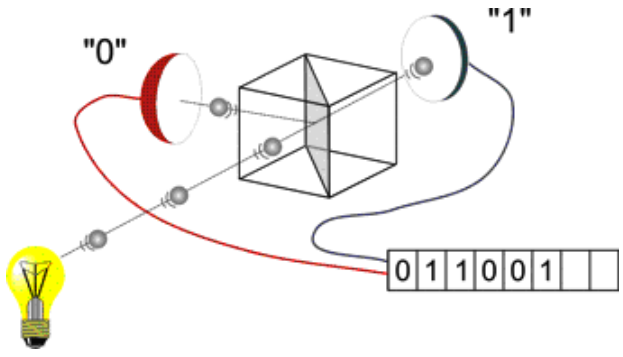
1 2 3 4 5 6 7	
1 2 3 4 6 7	5
1 3 4 6 7	5 2
3 4 6 7	5 2 1

Pour n objets, on choisit un entier de 1 à n,
puis un autre entier de 1 à $n-1$, puis de 1 à $n-2$, ...
On veut que chaque permutation ait la même probabilité.

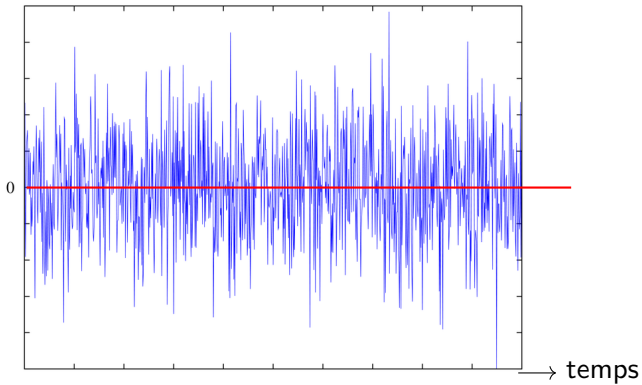
Ex.: pour permuter 52 cartes, il y a $52! \approx 2^{226}$ possibilités.

Bits aléatoires par des mécanismes physiques

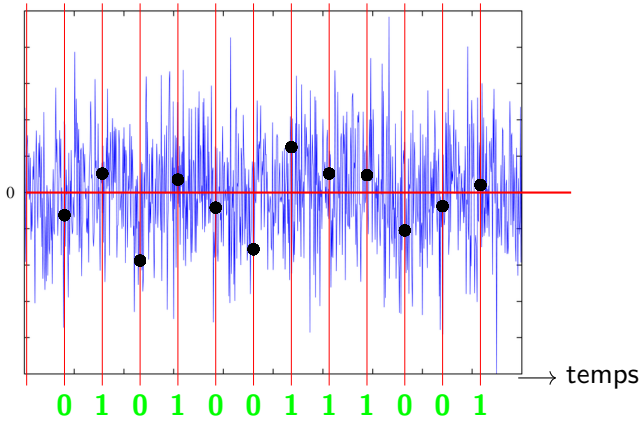
Trajectoires de photons (système vendu par **id-Quantique**):



Bruit thermique dans les résistances de circuits électroniques

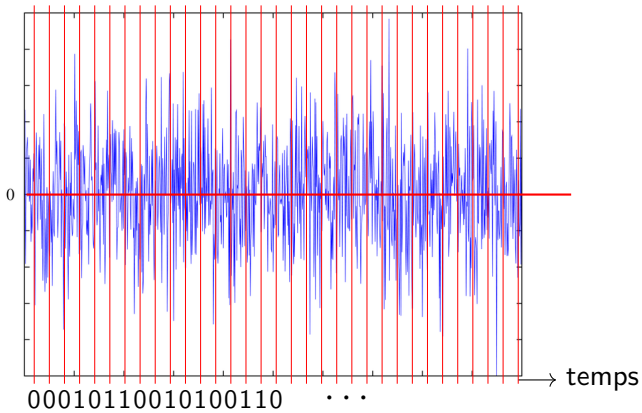


Bruit thermique dans les résistances de circuits électroniques



On échantillonne le signal (voltage) périodiquement.

Bruit thermique dans les résistances de circuits électroniques



On échantillonne le signal (voltage) périodiquement.

Plusieurs mécanismes sont brevetés et disponibles commercialement.

Aucun n'est parfait.

Plusieurs mécanismes sont brevetés et disponibles commercialement.

Aucun n'est parfait. On peut dimimuer le biais et/ou la dépendance en combinant des blocs de bits. Par exemple par un XOR:

$$\underbrace{0\,1}\ \underbrace{1\,0}\ \underbrace{0\,0}\ \underbrace{1\,0}\ \underbrace{0\,1}\ \underbrace{1\,0}\ \underbrace{1\,1}\ \underbrace{0\,1}\ \underbrace{0\,0}$$
$$1 \quad 1 \quad 0 \quad 1 \quad 1 \quad 1 \quad 0 \quad 1 \quad 0$$

Plusieurs mécanismes sont brevetés et disponibles commercialement.

Aucun n'est parfait. On peut dimimuer le biais et/ou la dépendance en combinant des blocs de bits. Par exemple par un XOR:

$$\underbrace{0\,1}\ \underbrace{1\,0}\ \underbrace{0\,0}\ \underbrace{1\,0}\ \underbrace{0\,1}\ \underbrace{1\,0}\ \underbrace{1\,1}\ \underbrace{0\,1}\ \underbrace{0\,0}$$
$$1 \quad 1 \quad 0 \quad 1 \quad 1 \quad 1 \quad 0 \quad 1 \quad 0$$

ou encore (élimine le biais):

$$\underbrace{0\,1}\ \underbrace{1\,0}\ \underbrace{0\,0}\ \underbrace{1\,0}\ \underbrace{0\,1}\ \underbrace{1\,0}\ \underbrace{1\,1}\ \underbrace{0\,1}\ \underbrace{0\,0}$$
$$0 \quad 1 \qquad 1 \quad 0 \quad 1 \qquad 0$$

Plusieurs mécanismes sont brevetés et disponibles commercialement.

Aucun n'est parfait. On peut dimimuer le biais et/ou la dépendance en combinant des blocs de bits. Par exemple par un XOR:

$$\underbrace{0\,1}\ \underbrace{1\,0}\ \underbrace{0\,0}\ \underbrace{1\,0}\ \underbrace{0\,1}\ \underbrace{1\,0}\ \underbrace{1\,1}\ \underbrace{0\,1}\ \underbrace{0\,0}$$
$$1 \quad 1 \quad 0 \quad 1 \quad 1 \quad 1 \quad 0 \quad 1 \quad 0$$

ou encore (élimine le biais):

$$\underbrace{0\,1}\ \underbrace{1\,0}\ \underbrace{0\,0}\ \underbrace{1\,0}\ \underbrace{0\,1}\ \underbrace{1\,0}\ \underbrace{1\,1}\ \underbrace{0\,1}\ \underbrace{0\,0}$$
$$0 \quad 1 \qquad 1 \quad 0 \quad 1 \qquad 0$$

Mécanisme physique:
Essentiel pour cryptologie, loteries, etc. Mais pas pour la simulation.
Encombrant, pas reproduisible, pas toujours fiable, pas d'analyse mathématique de l'uniformité et de l'indépendance à long terme.

Générateurs algorithmiques (pseudo-aléatoires, GPA)

Mini-exemple: On veut imiter des nombres de 1 à 100 tirés au hasard.

Générateurs algorithmiques (pseudo-aléatoires, GPA)

Mini-exemple: On veut imiter des nombres de 1 à 100 tirés au hasard.

1. Choisir un nombre x_0 au hasard dans $\{1, \ldots, 100\}$.
2. Pour $n = 1, 2, 3, \ldots$, retourner $\boxed{x_n = 12\, x_{n-1} \bmod 101}$.

Générateurs algorithmiques (pseudo-aléatoires, GPA)

Mini-exemple: On veut imiter des nombres de 1 à 100 tirés au hasard.

1. Choisir un nombre x_0 au hasard dans $\{1, \ldots, 100\}$.
2. Pour $n = 1, 2, 3, \ldots$, retourner $\boxed{x_n = 12\, x_{n-1} \bmod 101}$. Par exemple, si $x_0 = 1$:

$$x_1 \quad = \quad (12 \times 1 \bmod 101) = 12,$$

Générateurs algorithmiques (pseudo-aléatoires, GPA)

Mini-exemple: On veut imiter des nombres de 1 à 100 tirés au hasard.

1. Choisir un nombre x_0 au hasard dans $\{1, \ldots, 100\}$.
2. Pour $n = 1, 2, 3, \ldots$, retourner $\boxed{x_n = 12\, x_{n-1} \bmod 101}$. Par exemple, si $x_0 = 1$:

$$\begin{aligned} x_1 &= (12 \times 1 \bmod 101) = 12, \\ x_2 &= (12 \times 12 \bmod 101) = (144 \bmod 101) = 43, \end{aligned}$$

Générateurs algorithmiques (pseudo-aléatoires, GPA)

Mini-exemple: On veut imiter des nombres de 1 à 100 tirés au hasard.

1. Choisir un nombre x_0 au hasard dans $\{1, \ldots, 100\}$.
2. Pour $n = 1, 2, 3, \ldots$, retourner $\boxed{x_n = 12\, x_{n-1} \bmod 101}$. Par exemple, si $x_0 = 1$:

$$
\begin{aligned}
x_1 &= (12 \times 1 \bmod 101) = 12, \\
x_2 &= (12 \times 12 \bmod 101) = (144 \bmod 101) = 43, \\
x_3 &= (12 \times 43 \bmod 101) = (516 \bmod 101) = 11, \quad \text{etc.} \\
x_n &= 12^n \bmod 101.
\end{aligned}
$$

Visite tous les nombres de 1 à 100 une fois chacun avant de revenir à x_0.
(Parce que $m = 101$ est premier et $a = 12$ est primitif modulo 101.)

Générateurs algorithmiques (pseudo-aléatoires, GPA)

Mini-exemple: On veut imiter des nombres de 1 à 100 tirés au hasard.

1. Choisir un nombre x_0 au hasard dans $\{1, \ldots, 100\}$.
2. Pour $n = 1, 2, 3, \ldots$, retourner $\boxed{x_n = 12\, x_{n-1} \bmod 101}$. Par exemple, si $x_0 = 1$:

$$\begin{aligned} x_1 &= (12 \times 1 \bmod 101) = 12, \\ x_2 &= (12 \times 12 \bmod 101) = (144 \bmod 101) = 43, \\ x_3 &= (12 \times 43 \bmod 101) = (516 \bmod 101) = 11, \quad \text{etc.} \\ x_n &= 12^n \bmod 101. \end{aligned}$$

Visite tous les nombres de 1 à 100 une fois chacun avant de revenir à x_0.
(Parce que $m = 101$ est premier et $a = 12$ est primitif modulo 101.)

Si on veut des nombres réels entre 0 et 1:

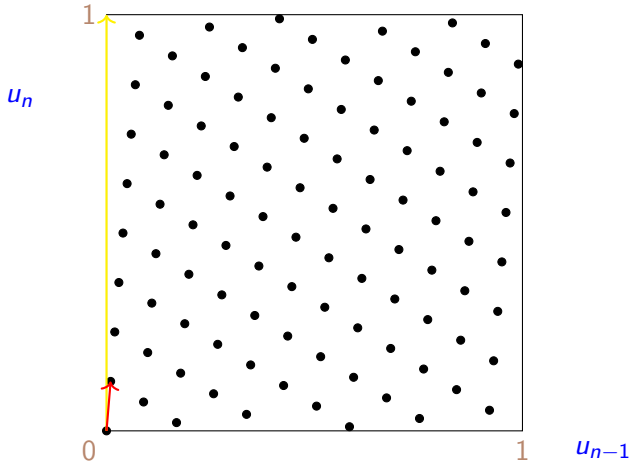
$$\begin{aligned} u_1 &= x_1/101 = 12/101 \approx 0.11881188..., \\ u_2 &= x_2/101 = 43/101 \approx 0.42574257..., \\ u_3 &= x_3/101 = 11/101 \approx 0.10891089..., \quad \text{etc.} \end{aligned}$$

Ici, u_i visite toutes les valeurs $\{1/101, 2/101, \ldots, 100/101\}$ quand i va de 1 à 100.
On a donc une excellente uniformité en une dimension.

$x_n = 12\,x_{n-1} \bmod 101; \qquad u_n = x_n/101 = 12\,u_{n-1} \bmod 1$

Paires de valeurs successives (u_{n-1}, u_n):

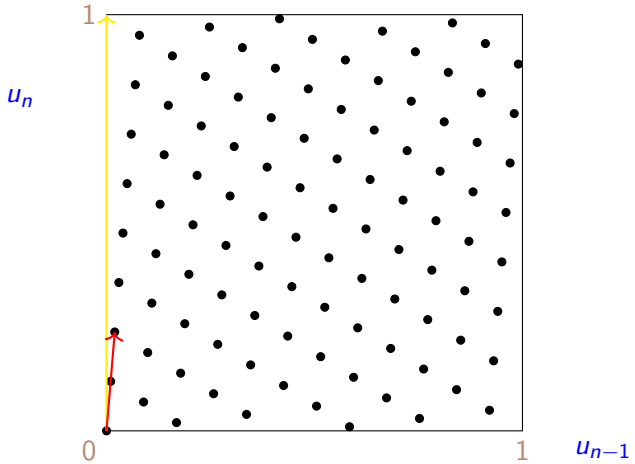
$(1/101, 12/101)$, $(2/101, 24/101)$, $(3/101, 36/101)$, ...



$x_n = 12\,x_{n-1} \bmod 101; \qquad u_n = x_n/101 = 12\,u_{n-1} \bmod 1$

Paires de valeurs successives (u_{n-1}, u_n):

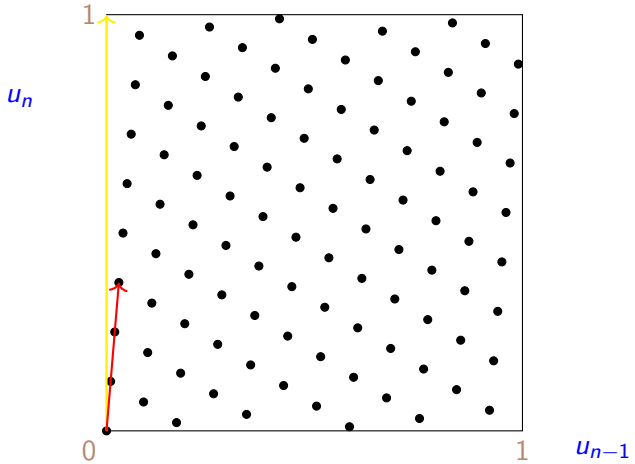
$(1/101, 12/101)$, $(2/101, 24/101)$, $(3/101, 36/101)$, ...



$x_n = 12\,x_{n-1} \bmod 101; \qquad u_n = x_n/101 = 12\,u_{n-1} \bmod 1$

Paires de valeurs successives (u_{n-1}, u_n):

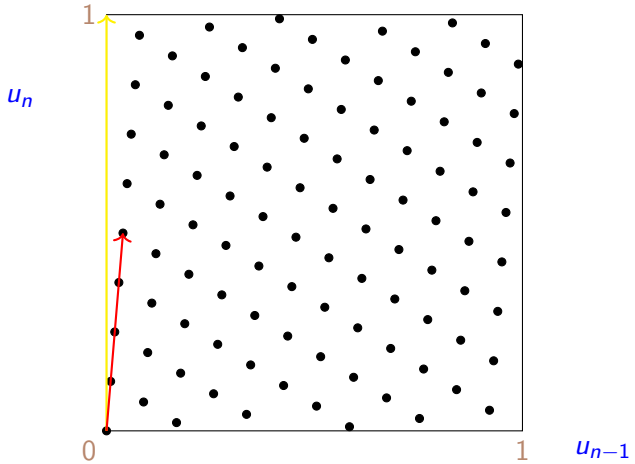
$(1/101, 12/101)$, $(2/101, 24/101)$, $(3/101, 36/101)$, ...



$x_n = 12\,x_{n-1} \bmod 101; \qquad u_n = x_n/101 = 12\,u_{n-1} \bmod 1$

Paires de valeurs successives (u_{n-1}, u_n):

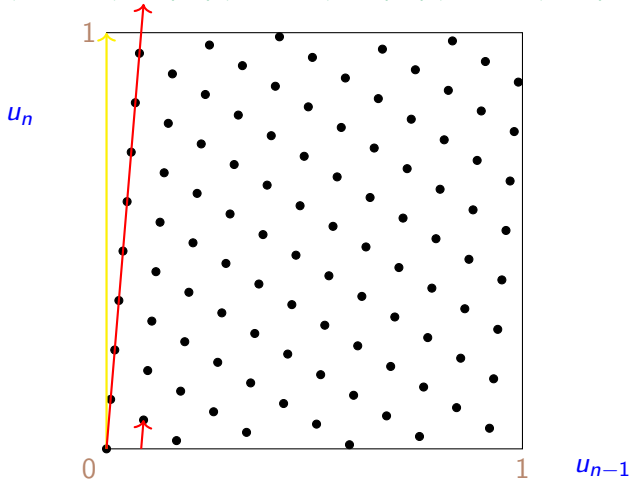
$(1/101, 12/101)$, $(2/101, 24/101)$, $(3/101, 36/101)$, ...



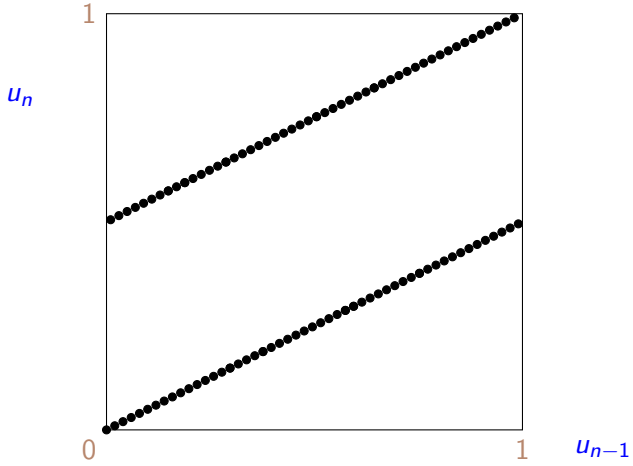
$x_n = 12\,x_{n-1} \bmod 101; \qquad u_n = x_n/101 = 12\,u_{n-1} \bmod 1$

Paires de valeurs successives (u_{n-1}, u_n):

$(1/101, 12/101)$, $(2/101, 24/101)$, $(3/101, 36/101)$, ...



$x_n = 51\, x_{n-1} \bmod 101; \qquad u_n = x_n/101.$



Ici, on a une bonne uniformité en une dimension, mais pas en deux!
On a les points $(1/101, 51/101)$, $(2/101, 1/101)$, $(3/101, 52/101)$, ...

Une récurrence linéaire plus longue

On choisit 3 entiers x_{-2}, x_{-1}, x_0 dans $\{0, 1, \ldots, 4294967086\}$, pas tous 0. Puis pour $n = 1, 2, \ldots,$ soit

$$\begin{aligned} x_n &= (1403580 x_{n-2} - 810728 x_{n-3}) \bmod 4294967087, \\ u_n &= x_n / 4294967087. \end{aligned}$$

Une récurrence linéaire plus longue

On choisit 3 entiers x_{-2}, x_{-1}, x_0 dans $\{0, 1, \ldots, 4294967086\}$, pas tous 0. Puis pour $n = 1, 2, \ldots$, soit

$$\begin{aligned} x_n &= (1403580 x_{n-2} - 810728 x_{n-3}) \bmod 4294967087, \\ u_n &= x_n / 4294967087. \end{aligned}$$

On peut prouver que la suite $x_0, x_1, x_2, \ldots$ est périodique, de période $4294967087^3 - 1 \approx 2^{96}$, et que le vecteur (x_{n-2}, x_{n-1}, x_n) visite chacun des $4294967087^3 - 1$ triplets non nuls exactement une fois lorsque n parcourt un cycle.

Combinaison de deux récurrences: MRG32k3a

On choisit 6 entiers:

x_0, x_1, x_2 dans $\{0, 1, \ldots, 4294967086\}$ (pas tous 0) et
y_0, y_1, y_2 dans $\{0, 1, \ldots, 4294944442\}$ (pas tous 0).

$$
\begin{aligned}
x_n &= (1403580 x_{n-2} - 810728 x_{n-3}) \bmod 4294967087, \\
y_n &= (527612 y_{n-1} - 1370589 y_{n-3}) \bmod 4294944443, \\
u_n &= [(x_n - y_n) \bmod 4294967087]/4294967087.
\end{aligned}
$$

Combinaison de deux récurrences: MRG32k3a

On choisit 6 entiers:
x_0, x_1, x_2 dans $\{0, 1, \ldots, 4294967086\}$ (pas tous 0) et
y_0, y_1, y_2 dans $\{0, 1, \ldots, 4294944442\}$ (pas tous 0).

$$
\begin{aligned}
x_n &= (1403580 x_{n-2} - 810728 x_{n-3}) \bmod 4294967087, \\
y_n &= (527612 y_{n-1} - 1370589 y_{n-3}) \bmod 4294944443, \\
u_n &= [(x_n - y_n) \bmod 4294967087]/4294967087.
\end{aligned}
$$

(x_{n-2}, x_{n-1}, x_n) visite chacune des $4294967087^3 - 1$ valeurs possibles.
(y_{n-2}, y_{n-1}, y_n) visite chacune des $4294944443^3 - 1$ valeurs possibles.

La suite $u_0, u_1, u_2, \ldots$ se répète avec une période proche de $2^{191} \approx 3.1 \times 10^{57}$.
L'uniformité des points $(u_{n+1}, \ldots, u_{n+s})$ dans $[0, 1)^s$ a été mesurée mathématiquement jusqu'en $s = 48$ dimensions.

Combinaison de deux récurrences: MRG32k3a

On choisit 6 entiers:
x_0, x_1, x_2 dans $\{0, 1, \ldots, 4294967086\}$ (pas tous 0) et
y_0, y_1, y_2 dans $\{0, 1, \ldots, 4294944442\}$ (pas tous 0).

$$
\begin{aligned}
x_n &= (1403580 x_{n-2} - 810728 x_{n-3}) \bmod 4294967087, \\
y_n &= (527612 y_{n-1} - 1370589 y_{n-3}) \bmod 4294944443, \\
u_n &= [(x_n - y_n) \bmod 4294967087]/4294967087.
\end{aligned}
$$

(x_{n-2}, x_{n-1}, x_n) visite chacune des $4294967087^3 - 1$ valeurs possibles.
(y_{n-2}, y_{n-1}, y_n) visite chacune des $4294944443^3 - 1$ valeurs possibles.

La suite $u_0, u_1, u_2, \ldots$ se répète avec une période proche de $2^{191} \approx 3.1 \times 10^{57}$.
L'uniformité des points $(u_{n+1}, \ldots, u_{n+s})$ dans $[0, 1)^s$ a été mesurée mathématiquement jusqu'en $s = 48$ dimensions.

Excellent générateur, robuste et fiable!

Disponible dans SSJ, SAS, R, MATLAB, Arena, Automod, Witness, machines Spielo, ...

Générateurs algorithmiques

C'est ce qu'on utilise pour la simulation.

Une fois les paramètres et l'état initial choisis, la suite devient complètement déterministe.
On peut choisir l'état initial au hasard si on veut.

Générateurs algorithmiques

C'est ce qu'on utilise pour la simulation.

Une fois les paramètres et l'état initial choisis, la suite devient complètement déterministe. On peut choisir l'état initial au hasard si on veut.

Avantages: pas de matériel à installer, un logiciel suffit; souvent plus rapide; on peut facilement répéter la même séquence.

Générateurs algorithmiques

C'est ce qu'on utilise pour la simulation.

Une fois les paramètres et l'état initial choisis, la suite devient complètement déterministe.
On peut choisir l'état initial au hasard si on veut.

Avantages: pas de matériel à installer, un logiciel suffit; souvent plus rapide; on peut facilement répéter la même séquence.

Désavantage: ne peut pas créer de l'entropie!
Il y a nécessairement des dépendances entre les nombres en sortie.

Qualités requises pour un bon générateur: dépend des applications.

Pour la cryptographie, un générateur algorithmique seul ne suffit pas.

Générateur algorithmique

$\mathcal{S}$, espace d'états fini;
$f : \mathcal{S} \to \mathcal{S}$, fonction de transition;
$g : \mathcal{S} \to [0, 1]$, fonction de sortie.

s_0, germe (état initial);

s_0

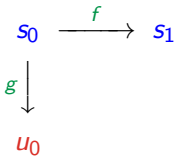
Générateur algorithmique

$\mathcal{S}$, espace d'états fini;
$f : \mathcal{S} \to \mathcal{S}$, fonction de transition;
$g : \mathcal{S} \to [0, 1]$, fonction de sortie.

s_0, germe (état initial);

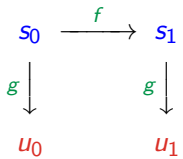
Générateur algorithmique

$\mathcal{S}$, espace d'états fini; s_0, germe (état initial);
$f : \mathcal{S} \to \mathcal{S}$, fonction de transition;
$g : \mathcal{S} \to [0, 1]$, fonction de sortie.



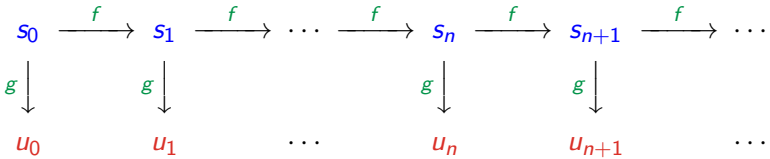
Générateur algorithmique

$\mathcal{S}$, espace d'états fini; s_0, germe (état initial);
$f : \mathcal{S} \to \mathcal{S}$, fonction de transition;
$g : \mathcal{S} \to [0, 1]$, fonction de sortie.



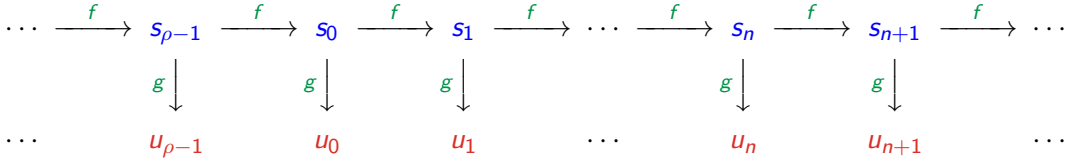
Générateur algorithmique

$\mathcal{S}$, espace d'états fini; s_0, germe (état initial);
$f : \mathcal{S} \to \mathcal{S}$, fonction de transition;
$g : \mathcal{S} \to [0, 1]$, fonction de sortie.



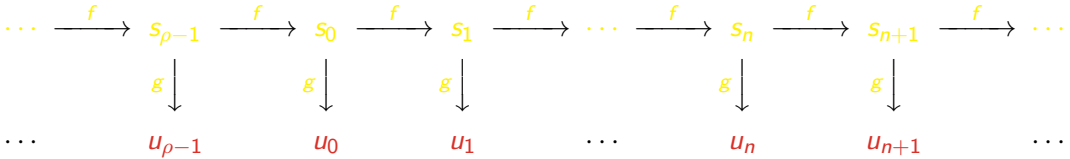
Générateur algorithmique

$\mathcal{S}$, espace d'états fini; s_0, germe (état initial);
$f : \mathcal{S} \to \mathcal{S}$, fonction de transition;
$g : \mathcal{S} \to [0, 1]$, fonction de sortie.

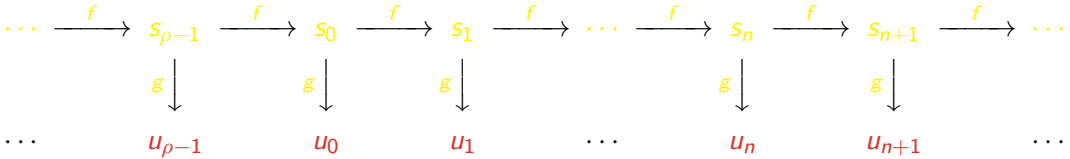


Période de $\{s_n,\ n \geq 0\}$: $\rho \leq$ cardinalité de $\mathcal{S}$.

Exemple: Dans MRG32k3a, s_n est un vecteur de 6 entiers de 32 bits et $\rho \approx 2^{191}$.

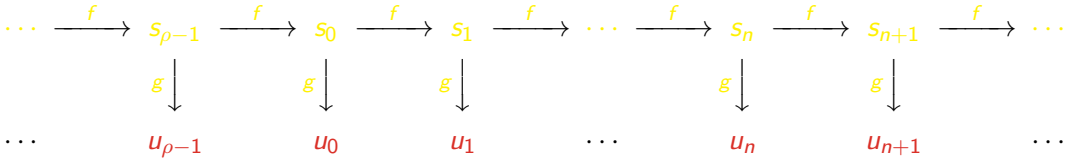


Objectif: en observant seulement $(u_0, u_1, \ldots)$, difficile de distinguer d'une suite de v.a. indépendantes uniformes sur $(0, 1)$.



Objectif: en observant seulement $(u_0, u_1, \ldots)$, difficile de distinguer d'une suite de v.a. indépendantes uniformes sur $(0, 1)$.

Utopie: passe tous les tests statistiques imaginables.
Impossible! On doit se contenter d'une approximation.

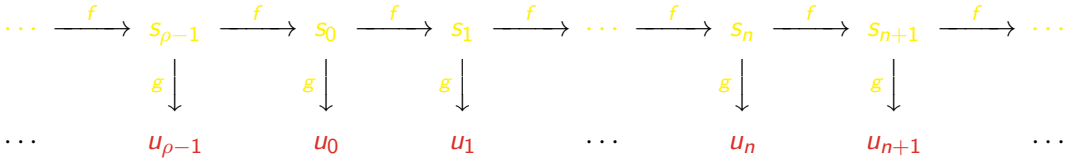


Objectif: en observant seulement $(u_0, u_1, \ldots)$, difficile de distinguer d'une suite de v.a. indépendantes uniformes sur $(0, 1)$.

Utopie: passe tous les tests statistiques imaginables.
Impossible! On doit se contenter d'une approximation.

On veut aussi: vitesse, facilité d'implantation, suites reproduisibles.

Compromis entre vitesse / propriétés statistiques / imprévisibilité.



Objectif: en observant seulement $(u_0, u_1, \ldots)$, difficile de distinguer d'une suite de v.a. indépendantes uniformes sur $(0, 1)$.

Utopie: passe tous les tests statistiques imaginables.
Impossible! On doit se contenter d'une approximation.

On veut aussi: vitesse, facilité d'implantation, suites reproduisibles.

Compromis entre vitesse / propriétés statistiques / imprévisibilité.

Machines de casinos et loteries: on modifie l'état s_n régulièrement à l'aide de mécanismes physiques.

La loi uniforme sur $[0,1]^s$.

Si on choisit s_0 au hasard dans $\mathcal{S}$ et on génère s nombres, cela correspond à choisir un point au hasard dans l'ensemble fini

$$\Psi_s = \{\boldsymbol{u} = (u_0, \ldots, u_{s-1}) = (g(s_0), \ldots, g(s_{s-1})),\ s_0 \in \mathcal{S}\}.$$

On veut approximer: "$\boldsymbol{u}$ suit la loi uniforme sur $[0,1]^s$."

La loi uniforme sur $[0,1]^s$.

Si on choisit s_0 au hasard dans $\mathcal{S}$ et on génère s nombres, cela correspond à choisir un point au hasard dans l'ensemble fini

$$\Psi_s = \{\boldsymbol{u} = (u_0, \ldots, u_{s-1}) = (g(s_0), \ldots, g(s_{s-1})),\ s_0 \in \mathcal{S}\}.$$

On veut approximer: "$\boldsymbol{u}$ suit la loi uniforme sur $[0,1]^s$."

Mesure de qualité: Ψ_s doit recouvrir $[0,1]^s$ très uniformément.

La loi uniforme sur $[0,1]^s$.

Si on choisit s_0 au hasard dans $\mathcal{S}$ et on génère s nombres, cela correspond à choisir un point au hasard dans l'ensemble fini

$$\Psi_s = \{\boldsymbol{u} = (u_0, \ldots, u_{s-1}) = (g(s_0), \ldots, g(s_{s-1})),\ s_0 \in \mathcal{S}\}.$$

On veut approximer: "$\boldsymbol{u}$ suit la loi uniforme sur $[0,1]^s$."

Mesure de qualité: Ψ_s doit recouvrir $[0,1]^s$ très uniformément.

Conception et analyse théorique des générateurs:

1. Définir une mesure d'uniformité de Ψ_s, calculable
   sans générer les points explicitement. GPA linéaires.
2. Choisir un type de construction (rapide, longue période, etc.)
   et chercher des paramètres qui "optimisent" cette mesure.

Mythe 1. Après au moins 60 ans à étudier les GPA et des milliers d'articles publiés, ce problème est certainement réglé et les GPA disponibles dans les logiciels populaires sont certainement fiables.

Mythe 1. Après au moins 60 ans à étudier les GPA et des milliers d'articles publiés, ce problème est certainement réglé et les GPA disponibles dans les logiciels populaires sont certainement fiables.

Non.

Mythe 2. Dans votre logiciel favori, le générateur a une période supérieure à 2^{1000}. Il est donc certainement excellent!

Mythe 1. Après au moins 60 ans à étudier les GPA et des milliers d'articles publiés, ce problème est certainement réglé et les GPA disponibles dans les logiciels populaires sont certainement fiables.

Non.

Mythe 2. Dans votre logiciel favori, le générateur a une période supérieure à 2^{1000}. Il est donc certainement excellent!

Non.

Exemple 1. $u_n = (n/2^{1000}) \bmod 1$ pour $n = 0, 1, 2, \ldots$.

Exemple 2. Lagged-Fibonacci, subtract-with-borrow.

Suites ("streams") et sous-suites multiples

Un seul GPA (monolithique) ne suffit pas. On a souvent besoin de plusieurs flux (ou suites, ou "streams") "indépendants" de nombres aléatoires. Exemples:

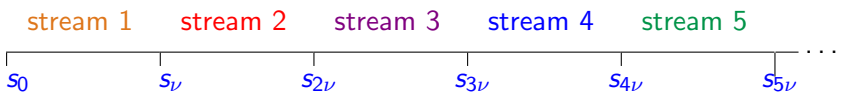
- exécuter une simulation sur plusieurs processeurs en parallèle,
- Comparaison de systèmes avec valeurs aléatoires communes
  (important pour analyse de sensibilité, estimation de dérivées, optimisation, ...).

Suites ("streams") et sous-suites multiples

Un seul GPA (monolithique) ne suffit pas. On a souvent besoin de plusieurs flux (ou suites, ou "streams") "indépendants" de nombres aléatoires. Exemples:

- exécuter une simulation sur plusieurs processeurs en parallèle,
- Comparaison de systèmes avec valeurs aléatoires communes (important pour analyse de sensibilité, estimation de dérivées, optimisation, ...).

Un logiciel développé au DIRO fournit de tels objets appelés `RandomStream`'s, qui agisssent comme des GPA virtuels. On peut en créer autant qu'on veut.
On partitionne la suite en segments disjoints (streams) de longueur ν:



Par exemple, pour MRG32k3a, la période est $\rho \approx 2^{191}$ et on a pris des segments de longueur $\nu = 2^{127}$. On peut donc avoir jusqu'à 2^{64} `RandomStream`'s disjoints.

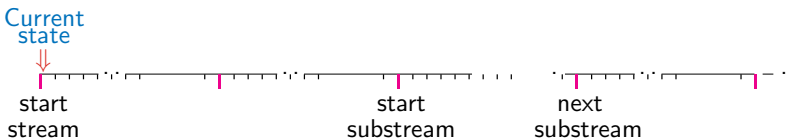
Suites multiples en Java

```
RandomStream stream1 = new MRG32k3a();
RandomStream stream2 = new MRG32k3a();

double u = stream1.nextDouble(); ....

stream1.resetNextSubstream(); ....
stream1.resetStartSubStream();
stream1.resetStartStream();
```

1



Sauter en avant

$$x_n = (a_1 x_{n-1} + \cdots + a_k x_{n-k}) \bmod m, \qquad u_n = x_n/m.$$

État à l'étape n: $s_n = \boldsymbol{x}_n = (x_{n-k+1}, \ldots, x_n)^{\mathrm{t}}$.

$$\boldsymbol{x}_n = \boldsymbol{A}\boldsymbol{x}_{n-1} \bmod m = \begin{pmatrix} 0 & 1 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \\ a_k & a_{k-1} & \cdots & a_1 \end{pmatrix} \boldsymbol{x}_{n-1} \bmod m.$$

Sauter en avant

$$x_n = (a_1 x_{n-1} + \cdots + a_k x_{n-k}) \bmod m, \qquad u_n = x_n/m.$$

État à l'étape n: $s_n = \boldsymbol{x}_n = (x_{n-k+1}, \ldots, x_n)^{\mathrm{t}}$.

$$\boldsymbol{x}_n = \boldsymbol{A}\boldsymbol{x}_{n-1} \bmod m = \begin{pmatrix} 0 & 1 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \\ a_k & a_{k-1} & \cdots & a_1 \end{pmatrix} \boldsymbol{x}_{n-1} \bmod m.$$

$$\boldsymbol{x}_{n+\nu} = \boldsymbol{A}^\nu \boldsymbol{x}_n \bmod m = (\boldsymbol{A}^\nu \bmod m)\boldsymbol{x}_n \bmod m.$$

On peut précalculer $\boldsymbol{A}^\nu \bmod m$ via

$$\boldsymbol{A}^\nu \bmod m = \begin{cases} (\boldsymbol{A}^{\nu/2} \bmod m)(\boldsymbol{A}^{\nu/2} \bmod m) \bmod m & \text{si } \nu \text{ est pair;} \\ \boldsymbol{A}(\boldsymbol{A}^{\nu-1} \bmod m) \bmod m & \text{si } \nu \text{ est impair.} \end{cases}$$

Fonctionne pour tout $m > 1$. Utilisé pour les grands m et aussi $m = 2$.

Variables aléatoires non uniformes: Inversion

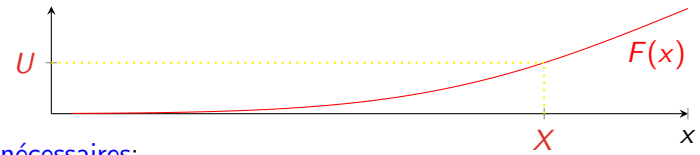
Une v.a. X a la fonction de répartition (cdf) F si $F(x) = \mathbb{P}[X \leq x]$ pour tout $x \in \mathbb{R}$.

Si $U \sim \mathcal{U}(0,1)$ (une v.a. uniforme sur $(0,1)$) et

$$X = F^{-1}(U) = \min\{x \mid F(x) \geq U\},$$

alors X est une v.a. dont la cdf est F.

Preuve: $\mathbb{P}[X \leq x] = \mathbb{P}[F^{-1}(U) \leq x] = \mathbb{P}[U \leq F(x)] = F(x)$.



Deux ingrédients nécessaires:

1. Un bon générateur uniforme pour générer U;
2. Une formule ou une bonne approximation pour calculer rapidement $F^{-1}(U)$.

Exemple: Loi de Weibull

$X \sim \text{Weibull}(\alpha, \lambda)$.

$$F(x) = 1 - \exp[-(\lambda x)^{\alpha}] \qquad \text{pour } x > 0.$$

Exemple: Loi de Weibull

$X \sim$ Weibull(α, λ).

$$F(x) = 1 - \exp[-(\lambda x)^{\alpha}] \qquad \text{pour } x > 0.$$

Inversion:

$$U = 1 - \exp[-(\lambda X)^{\alpha}]$$

Exemple: Loi de Weibull

$X \sim$ Weibull(α, λ).

$$F(x) = 1 - \exp[-(\lambda x)^{\alpha}] \qquad \text{pour } x > 0.$$

Inversion:

$$\begin{aligned} U &= 1 - \exp[-(\lambda X)^{\alpha}] \\ \exp[-(\lambda X)^{\alpha}] &= 1 - U \end{aligned}$$

Exemple: Loi de Weibull

$X \sim \text{Weibull}(\alpha, \lambda)$.

$$F(x) = 1 - \exp[-(\lambda x)^\alpha] \qquad \text{pour } x > 0.$$

Inversion:

$$\begin{aligned} U &= 1 - \exp[-(\lambda X)^\alpha] \\ \exp[-(\lambda X)^\alpha] &= 1 - U \\ (\lambda X)^\alpha &= -\ln(1-U) \end{aligned}$$

Exemple: Loi de Weibull

$X \sim \text{Weibull}(\alpha, \lambda)$.

$$F(x) = 1 - \exp[-(\lambda x)^\alpha] \qquad \text{pour } x > 0.$$

Inversion:

$$\begin{aligned} U &= 1 - \exp[-(\lambda X)^\alpha] \\ \exp[-(\lambda X)^\alpha] &= 1 - U \\ (\lambda X)^\alpha &= -\ln(1-U) \\ \lambda X &= [-\ln(1-U)]^{1/\alpha} \end{aligned}$$

Exemple: Loi de Weibull

$X \sim \text{Weibull}(\alpha, \lambda)$.

$$F(x) = 1 - \exp[-(\lambda x)^\alpha] \qquad \text{pour } x > 0.$$

Inversion:

$$\begin{aligned} U &= 1 - \exp[-(\lambda X)^\alpha] \\ \exp[-(\lambda X)^\alpha] &= 1 - U \\ (\lambda X)^\alpha &= -\ln(1-U) \\ \lambda X &= [-\ln(1-U)]^{1/\alpha} \\ X &= [-\ln(1-U)]^{1/\alpha}/\lambda \;=\; F^{-1}(U). \end{aligned}$$

Exemple: Loi de Weibull

$X \sim$ Weibull(α, λ).

$$F(x) = 1 - \exp[-(\lambda x)^\alpha] \qquad \text{pour } x > 0.$$

Inversion:

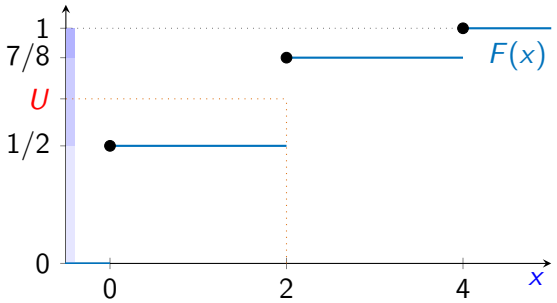
$$\begin{aligned} U &= 1 - \exp[-(\lambda X)^\alpha] \\ \exp[-(\lambda X)^\alpha] &= 1 - U \\ (\lambda X)^\alpha &= -\ln(1-U) \\ \lambda X &= [-\ln(1-U)]^{1/\alpha} \\ X &= [-\ln(1-U)]^{1/\alpha}/\lambda \;=\; F^{-1}(U). \end{aligned}$$

Cas particulier: si $\alpha = 1$, $X \sim$ Exponentielle(λ).
Pour générer: $X = F^{-1}(U) = -\ln(1-U)/\lambda$.

Exemple: Une loi discrète

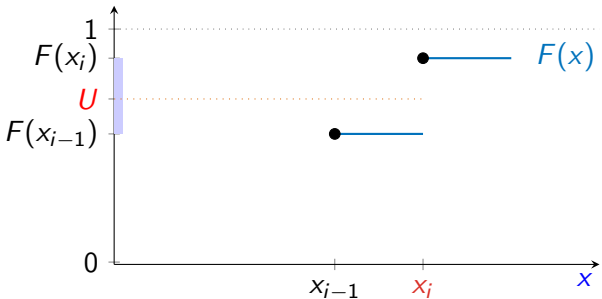
Soit $\mathbb{P}[X = i] = p_i$ où $p_0 = 1/2$, $p_2 = 3/8$, $p_4 = 1/8$, et $p_i = 0$ ailleurs.

Inversion: retourner 0 si $U < 1/2$, 2 si $1/2 \leq U < 7/8$, et 4 si $U \geq 7/8$.



Exemple: Loi discrète

$\mathbb{P}[X = x_i] = p_i$ pour $i = 0, 1, \ldots$.
On a $F(x_i) = p_0 + \cdots + p_i$ et (disons) $F(x_{-1}) = 0$.
L'inversion retourne $X = x_i$ ssi $F(x_i) \geq U > F(x_{i-1})$.



Algorithme: Générer $U \sim \mathcal{U}(0,1)$, trouver $I = \min\{i | F(x_i) \geq U\}$, et retourner x_I.

Mais trouver I en essayant $i = 0, 1, 2, 3, \ldots$ (recherche séquentielle) peut être très long.

Recherche par index

On partitionne $(0, 1)$ en c intervalles de longueur $1/c$.
On calcule et mémorise $i_s = \inf\{i : F(x_i) \geq s/c\}$ pour $s = 0, \ldots, c-1$.
Si $s = \lfloor cU \rfloor$, alors $U \in [s/c, (s+1)/c)$ et on a $I := F^{-1}(U) \in \{i_s, \ldots, i_{s+1}\}$.
On calcule et mémorise aussi les $F(x_i)$ pour $i = 0, 1, 2, \ldots$.

Recherche par index

On partitionne $(0,1)$ en c intervalles de longueur $1/c$.
On calcule et mémorise $i_s = \inf\{i : F(x_i) \geq s/c\}$ pour $s = 0, \ldots, c-1$.
Si $s = \lfloor cU \rfloor$, alors $U \in [s/c, (s+1)/c)$ et on a $I := F^{-1}(U) \in \{i_s, \ldots, i_{s+1}\}$.
On calcule et mémorise aussi les $F(x_i)$ pour $i = 0, 1, 2, \ldots$.

Il suffit alors de chercher dans cet intervalle, par recherche séquentielle ou binaire.

Inversion avec recherche par index (combiné avec recherche séquentielle);
générer $U \sim \mathcal{U}(0,1)$; poser $s = \lfloor cU \rfloor$ et $i = i_s$;
tant que $F(x_i) < U$ faire $i = i + 1$;
retourner x_i.

Recherche par index

On partitionne $(0, 1)$ en c intervalles de longueur $1/c$.
On calcule et mémorise $i_s = \inf\{i : F(x_i) \geq s/c\}$ pour $s = 0, \ldots, c-1$.
Si $s = \lfloor cU \rfloor$, alors $U \in [s/c, (s+1)/c)$ et on a $I := F^{-1}(U) \in \{i_s, \ldots, i_{s+1}\}$.
On calcule et mémorise aussi les $F(x_i)$ pour $i = 0, 1, 2, \ldots$.

Il suffit alors de chercher dans cet intervalle, par recherche séquentielle ou binaire.

Inversion avec recherche par index (combiné avec recherche séquentielle);

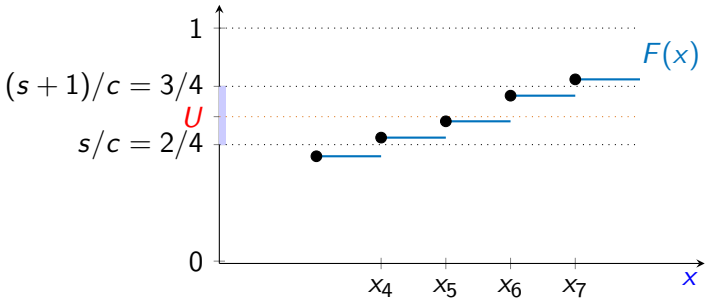
générer $U \sim \mathcal{U}(0,1)$; poser $s = \lfloor cU \rfloor$ et $i = i_s$;
tant que $F(x_i) < U$ faire $i = i + 1$;
retourner x_i.

Le nombre espéré d'itérations du "tant que" est environ k/c.
Si on choisit $c \approx k$, par exemple, alors on a un algorithme super rapide.
On paye un peu en terme de mémoire.

En pratique, on pourra prendre c jusqu'à $2^{12} = 4096$, par exemple.

Recherche par index

Posons $s = \lfloor cU \rfloor$.
On a $U \in [s/c, (s+1)/c)$ et $I := F^{-1}(U) \in \{i_s, \ldots, i_{s+1}\}$.



Pour ce U, on a $c = 4$, $s = 2$, $i_s = 4$, $i_{s+1} = 7$, donc $I \in \{4, 5, 6, 7\}$.

Exemple: Inversion pour la loi géométrique

$X \sim$ Géométrique(p), où $0 < p < 1$.
$\mathbb{P}[X = x] = p(1-p)^x$ pour $x = 0, 1, 2, \ldots$ et
$F(x) = 1 - (1-p)^{\lfloor x+1 \rfloor}$ pour $x \geq 0$.

Exemple: Inversion pour la loi géométrique

$X \sim$ Géométrique(p), où $0 < p < 1$.
$\mathbb{P}[X = x] = p(1-p)^x$ pour $x = 0, 1, 2, \ldots$ et
$F(x) = 1 - (1-p)^{\lfloor x+1 \rfloor}$ pour $x \geq 0$.

Pour $x \geq 0$ entier, $F(x) = 1 - (1-p)^{x+1}$.
L'inversion doit retourner $X = x$ ssi $F(x) \geq U > F(x-1)$,

Exemple: Inversion pour la loi géométrique

$X \sim$ Géométrique(p), où $0 < p < 1$.
$\mathbb{P}[X = x] = p(1-p)^x$ pour $x = 0, 1, 2, \ldots$ et
$F(x) = 1 - (1-p)^{\lfloor x+1 \rfloor}$ pour $x \geq 0$.

Pour $x \geq 0$ entier, $F(x) = 1 - (1-p)^{x+1}$.
L'inversion doit retourner $X = x$ ssi $F(x) \geq U > F(x-1)$,
que l'on peut réécrire

$$1 - (1-p)^{x+1} \;\geq\; U \;>\; 1 - (1-p)^x,$$

Exemple: Inversion pour la loi géométrique

$X \sim$ Géométrique(p), où $0 < p < 1$.
$\mathbb{P}[X = x] = p(1-p)^x$ pour $x = 0, 1, 2, \ldots$ et
$F(x) = 1 - (1-p)^{\lfloor x+1 \rfloor}$ pour $x \geq 0$.

Pour $x \geq 0$ entier, $F(x) = 1 - (1-p)^{x+1}$.
L'inversion doit retourner $X = x$ ssi $F(x) \geq U > F(x-1)$,
que l'on peut réécrire

$$1 - (1-p)^{x+1} \;\geq\; U \;>\; 1 - (1-p)^x, \qquad \text{ou}$$
$$(1-p)^{x+1} \;\leq\; 1 - U \;<\; (1-p)^x,$$

Exemple: Inversion pour la loi géométrique

$X \sim$ Géométrique(p), où $0 < p < 1$.
$\mathbb{P}[X = x] = p(1-p)^x$ pour $x = 0, 1, 2, \ldots$ et
$F(x) = 1 - (1-p)^{\lfloor x+1 \rfloor}$ pour $x \geq 0$.

Pour $x \geq 0$ entier, $F(x) = 1 - (1-p)^{x+1}$.
L'inversion doit retourner $X = x$ ssi $F(x) \geq U > F(x-1)$,
que l'on peut réécrire

$$
\begin{aligned}
1 - (1-p)^{x+1} \;\geq\; U \;>\; 1 - (1-p)^x, &\qquad \text{ou} \\
(1-p)^{x+1} \;\leq\; 1 - U \;<\; (1-p)^x, &\qquad \text{ou} \\
(x+1)\ln(1-p) \;\leq\; \ln(1-U) \;<\; x\ln(1-p), &
\end{aligned}
$$

Exemple: Inversion pour la loi géométrique

$X \sim$ Géométrique(p), où $0 < p < 1$.
$\mathbb{P}[X = x] = p(1-p)^x$ pour $x = 0, 1, 2, \ldots$ et
$F(x) = 1 - (1-p)^{\lfloor x+1 \rfloor}$ pour $x \geq 0$.

Pour $x \geq 0$ entier, $F(x) = 1 - (1-p)^{x+1}$.
L'inversion doit retourner $X = x$ ssi $F(x) \geq U > F(x-1)$,
que l'on peut réécrire

$$
\begin{aligned}
1 - (1-p)^{x+1} \;\geq\; U \;>\; 1 - (1-p)^x, &\qquad \text{ou} \\
(1-p)^{x+1} \;\leq\; 1 - U \;<\; (1-p)^x, &\qquad \text{ou} \\
(x+1)\ln(1-p) \;\leq\; \ln(1-U) \;<\; x\ln(1-p), &\qquad \text{ou} \\
(x+1) \;\geq\; \ln(1-U)/\ln(1-p) \;>\; x, &\qquad (\text{car } \ln(1-p) < 0)
\end{aligned}
$$

ce qui donne $x + 1 = \lceil \ln(1-U)/\ln(1-p) \rceil$.
Avec probabilité 1, c'est la même chose que retourner $X = \lfloor \ln(1-U)/\ln(1-p) \rfloor$.

Autres situations

- Dans plusieurs cas (normale, Student, chi-deux, etc.), pas de formule pour F^{-1}, mais approximation numérique.

Autres situations

- Dans plusieurs cas (normale, Student, chi-deux, etc.), pas de formule pour F^{-1}, mais approximation numérique.
- Plus difficile lorsque la forme de F dépend des paramètres (beta, gamma, par exemple).

Autres situations

- Dans plusieurs cas (normale, Student, chi-deux, etc.), pas de formule pour F^{-1}, mais approximation numérique.
- Plus difficile lorsque la forme de F dépend des paramètres (beta, gamma, par exemple).
- Inversion préférable car monotone (on verra pourquoi plus tard).

Autres situations

- Dans plusieurs cas (normale, Student, chi-deux, etc.), pas de formule pour F^{-1}, mais approximation numérique.
- Plus difficile lorsque la forme de F dépend des paramètres (beta, gamma, par exemple).
- Inversion préférable car monotone (on verra pourquoi plus tard).
- Mais d'autres méthodes sont parfois beaucoup plus rapides.

Autres situations

- Dans plusieurs cas (normale, Student, chi-deux, etc.), pas de formule pour F^{-1}, mais approximation numérique.
- Plus difficile lorsque la forme de F dépend des paramètres (beta, gamma, par exemple).
- Inversion préférable car monotone (on verra pourquoi plus tard).
- Mais d'autres méthodes sont parfois beaucoup plus rapides.

Inversion pour la loi de Y conditionnelle à $Y \in (a, b]$.

Théorème: Soit F la cdf d'une variable aléatoire Y et soit $-\infty \leq a < b \leq \infty$. Si $U \sim \text{Uniforme}(F(a), F(b))$ et $X = F^{-1}(U)$, alors la loi de probabilité de X (sans condition) est la même que la loi de Y conditionnelle à $Y \in (a, b]$.

Preuve: Puisque $F(X) = U$, on a

$$\mathbb{P}(X \leq x) = \mathbb{P}(U \leq F(x)) = \frac{F(x)-F(a)}{F(b)-F(a)} \quad \text{pour } F(a) \leq F(x) \leq F(b).$$

C'est la même cdf que la cdf de Y conditionnelle à $Y \in (a, b]$:

$$\mathbb{P}[Y \leq y \mid a < Y \leq b] = \frac{F(y)-F(a)}{F(b)-F(a)} \quad \text{pour } a \leq y \leq b.$$

Méthode de rejet

Technique la plus importante après l'inversion.
Peut fournir une solution efficace lorsque l'inversion est trop difficile ou coûteuse.

On veut générer X selon une densité f. La surface sous f est:

$$\mathcal{S}(f) = \{(x, y) \in \mathbb{R}^2 : 0 \leq y \leq f(x)\}.$$

Proposition. Si le point (X, Y) est uniforme sur $\mathcal{S}(f)$, alors X a la densité f.

Preuve. Si (X, Y) est uniforme sur $\mathcal{S}(f)$, alors $\mathbb{P}[X \leq x]$ est égal à la surface de $\{(z, y) \in \mathcal{S}(f) : z \leq x\}$, qui est $\int_{-\infty}^{x} f(z)dz$. $\square$

Méthode de rejet

Technique la plus importante après l'inversion.
Peut fournir une solution efficace lorsque l'inversion est trop difficile ou coûteuse.

On veut générer X selon une densité f. La surface sous f est:

$$\mathcal{S}(f) = \{(x, y) \in \mathbb{R}^2 : 0 \leq y \leq f(x)\}.$$

Proposition. Si le point (X, Y) est uniforme sur $\mathcal{S}(f)$, alors X a la densité f.

Preuve. Si (X, Y) est uniforme sur $\mathcal{S}(f)$, alors $\mathbb{P}[X \leq x]$ est égal à la surface de $\{(z, y) \in \mathcal{S}(f) : z \leq x\}$, qui est $\int_{-\infty}^{x} f(z)dz$. $\square$

On va générer (X, Y) uniformément sur $\mathcal{S}(f)$. Comment, si $\mathcal{S}(f)$ est compliqué?

Méthode de rejet

Technique la plus importante après l'inversion.
Peut fournir une solution efficace lorsque l'inversion est trop difficile ou coûteuse.

On veut générer X selon une densité f. La surface sous f est:

$$\mathcal{S}(f) = \{(x, y) \in \mathbb{R}^2 : 0 \leq y \leq f(x)\}.$$

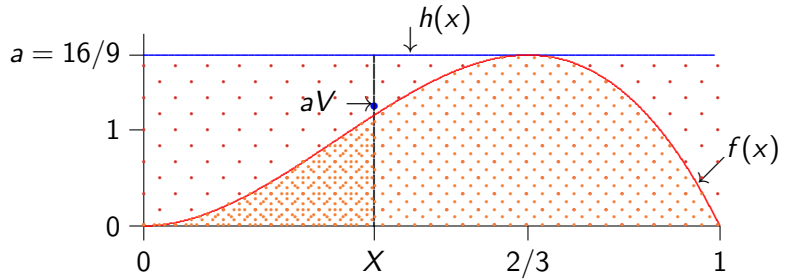
Proposition. Si le point (X, Y) est uniforme sur $\mathcal{S}(f)$, alors X a la densité f.

Preuve. Si (X, Y) est uniforme sur $\mathcal{S}(f)$, alors $\mathbb{P}[X \leq x]$ est égal à la surface de $\{(z, y) \in \mathcal{S}(f) : z \leq x\}$, qui est $\int_{-\infty}^{x} f(z)dz$. $\square$

On va générer (X, Y) uniformément sur $\mathcal{S}(f)$. Comment, si $\mathcal{S}(f)$ est compliqué?

Idée: Choisir une surface simple $\mathcal{B}$ qui contient $\mathcal{S}(f)$, et générer (X, Y) uniformément dans $\mathcal{B}$. Si $(X, Y) \in \mathcal{S}(f)$, c'est bon, sinon on recommence.
On va montrer que le point (X, Y) retenu suit une loi uniforme sur $\mathcal{S}(f)$.

Exemple: On veut générer $X \sim \text{Beta}(3, 2)$, de densité $f(x) = 12x^2(1-x)$ sur $(0, 1)$.
La densité est maximale à $x_* = 2/3$. On a $f(2/3) = 16/9 \approx 1.77778$.
Ici, $\mathcal{S}(f)$ est la surface sous la courbe en rouge.
Alors on peut prendre $\mathcal{B} = \{(x, y) : 0 \le x \le 1,\ 0 \le y \le 16/9\}$ (un rectangle).



Pour générer un point dans $\mathcal{B}$, on génère $U, V \sim \mathcal{U}(0, 1)$ indépendantes, et on pose $(X, Y) = (U, aV)$ où $a = 16/9$.

La probabilité que le point soit dans $\mathcal{S}(f)$ est $1/a = 9/16$.
Le nombre espéré de points (X, Y) qu'il faudra générer est $a = 16/9$.

Méthode de rejet générale

On veut générer un point $\boldsymbol{X}$ uniformément dans un ensemble $\mathcal{A} \subset \mathbb{R}^d$.

On choisit un ensemble $\mathcal{B}$ plus "simple" tel que $\mathcal{A} \subset \mathcal{B}$.

On génère des points indépendants dans $\mathcal{B}$, et on retient le premier qui tombe dans $\mathcal{A}$.

Proposition. Le point $\boldsymbol{X}$ retenu suit la loi uniforme sur $\mathcal{A}$.

Preuve. On veut montrer que pour tout $\mathcal{D} \subseteq \mathcal{A}$, on a $\mathbb{P}[\boldsymbol{X} \in \mathcal{D} \mid \boldsymbol{X} \in \mathcal{A}] = \mathrm{vol}(\mathcal{D})/\mathrm{vol}(\mathcal{A})$. Puisque $\boldsymbol{X}$ est uniforme sur $\boldsymbol{B}$, on a $\mathbb{P}[\boldsymbol{X} \in \mathcal{D}] = \mathrm{vol}(\mathcal{D})/\mathrm{vol}(\mathcal{B})$ et donc

$$\mathbb{P}[\boldsymbol{X} \in \mathcal{D} \mid \boldsymbol{X} \in \mathcal{A}] = \frac{\mathbb{P}[\boldsymbol{X} \in \mathcal{D} \cap \mathcal{A}]}{\mathbb{P}[\boldsymbol{X} \in \mathcal{A}]} = \frac{\mathrm{vol}(\mathcal{D})/\mathrm{vol}(\mathcal{B})}{\mathrm{vol}(\mathcal{A})/\mathrm{vol}(\mathcal{B})} = \frac{\mathrm{vol}(\mathcal{D})}{\mathrm{vol}(\mathcal{A})}.$$

Ainsi la loi de $\boldsymbol{X}$ conditionelle à $\boldsymbol{X} \in \mathcal{A}$ est uniforme sur $\mathcal{A}$.

Méthode de rejet avec fonction chapeau

Pour générer X selon f, on choisit une autre densité g et une constante $a \geq 1$ telle que

$$f(x) \leq h(x) \stackrel{\text{def}}{=} ag(x)$$

pour tout x, et telle qu'il est facile de générer X selon g. La fonction chapeau est h.

Méthode de rejet avec fonction chapeau

Pour générer X selon f, on choisit une autre densité g et une constante $a \geq 1$ telle que

$$f(x) \leq h(x) \stackrel{\text{def}}{=} ag(x)$$

pour tout x, et telle qu'il est facile de générer X selon g. La fonction chapeau est h.

On applique la méthode de rejet avec $\mathcal{A} = \mathcal{S}(f)$ et

$$\mathcal{B} = \mathcal{S}(h) = \{(x, y) \in \mathbb{R}^2 : 0 \leq y \leq h(x)\}, \qquad \text{la surface sous } h.$$

Algorithme de rejet;

répéter

générer X selon la densité g et $V \sim \mathcal{U}(0, 1)$, indépendants;

// Le point $(X, h(X)V)$ est uniforme sur $\mathcal{S}(h)$.

jusqu'à ce que $V\, h(X) \leq f(X)$;

retourner X. // Le point X retourné est uniforme sur $\mathcal{A} = \mathcal{S}(f)$.

Proposition. La v.a. X retournée a la densité f.

Algorithme de rejet;

répéter

générer X selon la densité g et $V \sim \mathcal{U}(0,1)$, indépendants;

jusqu'à ce que $V\, h(X) \leq f(X)$;

retourner X.

À chaque tour de boucle, la probabilité d'accepter X est $1/a$.
Le nombre R de rejets avant l'acceptation est une v.a. géométrique de paramètre $p = 1/a$.
Le nombre moyen de tours de boucle requis par v.a. est donc $1/p = a$.

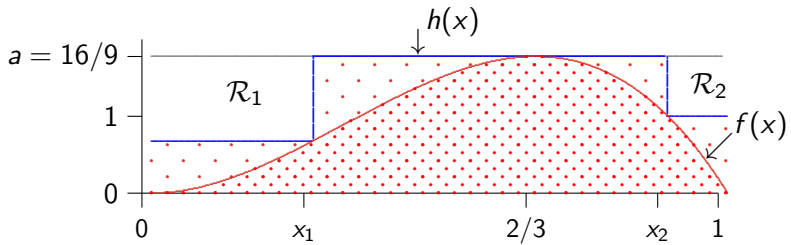
On veut $a \geq 1$ le plus petit possible.
Compromis entre diminuer a et garder g simple.

Exemple: $X \sim \mathrm{Beta}(3,2)$ (suite). Baisser le chapeau.

Pour réduire a, on peut prendre la fonction chapeau:

$$h(x) = \begin{cases} f(x_1) & \text{pour } x < x_1; \\ 16/9 & \text{pour } x_1 \le x \le x_2; \\ f(x_2) & \text{pour } x > x_2, \end{cases}$$

où x_1 et x_2 satisfont $0 < x_1 < 2/3 < x_2 < 1$.



On génère un point au hasard dans la surface sous h et on l'accepte s'il est sous la courbe rouge.

La surface sous h est minimisée en prenant $x_1 = 0.281023$ et $x_2 = 0.89538$. Elle est alors réduite de 1.77778 à 1.38997. La probabilité d'accepter passe de $1/1.77778$ à $1/1.38997$.

La fonction de répartition inverse de g est linéaire par morceaux.

Pourquoi ne pas prendre une fonction h linéaire par morceaux au lieu de constante par morceaux? Le calcul de G^{-1} demande alors des racines carrées... et on perd en efficacité.

Point au hasard sur la surface d'une hypersphère

Sphère de rayon 1 centrée à l'origine, en d dimensions.
Il suffit de génener un point dans $\mathbb{R}^d$, selon une densité radialement symétrique (i.e., qui ne dépend que de la distance à l'origine). Par exemple, la densité multinormale standard.

Point au hasard sur la surface d'une hypersphère

Sphère de rayon 1 centrée à l'origine, en d dimensions.
Il suffit de génener un point dans $\mathbb{R}^d$, selon une densité radialement symétrique (i.e., qui ne dépend que de la distance à l'origine). Par exemple, la densité multinormale standard.

On génère $\boldsymbol{Z} = (Z_1, \ldots, Z_d) \sim N(\boldsymbol{0}, \boldsymbol{I})$ et on pose $\boldsymbol{X} = \boldsymbol{Z}/\|\boldsymbol{Z}\|_2$, où $\|\boldsymbol{Z}\|_2^2 = \sum_{j=1}^d Z_j^2$.
Ici, $Z_1, \ldots, Z_d$ sont indépendantes et $N(0, 1)$.

Point au hasard sur la surface d'une hypersphère

Sphère de rayon 1 centrée à l'origine, en d dimensions.
Il suffit de généner un point dans $\mathbb{R}^d$, selon une densité radialement symétrique (i.e., qui ne dépend que de la distance à l'origine). Par exemple, la densité multinormale standard.

On génère $\boldsymbol{Z} = (Z_1, \ldots, Z_d) \sim N(\boldsymbol{0}, \boldsymbol{I})$ et on pose $\boldsymbol{X} = \boldsymbol{Z}/\|\boldsymbol{Z}\|_2$, où $\|\boldsymbol{Z}\|_2^2 = \sum_{j=1}^d Z_j^2$.
Ici, $Z_1, \ldots, Z_d$ sont indépendantes et $N(0,1)$.

En deux dimensions: on peut générer Θ uniformément sur $(0, 2\pi)$, puis poser $(X_1, X_2) = (\cos\Theta, \sin\Theta)$.

Point au hasard sur la surface d'une hypersphère

Sphère de rayon 1 centrée à l'origine, en d dimensions.
Il suffit de généner un point dans $\mathbb{R}^d$, selon une densité radialement symétrique (i.e., qui ne dépend que de la distance à l'origine). Par exemple, la densité multinormale standard.

On génère $\boldsymbol{Z} = (Z_1, \ldots, Z_d) \sim N(\boldsymbol{0}, \boldsymbol{I})$ et on pose $\boldsymbol{X} = \boldsymbol{Z}/\|\boldsymbol{Z}\|_2$, où $\|\boldsymbol{Z}\|_2^2 = \sum_{j=1}^d Z_j^2$.
Ici, $Z_1, \ldots, Z_d$ sont indépendantes et $N(0,1)$.

En deux dimensions: on peut générer Θ uniformément sur $(0, 2\pi)$, puis poser $(X_1, X_2) = (\cos\Theta, \sin\Theta)$.

En trois dimensions: chaque coordonnée X_j est uniforme sur $(-1, 1)$!
On peut générer $X_3 \sim \mathcal{U}(-1, 1)$, puis (X_1, X_2) sur le cercle qui reste, dont le rayon est $\tilde{r} = \cos(\arcsin x_3)$.
Pour cela, générer $\Theta \sim \mathcal{U}(0, 2\pi)$, et poser $(X_1, X_2) = (\tilde{r}\cos\Theta, \tilde{r}\sin\Theta)$.

Méthode Monte Carlo

Intégration par Monte Carlo: Monte Carlo est souvent utilisé pour estimer une espérance

$$\mu = \mathbb{E}[X]$$

où X est une v.a. dont chaque réalisation est calculée par simulation. On peut interpréter X comme une fonction d'un vecteur $\boldsymbol{U} = (U_0, \ldots, U_{s-1})$ de s v.a. i.i.d. $\mathcal{U}(0,1)$,
$X = f(\boldsymbol{U}) = f(U_1, \ldots, U_s)$, et on a

$$\mu = \mathbb{E}[X] = \mathbb{E}[f(\boldsymbol{U})] = \int_{[0,1)^s} f(\boldsymbol{u}) d\boldsymbol{u} = \int_0^1 \cdots \int_0^1 f(u_0, \ldots, u_{s-1}) du_0 \cdots du_{s-1},$$

La dimension s peut parfois être aléatoire ou infinie.
La fonction f est souvent très compliquée.
μ est la valeur moyenne de f sur $[0,1)^s$.

Rendu ici, 4 nov. 2024

Exemple: réseau d'activités stochastique à 13 activités. On a $s = 13$.
À un point $\boldsymbol{U} = (U_0, \ldots, U_{12}) \in [0,1)^{13}$ correspond un vecteur de durées $(Y_0, \ldots, Y_{12}) = (F_0^{-1}(U_0), \ldots, F_{12}^{-1}(U_{12}))$ et une durée du projet T.

Exemple: réseau d'activités stochastique à 13 activités. On a $s = 13$.
À un point $\boldsymbol{U} = (U_0, \ldots, U_{12}) \in [0,1)^{13}$ correspond un vecteur de durées $(Y_0, \ldots, Y_{12}) = (F_0^{-1}(U_0), \ldots, F_{12}^{-1}(U_{12}))$ et une durée du projet T.

Pour estimer $\mu = \mathbb{E}[T]$, on peut définir $f(U_0, \ldots, U_{12}) = T$.
Pour estimer $\mu = \mathbb{P}[T > x] = \mathbb{E}[\mathbb{I}(T > x)]$, on peut définir $f(U_0, \ldots, U_{12}) = \mathbb{I}(T > x)$.
Rappel: $\mathbb{I}(T > x) = 1$ si $T > x$ et $= 0$ sinon.
Dans les deux cas, on a une intégrale à 13 dimensions.

Si s ne dépasse pas 2 ou 3 et f est assez lisse, peut utiliser des méthodes d'intégration numériques (méthode du trapèze, méthode de Simpson, etc.).

Si s ne dépasse pas 2 ou 3 et f est assez lisse, peut utiliser des méthodes d'intégration numériques (méthode du trapèze, méthode de Simpson, etc.).

Si s est grand, estimateur de Monte Carlo de $\mu = \mathbb{E}[X]$ pour $X = f(\boldsymbol{U})$:

$$\hat{\mu}_n = \frac{1}{n}\sum_{i=1}^{n} X_i \qquad \text{où } X_i = f(\boldsymbol{U}_i),$$

n est la taille de l'échantillon (nombre de simulations) et $\boldsymbol{U}_1, \ldots, \boldsymbol{U}_n$ sont i.i.d. uniformes sur $[0,1)^s$.

$$\begin{aligned}
\mathbb{E}[\hat{\mu}_n] &= \mathbb{E}[f(\boldsymbol{U}_i)] = \int_{[0,1)^s} f(\boldsymbol{u})d\boldsymbol{u} = \mu = \mathbb{E}[X] \\
\mathrm{Var}[\hat{\mu}_n] &= \frac{1}{n}\left[\mathbb{E}[f^2(\boldsymbol{U}_i)] - \mu^2\right] = \frac{1}{n}\mathrm{Var}[X] = \frac{\sigma^2}{n} \\
\text{où } \sigma^2 &= \int_{[0,1)^s} f^2(\boldsymbol{u})d\boldsymbol{u} - \mu^2 = \mathrm{Var}[f(\boldsymbol{U}_i)] = \mathrm{Var}[X]
\end{aligned}$$

Convergence

Théorème. Supposons que $\sigma^2 < \infty$. Lorsque $n \to \infty$:

(i) Loi forte des grands nombres: $\lim_{n\to\infty} \hat{\mu}_n = \mu$ avec probabilité 1.

Convergence

Théorème. Supposons que $\sigma^2 < \infty$. Lorsque $n \to \infty$:

(i) Loi forte des grands nombres: $\lim_{n\to\infty} \hat{\mu}_n = \mu$ avec probabilité 1.

(ii) Théorème de la limite centrale:

$$\frac{\sqrt{n}(\hat{\mu}_n - \mu)}{\sigma} \Rightarrow N(0,1), \text{ i.e.,}$$

$$\lim_{n\to\infty} \mathbb{P}\left[\frac{\sqrt{n}(\hat{\mu}_n - \mu)}{\sigma} \le x\right] = \Phi(x) = \mathbb{P}[Z \le x]$$

pour tout $x \in \mathbb{R}$, où $Z \sim N(0,1)$ et $\Phi(\cdot)$ sa fonction de répartition.

Convergence

Théorème. Supposons que $\sigma^2 < \infty$. Lorsque $n \to \infty$:

(i) Loi forte des grands nombres: $\lim_{n\to\infty} \hat{\mu}_n = \mu$ avec probabilité 1.

(ii) Théorème de la limite centrale:

$$\frac{\sqrt{n}(\hat{\mu}_n - \mu)}{\sigma} \;\Rightarrow\; N(0,1), \text{ i.e.,}$$

$$\lim_{n\to\infty} \mathbb{P}\left[\frac{\sqrt{n}(\hat{\mu}_n - \mu)}{\sigma} \le x\right] = \Phi(x) = \mathbb{P}[Z \le x]$$

pour tout $x \in \mathbb{R}$, où $Z \sim N(0,1)$ et $\Phi(\cdot)$ sa fonction de répartition.

La propriété (ii) tient aussi si on remplace σ^2 par son estimateur sans biais

$$S_n^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X}_n)^2 = \frac{1}{n-1}\left(\sum_{i=1}^{n} X_i^2 \; - n(\bar{X}_n)^2\right),$$

où $X_i = f(\boldsymbol{U}_i)$. On a $\sqrt{n}(\hat{\mu}_n - \mu)/S_n \;\Rightarrow\; N(0,1)$. Donne une idée de la distrib. de l'erreur.

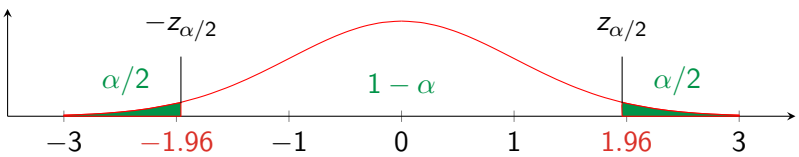
Pour n grand et un niveau de confiance $1-\alpha$, on a

$$\mathbb{P}[\hat{\mu}_n - \mu \le x S_n/\sqrt{n}] = \mathbb{P}[\sqrt{n}(\hat{\mu}_n - \mu)/S_n \le x] \approx \Phi(x).$$

Intervalle de confiance au niveau $1-\alpha$ (on veut $\Phi(x) = 1-\alpha/2$):

$$(\hat{\mu}_n \pm z_{\alpha/2} S_n/\sqrt{n}), \quad \text{où } \Phi(z_{\alpha/2}) = 1-\alpha/2.$$

Exemple: $z_{\alpha/2} \approx 1.96$ pour $\alpha = 0.05$.



La largeur de l'intervalle de confiance est asymptotiquement proportionnelle à $\sigma/\sqrt{n}$, donc converge en $O(n^{-1/2})$.

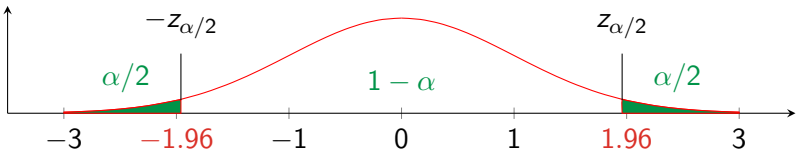
Pour n grand et un niveau de confiance $1-\alpha$, on a

$$\mathbb{P}[\hat{\mu}_n - \mu \le x S_n/\sqrt{n}] = \mathbb{P}[\sqrt{n}(\hat{\mu}_n - \mu)/S_n \le x] \approx \Phi(x).$$

Intervalle de confiance au niveau $1-\alpha$ (on veut $\Phi(x) = 1-\alpha/2$):

$$(\hat{\mu}_n \pm z_{\alpha/2} S_n/\sqrt{n}), \quad \text{où } \Phi(z_{\alpha/2}) = 1-\alpha/2.$$

Exemple: $z_{\alpha/2} \approx 1.96$ pour $\alpha = 0.05$.



La largeur de l'intervalle de confiance est asymptotiquement proportionnelle à $\sigma/\sqrt{n}$, donc converge en $O(n^{-1/2})$.

Si n est petit et les X_i suivent la loi normale, alors $\sqrt{n}(\hat{\mu}_n - \mu)/S_n \sim$ Student$(n-1)$.

Si la loi des X_i est très asymétrique, ces intervalles ne sont plus valides même comme approximations.

Exemple: Réseau d'activités stochastique. Pour $\mu = \mathbb{P}[T > x]$, on a

$$\begin{aligned} X_i &= \mathbb{I}[T_i > x], \\ \hat{\mu}_n &= \bar{X}_n = \frac{1}{n}\sum_{i=1}^{n} X_i = \frac{Y}{n}, \\ S_n^2 &= \frac{1}{n-1}\sum_{i=1}^{n}(X_i - Y/n)^2 = \frac{Y(1-Y/n)}{n-1}. \end{aligned}$$

Intervalle de confiance à 95%: $(Y/n \pm 1.96 S_n/\sqrt{n})$.
Raisonnable si μ n'est pas trop proche de 0 ou 1.

En fait, $Y = \sum_{i=1}^{n} X_i \sim$ Binomiale(n, μ).

Supposons par ex. que $n = 1000$ et qu'on observe $Y = 882$. On a alors:

$\bar{X}_n = 882/1000 = 0.882$;

$S_n^2 = \bar{X}_n(1 - \bar{X}_n)n/(n-1) \approx 0.1042$.

Supposons par ex. que $n = 1000$ et qu'on observe $Y = 882$. On a alors:

$\bar{X}_n = 882/1000 = 0.882$;

$S_n^2 = \bar{X}_n(1 - \bar{X}_n)n/(n-1) \approx 0.1042$.

On obtient l'intervalle de confiance à 95%:

$(\bar{X}_n \pm 1.96 S_n/\sqrt{n}) \approx (0.882 \pm 0.020) = (0.862, 0.902)$.

Supposons par ex. que $n = 1000$ et qu'on observe $Y = 882$. On a alors:

$\bar{X}_n = 882/1000 = 0.882;$

$S_n^2 = \bar{X}_n(1 - \bar{X}_n)n/(n-1) \approx 0.1042.$

On obtient l'intervalle de confiance à 95%:

$(\bar{X}_n \pm 1.96 S_n/\sqrt{n}) \approx (0.882 \pm 0.020) = (0.862, 0.902).$

Notre estimateur de μ a donc deux chiffres significatifs: $\mu \approx 0.88$.
Le "2" dans 0.882 n'est pas significatif.

Supposons par ex. que $n = 1000$ et qu'on observe $Y = 882$. On a alors:

$\bar{X}_n = 882/1000 = 0.882$;

$S_n^2 = \bar{X}_n(1 - \bar{X}_n)n/(n-1) \approx 0.1042$.

On obtient l'intervalle de confiance à 95%:

$(\bar{X}_n \pm 1.96 S_n/\sqrt{n}) \approx (0.882 \pm 0.020) = (0.862, 0.902)$.

Notre estimateur de μ a donc deux chiffres significatifs: $\mu \approx 0.88$.
Le "2" dans 0.882 n'est pas significatif.

Il faut éviter de donner des chiffres non significatifs dans les résultats, car cela peut induire en erreur.

Avantages de MC pour l'intégration

- Ne requiert qu'une hypothèse très faible sur f.
- Le taux de convergence de l'erreur ne dépend pas de la dimension s, contrairement aux méthodes d'intégration numérique classiques.
- On peut estimer l'erreur de manière probabiliste (intervalles de confiance).
  Les méthodes d'intégration numériques donnent des bornes déterministes sur l'erreur, mais souvent on ne peut pas les calculer, donc peu pratiques.

Efficacité des estimateurs en simulation

Soit X estimateur de μ.

La variance $\mathrm{Var}[X]$ n'est pas la seule mesure de qualité de l'estimateur X.

$$
\begin{aligned}
\beta &= \mathbb{E}[X] - \mu \quad \text{biais} \\
\sigma^2 &= \mathrm{Var}(X) = \mathbb{E}[(X - \mathbb{E}[X])^2] \quad \text{variance} \\
\mathrm{MSE}[X] &= \mathbb{E}[(X-\mu)^2] = \beta^2 + \sigma^2 \quad \text{erreur quadratique moyenne} \\
&\sqrt{\mathrm{MSE}[X]} \quad \text{erreur absolue} \\
\mathrm{RE}[X] &= \sqrt{\mathrm{MSE}[X]}/|\mu|, \quad \text{pour } \mu \neq 0 \quad \text{erreur relative}
\end{aligned}
$$

Soit $C(X)$ l'espérance mathématique du temps de calcul de X.
Variance normalisée par le coût: $C(X) \cdot \mathrm{MSE}(X)$.
Efficacité de l'estimateur X:

$$\mathrm{Eff}(X) = \frac{1}{C(X) \cdot \mathrm{MSE}(X)}.$$

X est plus efficace que Y si $\mathrm{Eff}(X) > \mathrm{Eff}(Y)$.

Soit $C(X)$ l'espérance mathématique du temps de calcul de X.
Variance normalisée par le coût: $C(X) \cdot \mathrm{MSE}(X)$.
Efficacité de l'estimateur X:

$$\mathrm{Eff}(X) = \frac{1}{C(X) \cdot \mathrm{MSE}(X)}.$$

X est plus efficace que Y si $\mathrm{Eff}(X) > \mathrm{Eff}(Y)$.

Amélioration de l'efficacité: trouver des estimateurs plus efficaces, en ce sens, soit en diminuant la variance, ou le biais, ou le temps de calcul.

Soit $C(X)$ l'espérance mathématique du temps de calcul de X.
Variance normalisée par le coût: $C(X) \cdot \mathrm{MSE}(X)$.
Efficacité de l'estimateur X:

$$\mathrm{Eff}(X) \;=\; \frac{1}{C(X) \cdot \mathrm{MSE}(X)}.$$

X est plus efficace que Y si $\mathrm{Eff}(X) > \mathrm{Eff}(Y)$.

Amélioration de l'efficacité: trouver des estimateurs plus efficaces, en ce sens, soit en diminuant la variance, ou le biais, ou le temps de calcul.

Soit $X_1, \ldots, X_n$ i.i.d., $\mathbb{E}[X_i] = \mu$, et $C(\bar{X}_n) = \kappa n$. On a $\mathrm{Var}[\bar{X}_n] = \mathrm{Var}[X_i]/n = \sigma^2/n$ et

$$\mathrm{Eff}[\bar{X}_n] = \frac{1}{C(\bar{X}_n) \cdot \mathrm{MSE}(\bar{X}_n)} = \frac{1}{\kappa n \sigma^2/n} = \frac{1}{\kappa \sigma^2}.$$

Cette mesure d'efficacité ne dépend pas de n, ce qui est bien.

Exemple de difficulté: événements rares

On veut estimer $p = \mathbb{P}\{A\}$ où A est un événement rare (p est proche de 0).
La variable binaire $X = \mathbb{I}[A]$ est un estimateur sans biais de p,
de variance (et MSE) $\mathrm{Var}[X] = \mathrm{MSE}[X] = p(1-p)$.

Exemple de difficulté: événements rares

On veut estimer $p = \mathbb{P}\{A\}$ où A est un événement rare (p est proche de 0).
La variable binaire $X = \mathbb{I}[A]$ est un estimateur sans biais de p,
de variance (et MSE) $\mathrm{Var}[X] = \mathrm{MSE}[X] = p(1-p)$.
Si p est petit, $\mathrm{Var}[X] = \mathrm{MSE}[X] \approx p$ est petite.

Exemple de difficulté: événements rares

On veut estimer $p = \mathbb{P}\{A\}$ où A est un événement rare (p est proche de 0).
La variable binaire $X = \mathbb{I}[A]$ est un estimateur sans biais de p,
de variance (et MSE) $\mathrm{Var}[X] = \mathrm{MSE}[X] = p(1-p)$.
Si p est petit, $\mathrm{Var}[X] = \mathrm{MSE}[X] \approx p$ est petite.

Mais l'estimateur trivial $Y = 0$ donne déjà $\mathrm{Var}[Y] = 0$ et $\mathrm{MSE}[Y] = p^2$,
ce qui est encore plus petit et ne coûte rien! On veut faire mieux que cela.

Si on prend $\bar{X}_n$ comme estimateur, alors $\mathrm{Var}[\bar{X}_n] = p(1-p)/n$.
On a $\mathrm{MSE}[\bar{X}_n] < \mathrm{MSE}[Y]$ ssi $p(1-p)/n < p^2$ ssi $n > (1-p)/p$.

Exemple de difficulté: événements rares

On veut estimer $p = \mathbb{P}\{A\}$ où A est un événement rare (p est proche de 0).
La variable binaire $X = \mathbb{I}[A]$ est un estimateur sans biais de p,
de variance (et MSE) $\mathrm{Var}[X] = \mathrm{MSE}[X] = p(1-p)$.
Si p est petit, $\mathrm{Var}[X] = \mathrm{MSE}[X] \approx p$ est petite.

Mais l'estimateur trivial $Y = 0$ donne déjà $\mathrm{Var}[Y] = 0$ et $\mathrm{MSE}[Y] = p^2$,
ce qui est encore plus petit et ne coûte rien! On veut faire mieux que cela.

Si on prend $\bar{X}_n$ comme estimateur, alors $\mathrm{Var}[\bar{X}_n] = p(1-p)/n$.
On a $\mathrm{MSE}[\bar{X}_n] < \mathrm{MSE}[Y]$ ssi $p(1-p)/n < p^2$ ssi $n > (1-p)/p$.

Si $|p|$ est petit, il est plus approprié de considérer le MSE relatif $\mathrm{MSE}[X]/p^2$, ou l'erreur relative $\mathrm{RE}[X]$, car la largeur relative d'un intervalle de confiance sur p est à peu près proportionnelle à $\mathrm{RE}[\bar{X}_n] = \mathrm{RE}[X]/\sqrt{n}$. Ici, $\mathrm{RE}[X] = \sqrt{(1-p)/p} \to \infty$ lorsque $p \to 0$.

Par ex., si $p \approx 10^{-10}$, il faut $n \approx 10^{12}$ pour avoir $\mathrm{RE}[\bar{X}_n] \approx 10\%$.

Autre point de vue: Si p est très petit (np est petit), on a de fortes chances d'avoir $X_1 = \cdots = X_n = 0$, ce qui donne $\bar{X}_n = S_n^2 = 0$ et un intervalle de confiance de largeur 0.

$$\mathbb{P}[\bar{X}_n = S_n^2 = 0] = (\mathbb{P}[X_i = 0])^n = (1-p)^n \approx 1 - np.$$

Autre point de vue: Si p est très petit (np est petit), on a de fortes chances d'avoir $X_1 = \cdots = X_n = 0$, ce qui donne $\bar{X}_n = S_n^2 = 0$ et un intervalle de confiance de largeur 0.

$$\mathbb{P}[\bar{X}_n = S_n^2 = 0] = (\mathbb{P}[X_i = 0])^n = (1-p)^n \approx 1 - np.$$

Ici, $Y = n\bar{X}_n$ suit la loi binomiale de paramètres (n, p) et l'approximation normale est bonne seulement si n et np sont grands.

Autre point de vue: Si p est très petit (np est petit), on a de fortes chances d'avoir $X_1 = \cdots = X_n = 0$, ce qui donne $\bar{X}_n = S_n^2 = 0$ et un intervalle de confiance de largeur 0.

$$\mathbb{P}[\bar{X}_n = S_n^2 = 0] = (\mathbb{P}[X_i = 0])^n = (1-p)^n \approx 1 - np.$$

Ici, $Y = n\bar{X}_n$ suit la loi binomiale de paramètres (n, p) et l'approximation normale est bonne seulement si n et np sont grands.
Si n est grand et np est petit, alors Y approx. Poisson(np).

Autre point de vue: Si p est très petit (np est petit), on a de fortes chances d'avoir $X_1 = \cdots = X_n = 0$, ce qui donne $\bar{X}_n = S_n^2 = 0$ et un intervalle de confiance de largeur 0.

$$\mathbb{P}[\bar{X}_n = S_n^2 = 0] = (\mathbb{P}[X_i = 0])^n = (1-p)^n \approx 1 - np.$$

Ici, $Y = n\bar{X}_n$ suit la loi binomiale de paramètres (n, p) et l'approximation normale est bonne seulement si n et np sont grands.
Si n est grand et np est petit, alors Y approx. Poisson(np).

Il existe des techniques de simulation spéciales pour ce contexte d'événements rares.

Choix de la loi d'échantillonnage: "importance sampling"

Supposons que Y a la densité π sur $\mathbb{R}$ et qu'on veut estimer

$$\mu = \mathbb{E}_\pi[h(Y)] = \int_{-\infty}^{\infty} h(y)\pi(y)\mathrm{d}y < \infty,$$

où $h : \mathbb{R} \to [0, \infty)$. Monte Carlo: générer $Y \sim \pi$, retourner $h(Y)$.

Soit g une autre densité sur $\mathbb{R}$, avec $g(y) > 0$ quand $h(y)\pi(y) > 0$. On a

$$\mu = \int_{-\infty}^{\infty} \left[\frac{h(y)\pi(y)}{g(y)}\right] g(y)\mathrm{d}y = \mathbb{E}_g\left[\frac{h(Y)\pi(Y)}{g(Y)}\right] = \mathbb{E}_g\left[h(Y)L\right]$$

où $L = L(Y) = \pi(Y)/g(Y)$ est un rapport de vraisemblance.

Choix de la loi d'échantillonnage: "importance sampling"

Supposons que Y a la densité π sur $\mathbb{R}$ et qu'on veut estimer

$$\mu = \mathbb{E}_\pi[h(Y)] = \int_{-\infty}^{\infty} h(y)\pi(y)\mathrm{d}y < \infty,$$

où $h : \mathbb{R} \to [0, \infty)$. Monte Carlo: générer $Y \sim \pi$, retourner $h(Y)$.

Soit g une autre densité sur $\mathbb{R}$, avec $g(y) > 0$ quand $h(y)\pi(y) > 0$. On a

$$\mu = \int_{-\infty}^{\infty} \left[\frac{h(y)\pi(y)}{g(y)}\right] g(y)\mathrm{d}y = \mathbb{E}_g\left[\frac{h(Y)\pi(Y)}{g(Y)}\right] = \mathbb{E}_g\left[h(Y)L\right]$$

où $L = L(Y) = \pi(Y)/g(Y)$ est un rapport de vraisemblance. Pour estimer μ (sans biais), on peut générer $Y_1, \ldots, Y_n$ i.i.d. selon g, et prendre la moyenne:

$$\hat{\mu}_n = \frac{1}{n}\sum_{i=1}^{n} h(Y_i)\frac{\pi(Y_i)}{g(Y_i)} = \frac{1}{n}\sum_{i=1}^{n} h(Y_i)L_i.$$

Par exemple, g pourrait être une densité normale même si π ne l'est pas.

On a $\mathrm{Var}[\hat{\mu}_n] = \mathrm{Var}_g[h(Y)L]/n$, où

$$\begin{aligned}\mathrm{Var}_g[h(Y)L] &= \mathbb{E}_g\left[h^2(Y)L^2\right] - \mu^2 \\ &= \int_{-\infty}^{\infty} \frac{h^2(y)\pi^2(y)}{g^2(y)} g(y)\mathrm{d}y - \mu^2 \;=\; \int_{-\infty}^{\infty} \frac{h^2(y)\pi^2(y)}{g(y)}\mathrm{d}y - \mu^2.\end{aligned}$$

Si $g(y) \ll h(y)\pi(y)$ quelque part, ou si $h^2(y)\pi(y)/g(y)$ converge vers 0 trop lentement lorsque $y \to \pm\infty$, alors la variance peut être énorme, voire infinie. Le choix de g est crucial.

On a $\mathrm{Var}[\hat{\mu}_n] = \mathrm{Var}_g[h(Y)L]/n$, où

$$\begin{aligned}\mathrm{Var}_g[h(Y)L] &= \mathbb{E}_g\left[h^2(Y)L^2\right] - \mu^2 \\ &= \int_{-\infty}^{\infty} \frac{h^2(y)\pi^2(y)}{g^2(y)} g(y)\mathrm{d}y - \mu^2 \;=\; \int_{-\infty}^{\infty} \frac{h^2(y)\pi^2(y)}{g(y)}\mathrm{d}y - \mu^2.\end{aligned}$$

Si $g(y) \ll h(y)\pi(y)$ quelque part, ou si $h^2(y)\pi(y)/g(y)$ converge vers 0 trop lentement lorsque $y \to \pm\infty$, alors la variance peut être énorme, voire infinie. Le choix de g est crucial.

Si $L \leq 1$ quand $h(Y) > 0$, alors

$$\mathrm{Var}_g[h(Y)L] = \mathbb{E}_g[h(Y)^2L^2] - \mu^2 = \mathbb{E}_\pi[h(Y)^2\, L] - \mu^2 \leq \mathbb{E}_\pi[h(Y)^2] - \mu^2 = \mathrm{Var}[X].$$

On a $\mathrm{Var}[\hat{\mu}_n] = \mathrm{Var}_g[h(Y)L]/n$, où

$$\begin{aligned}\mathrm{Var}_g[h(Y)L] &= \mathbb{E}_g\left[h^2(Y)L^2\right] - \mu^2 \\ &= \int_{-\infty}^{\infty} \frac{h^2(y)\pi^2(y)}{g^2(y)} g(y)\mathrm{d}y - \mu^2 \;=\; \int_{-\infty}^{\infty} \frac{h^2(y)\pi^2(y)}{g(y)}\mathrm{d}y - \mu^2.\end{aligned}$$

Si $g(y) \ll h(y)\pi(y)$ quelque part, ou si $h^2(y)\pi(y)/g(y)$ converge vers 0 trop lentement lorsque $y \to \pm\infty$, alors la variance peut être énorme, voire infinie. Le choix de g est crucial.

Si $L \leq 1$ quand $h(Y) > 0$, alors

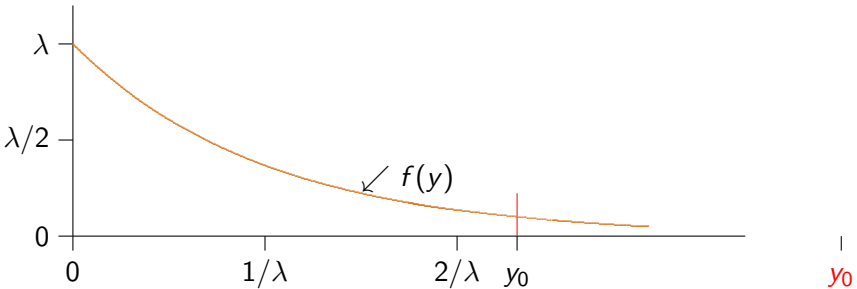
$$\mathrm{Var}_g[h(Y)L] = \mathbb{E}_g[h(Y)^2L^2] - \mu^2 = \mathbb{E}_\pi[h(Y)^2\, L] - \mu^2 \leq \mathbb{E}_\pi[h(Y)^2] - \mu^2 = \mathrm{Var}[X].$$

Quel est le g optimal? Si on prend $g(y)$ proportionnel à $h(y)\pi(y)$, alors l'estimateur IS devient une constante, il ne dépend plus de y, donc sa variance est zéro! Comme l'estimateur est sans biais, cette constante est nécessairement $\mu = \int_{-\infty}^{\infty} h(y)\pi(y)\mathrm{d}y$.

Implanter ceci exactement est rarement possible, mais on peut souvent l'approximer.

Mini-Exemple: estimer une probabilité

On veut estimer $p = \mathbb{P}[Y > y_0] = \mathbb{E}[\mathbb{I}[Y > y_0]]$, où Y est exponentielle de paramètre (taux) λ, i.e., $\pi(y) = \lambda e^{-\lambda y}$ pour $y \geq 0$.
(Exemple purement académique; on sait que $\mathbb{P}[Y > y] = e^{-\lambda y}$.)



Monte Carlo: générer Y selon la bonne densité exponentielle π, soit $Y = -\ln(1-U)/\lambda$, et calculer $X = \mathbb{I}[Y > y_0]$. Répéter n fois et calculer la moyenne des n réalisations de X.

Mini-Exemple: estimer une probabilité

On veut estimer $p = \mathbb{P}[Y > y_0] = \mathbb{E}[\mathbb{I}[Y > y_0]]$, où Y est exponentielle de paramètre (taux) λ, i.e., $\pi(y) = \lambda e^{-\lambda y}$ pour $y \geq 0$.

Soit $g = \pi_0$ une autre densité exponentielle, de paramètre $\lambda_0 \neq \lambda$. On a

$$p = \int_0^\infty \mathbb{I}[y \geq y_0]\pi(y)\mathrm{d}y = \int_0^\infty \mathbb{I}[y \geq y_0]\frac{\pi(y)}{\pi_0(y)}\pi_0(y)\mathrm{d}y = \mathbb{E}_{\pi_0}[X_{\mathrm{is}}],$$

où

$$X_{\mathrm{is}} = \mathbb{I}[Y_0 \geq y_0]\frac{\pi(Y_0)}{\pi_0(Y_0)} = \mathbb{I}[Y_0 \geq y_0]\frac{\lambda \exp[-\lambda Y_0]}{\lambda_0 \exp[-\lambda_0 Y_0]} = \mathbb{I}[Y_0 \geq y_0]\frac{\lambda}{\lambda_0}\exp[-(\lambda - \lambda_0)Y_0]$$

et $Y_0 = -\ln(1-U)/\lambda_0$ est une v.a. exponentielle de taux λ_0.

Cela donne un estimateur sans biais peu importe $\lambda_0 > 0$.

On a aussi

$$\begin{aligned}
\mathrm{Var}[X_{\mathrm{is}}] &= \mathbb{E}[X_{\mathrm{is}}^2] - p^2 \\
&= \int_{y_0}^{\infty} \frac{\pi^2(y)}{\pi_0^2(y)} \pi_0(y) \mathrm{d}y \; - p^2 \\
&= \int_{y_0}^{\infty} (\lambda/\lambda_0)^2 \exp[-2(\lambda - \lambda_0)y] \lambda_0 \exp[-\lambda_0 y] \mathrm{d}y \; - p^2 \\
&= \begin{cases} \dfrac{\lambda^2}{\lambda_0(2\lambda - \lambda_0)} \exp[-(2\lambda - \lambda_0)y_0] - p^2 & \text{si } 0 < \lambda_0 < 2\lambda, \\ \infty & \text{sinon.} \end{cases}
\end{aligned}$$

Pour $\lambda_0 = \lambda$, la variance est $p(1-p)$. Lorsque λ_0 s'approche de 0 ou de 2λ, la variance tend vers l'infini.

On a aussi

$$\begin{aligned}
\mathrm{Var}[X_{\mathrm{is}}] &= \mathbb{E}[X_{\mathrm{is}}^2] - p^2 \\
&= \int_{y_0}^{\infty} \frac{\pi^2(y)}{\pi_0^2(y)} \pi_0(y) \mathrm{d}y \; - p^2 \\
&= \int_{y_0}^{\infty} (\lambda/\lambda_0)^2 \exp[-2(\lambda - \lambda_0)y] \lambda_0 \exp[-\lambda_0 y] \mathrm{d}y \; - p^2 \\
&= \begin{cases} \dfrac{\lambda^2}{\lambda_0(2\lambda - \lambda_0)} \exp[-(2\lambda - \lambda_0)y_0] - p^2 & \text{si } 0 < \lambda_0 < 2\lambda, \\ \infty & \text{sinon.} \end{cases}
\end{aligned}$$

Pour $\lambda_0 = \lambda$, la variance est $p(1-p)$. Lorsque λ_0 s'approche de 0 ou de 2λ, la variance tend vers l'infini. Exercice: La variance est minimisée pour $\lambda_0 = \lambda + 1/y_0 - (\lambda^2 + 1/y_0^2)^{1/2} < \lambda$.

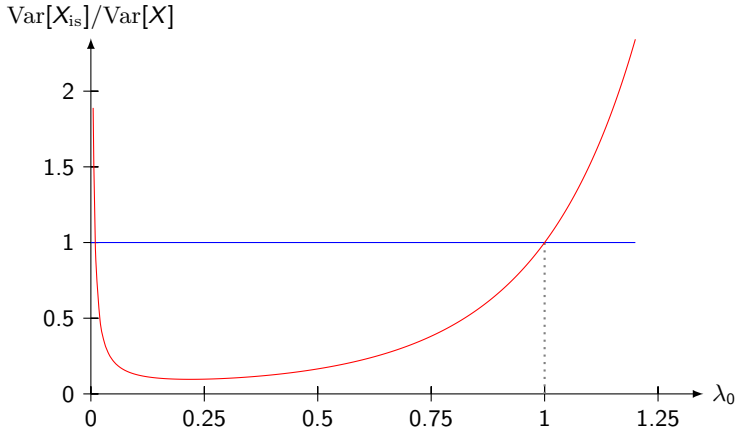
On a aussi

$$\begin{aligned}
\mathrm{Var}[X_{\mathrm{is}}] &= \mathbb{E}[X_{\mathrm{is}}^2] - p^2 \\
&= \int_{y_0}^{\infty} \frac{\pi^2(y)}{\pi_0^2(y)} \pi_0(y) \mathrm{d}y \; - p^2 \\
&= \int_{y_0}^{\infty} (\lambda/\lambda_0)^2 \exp[-2(\lambda - \lambda_0)y] \lambda_0 \exp[-\lambda_0 y] \mathrm{d}y \; - p^2 \\
&= \begin{cases} \dfrac{\lambda^2}{\lambda_0(2\lambda - \lambda_0)} \exp[-(2\lambda - \lambda_0)y_0] - p^2 & \text{si } 0 < \lambda_0 < 2\lambda, \\ \infty & \text{sinon.} \end{cases}
\end{aligned}$$

Pour $\lambda_0 = \lambda$, la variance est $p(1-p)$. Lorsque λ_0 s'approche de 0 ou de 2λ, la variance tend vers l'infini. Exercice: La variance est minimisée pour $\lambda_0 = \lambda + 1/y_0 - (\lambda^2 + 1/y_0^2)^{1/2} < \lambda$.

Cet exemple montre que remplacer π par une autre densité g peut réduire la variance, mais peut aussi l'augmenter, et même la rendre infinie.
La variance est souvent très sensible au choix de g.

Exemple numérique: $\lambda = 1$ et $y_0 = 4$.



Variance min. pour $\lambda_0 \approx 0.2192$; donne $\mathrm{Var}[X_{\mathrm{is}}]/\mathrm{Var}[X] \approx 0.0962$.

Ainsi, pour estimer $p = \mathbb{P}[Y > 4]$, on change la moyenne de Y pour $1/\lambda_0 \approx 1/.2192 \approx 4.56$, qui est proche de 4. Semble raisonnable.

Densité g optimale pour cet exemple: $g(y) \propto \mathbb{I}[y > y_0]\pi(y)$.
C'est une exponentielle tronquée à $[y_0, \infty)$:

$$g(y) = \frac{\pi(y)}{\mathbb{P}[Y > y_0]} = \frac{\lambda e^{-\lambda y}}{e^{-\lambda y_0}} = \lambda e^{-\lambda(y-y_0)} \qquad \text{pour } y > y_0, \text{ et 0 ailleurs.}$$

C'est la densité de $y_0 + X$ où X est une v.a. exponentielle de taux λ.
On peut donc générer $Y = y_0 - \ln(1-U)/\lambda$.

Densité g optimale pour cet exemple: $g(y) \propto \mathbb{I}[y > y_0]\pi(y)$.
C'est une exponentielle tronquée à $[y_0, \infty)$:

$$g(y) = \frac{\pi(y)}{\mathbb{P}[Y > y_0]} = \frac{\lambda e^{-\lambda y}}{e^{-\lambda y_0}} = \lambda e^{-\lambda(y-y_0)} \qquad \text{pour } y > y_0, \text{ et 0 ailleurs.}$$

C'est la densité de $y_0 + X$ où X est une v.a. exponentielle de taux λ.
On peut donc générer $Y = y_0 - \ln(1-U)/\lambda$.

Le rapport de vraisemblance est $L = \pi(Y)/g(Y) = e^{-\lambda y_0}$ et on a

$$X_{\text{is}} = \mathbb{I}[Y \geq y_0] \exp[-\lambda y_0] = \exp[-\lambda y_0] = p,$$

car on a toujours $Y \geq y_0$.
Ce X_{is} est donc un estimateur sans biais de variance zero!

Densité g optimale pour cet exemple: $g(y) \propto \mathbb{I}[y > y_0]\pi(y)$.
C'est une exponentielle tronquée à $[y_0, \infty)$:

$$g(y) = \frac{\pi(y)}{\mathbb{P}[Y > y_0]} = \frac{\lambda e^{-\lambda y}}{e^{-\lambda y_0}} = \lambda e^{-\lambda(y-y_0)} \qquad \text{pour } y > y_0, \text{ et 0 ailleurs.}$$

C'est la densité de $y_0 + X$ où X est une v.a. exponentielle de taux λ.
On peut donc générer $Y = y_0 - \ln(1-U)/\lambda$.

Le rapport de vraisemblance est $L = \pi(Y)/g(Y) = e^{-\lambda y_0}$ et on a

$$X_{\text{is}} = \mathbb{I}[Y \geq y_0] \exp[-\lambda y_0] = \exp[-\lambda y_0] = p,$$

car on a toujours $Y \geq y_0$.
Ce X_{is} est donc un estimateur sans biais de variance zero!

En théorie, de tels estimateurs "magiques" existent toujours. En pratique, il sont très difficiles à trouver et implanter. Par contre, on peut souvent les approximer et construire ainsi des estimateurs plus performants.

Exemple. Soient Y_1 et Y_2 des v.a. indép. de densités π_1 et π_2, sur $\mathbb{R}$.

$$X = \begin{cases} Y_1 + Y_2 - K & \text{si } Y_1 \leq a \text{ et } Y_1 + Y_2 \geq b, \\ 0 & \text{sinon,} \end{cases}$$

où $K > 0$, et a et b sont des constantes. On veut estimer $\mu = \mathbb{E}[X]$.

Exemple. Soient Y_1 et Y_2 des v.a. indép. de densités π_1 et π_2, sur $\mathbb{R}$.

$$X = \begin{cases} Y_1 + Y_2 - K & \text{si } Y_1 \leq a \text{ et } Y_1 + Y_2 \geq b, \\ 0 & \text{sinon,} \end{cases}$$

où $K > 0$, et a et b sont des constantes. On veut estimer $\mu = \mathbb{E}[X]$.

MC standard: générer Y_1 et Y_2 selon π_1 et π_2, et calculer X.

Exemple. Soient Y_1 et Y_2 des v.a. indép. de densités π_1 et π_2, sur $\mathbb{R}$.

$$X = \begin{cases} Y_1 + Y_2 - K & \text{si } Y_1 \le a \text{ et } Y_1 + Y_2 \ge b, \\ 0 & \text{sinon,} \end{cases}$$

où $K > 0$, et a et b sont des constantes. On veut estimer $\mu = \mathbb{E}[X]$.

MC standard: générer Y_1 et Y_2 selon π_1 et π_2, et calculer X.

Stratégie IS: Éviter de gaspiller des échantillons dans la région où $X = 0$.
Par exemple, générer Y_1 selon sa densité conditionnelle à $Y_1 \le a$,
puis générer Y_2 selon sa densité conditionnelle à $Y_1 + Y_2 \ge b$,
i.e., tronquée à l'intervalle $[b - Y_1, \infty)$.

Exemple. Soient Y_1 et Y_2 des v.a. indép. de densités π_1 et π_2, sur $\mathbb{R}$.

$$X = \begin{cases} Y_1 + Y_2 - K & \text{si } Y_1 \le a \text{ et } Y_1 + Y_2 \ge b, \\ 0 & \text{sinon,} \end{cases}$$

où $K > 0$, et a et b sont des constantes. On veut estimer $\mu = \mathbb{E}[X]$.

MC standard: générer Y_1 et Y_2 selon π_1 et π_2, et calculer X.

Stratégie IS: Éviter de gaspiller des échantillons dans la région où $X = 0$.
Par exemple, générer Y_1 selon sa densité conditionnelle à $Y_1 \le a$,
puis générer Y_2 selon sa densité conditionnelle à $Y_1 + Y_2 \ge b$,
i.e., tronquée à l'intervalle $[b - Y_1, \infty)$.

La nouvelle densité de Y_1 est

$$g_1(y) = \frac{\pi_1(y)}{\mathbb{P}[Y_1 \le a]} = \frac{\pi_1(y)}{F_1(a)} \qquad \text{pour } y \le a,$$

et celle de Y_2 conditionelle à $Y_1 = y_1$ est

$$g_2(y \mid y_1) = \frac{\pi_2(y)}{\mathbb{P}[Y_2 \geq b - y_1]} = \frac{\pi_2(y)}{1 - F_2(b - y_1)} \qquad \text{pour } y \geq b - y_1,$$

où F_1 et F_2 sont les fonctions de répartition de Y_1 et Y_2.

$$g_2(y \mid y_1) = \frac{\pi_2(y)}{\mathbb{P}[Y_2 \geq b - y_1]} = \frac{\pi_2(y)}{1 - F_2(b - y_1)} \qquad \text{pour } y \geq b - y_1,$$

où F_1 et F_2 sont les fonctions de répartition de Y_1 et Y_2. On a

$$\begin{aligned}
\mu &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} X\, \pi_2(y_2)\pi_1(y_1) \mathrm{d}y_2 \mathrm{d}y_1 \\
&= \int_{-\infty}^{a} \int_{b-y_1}^{\infty} X \frac{\pi_2(y_2)\pi_1(y_1)}{g_2(y_2 \mid y_1) g_1(y_1)} g_2(y_2 \mid y_1) g_1(y_1) \mathrm{d}y_2 \mathrm{d}y_1 \\
&= \int_{-\infty}^{a} \int_{b-y_1}^{\infty} X\, F_1(a)\,(1 - F_2(b - y_1)) g_2(y_2 \mid y_1) g_1(y_1) \mathrm{d}y_2 \mathrm{d}y_1 \\
&= \mathbb{E}_g[X_{\mathrm{is}}]
\end{aligned}$$

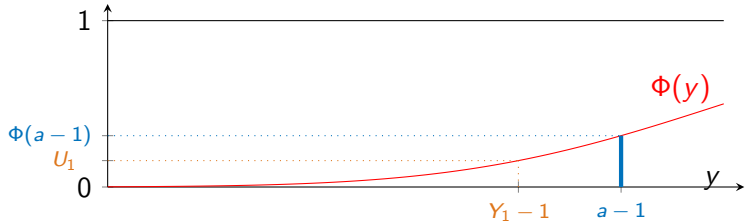
où $X_{\mathrm{is}} = X\, L$ avec $L = F_1(a)\,(1 - F_2(b - Y_1))$, et $\mathbb{E}_g$ désigne l'espérance sous g_1 et g_2.

On a toujours $\mathrm{Var}[X_{\mathrm{is}}] = \mathbb{E}_g[X^2 L^2] - \mu^2 = \mathbb{E}_\pi[X^2\, L] - \mu^2 < \mathrm{Var}[X]$ car $L < 1$.

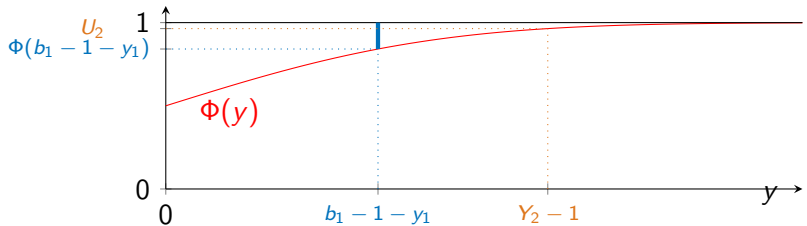
Expérience: Supposons π_1 et $\pi_2 \sim \mathcal{N}(1,1)$, $K = 1$, $b = 2$, et $a = 1/2$.
Ici, $Y_1 - 1$ et $Y_2 - 1$ sont $\mathcal{N}(0,1)$, de cdf Φ.
Essayer MC vs IS, avec $n = 10^5$, et comparer les variances.

Expérience: Supposons π_1 et $\pi_2 \sim \mathcal{N}(1,1)$, $K = 1$, $b = 2$, et $a = 1/2$.
Ici, $Y_1 - 1$ et $Y_2 - 1$ sont $\mathcal{N}(0,1)$, de cdf Φ.
Essayer MC vs IS, avec $n = 10^5$, et comparer les variances.

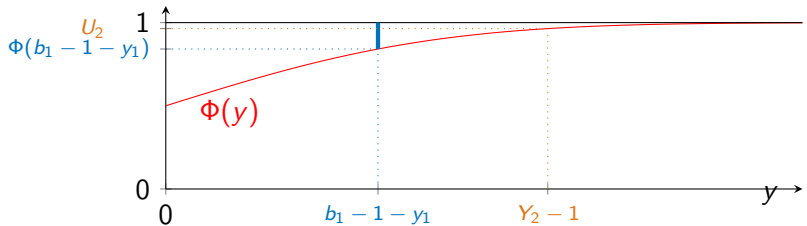
On a $F_1(a) = \mathbb{P}[Y_1 < a] = \mathbb{P}[Y_1 - 1 < a - 1] = \Phi(a-1)$.
On pose $U_1 \sim \mathcal{U}(0, \Phi(a-1))$ et $Y_1 = 1 + \Phi^{-1}(U_1)$.



Ensuite, pour $Y_1 = y_1$ donné, on veut $Y_2 > b - y_1$. On a
$1 - F_2(b - y_1) = \mathbb{P}[Y_2 > b - y_1] = \mathbb{P}[Y_2 - 1 > b - 1 - y_1] = 1 - \Phi(b - 1 - y_1)$.
On pose $U_2 \sim \mathcal{U}(\Phi(b - 1 - y_1), 1))$ et $Y_2 = 1 + \Phi^{-1}(U_2)$.

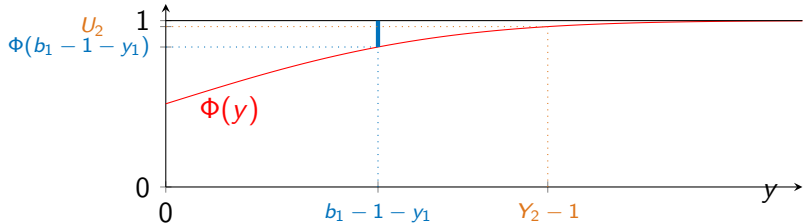


Ensuite, pour $Y_1 = y_1$ donné, on veut $Y_2 > b - y_1$. On a
$1 - F_2(b - y_1) = \mathbb{P}[Y_2 > b - y_1] = \mathbb{P}[Y_2 - 1 > b - 1 - y_1] = 1 - \Phi(b - 1 - y_1)$.
On pose $U_2 \sim \mathcal{U}(\Phi(b - 1 - y_1), 1))$ et $Y_2 = 1 + \Phi^{-1}(U_2)$.



On calcule l'estimateur $X_{\text{is}} = X\Phi(a - 1)(1 - \Phi(b - 1 - Y_1))$.

Ensuite, pour $Y_1 = y_1$ donné, on veut $Y_2 > b - y_1$. On a
$1 - F_2(b - y_1) = \mathbb{P}[Y_2 > b - y_1] = \mathbb{P}[Y_2 - 1 > b - 1 - y_1] = 1 - \Phi(b - 1 - y_1)$.
On pose $U_2 \sim \mathcal{U}(\Phi(b - 1 - y_1), 1))$ et $Y_2 = 1 + \Phi^{-1}(U_2)$.



On calcule l'estimateur $X_{\text{is}} = X\Phi(a - 1)(1 - \Phi(b - 1 - Y_1))$.

La variance S_n^2 est environ 40 fois plus petite avec X_{is} qu'avec X.

Estimator	$\hat{\mu}_n$	S_n^2	IC à 95%
X	0.0733	0.1188	(0.071, 0.075)
X_{is}	0.0742	0.0027	(0.074, 0.075)

Exemple: Une chaine de Markov

$\{X_n,\ n \geq 0\}$ sur les états $\{0, 1, \ldots, K\}$, avec $X_0 = x_0$.
Probabilités de transition $P_{i,j} = \mathbb{P}[X_n = j \mid X_{n-1} = i]$.
On veut estimer la probabilité $\mu = \mu(x_0)$ d'atteindre K avant de revenir à 0.

Exemple: Une chaine de Markov

$\{X_n,\ n \geq 0\}$ sur les états $\{0, 1, \ldots, K\}$, avec $X_0 = x_0$.
Probabilités de transition $P_{i,j} = \mathbb{P}[X_n = j \mid X_{n-1} = i]$.
On veut estimer la probabilité $\mu = \mu(x_0)$ d'atteindre K avant de revenir à 0.

Estimateur naif: $\mathbb{I}[X_T = K]$ où $T = \inf\{n \geq 1 : X_n \in \{0, K\}\}$.

Exemple: Une chaine de Markov

$\{X_n,\ n \geq 0\}$ sur les états $\{0, 1, \ldots, K\}$, avec $X_0 = x_0$.
Probabilités de transition $P_{i,j} = \mathbb{P}[X_n = j \mid X_{n-1} = i]$.
On veut estimer la probabilité $\mu = \mu(x_0)$ d'atteindre K avant de revenir à 0.

Estimateur naif: $\mathbb{I}[X_T = K]$ où $T = \inf\{n \geq 1 : X_n \in \{0, K\}\}$.

IS: changer les $P_{i,j}$ pour des $Q_{i,j}$ pour augmenter les chances d'aller à K.

Exemple: Une chaine de Markov

$\{X_n,\ n \geq 0\}$ sur les états $\{0, 1, \ldots, K\}$, avec $X_0 = x_0$.
Probabilités de transition $P_{i,j} = \mathbb{P}[X_n = j \mid X_{n-1} = i]$.
On veut estimer la probabilité $\mu = \mu(x_0)$ d'atteindre K avant de revenir à 0.

Estimateur naif: $\mathbb{I}[X_T = K]$ où $T = \inf\{n \geq 1 : X_n \in \{0, K\}\}$.

IS: changer les $P_{i,j}$ pour des $Q_{i,j}$ pour augmenter les chances d'aller à K.

La probabilité d'une trajectoire $X_1, X_2, \ldots, X_T$ est $\prod_{n=1}^{T} P_{X_{n-1},X_n}$ avec les probabilités originales et $\prod_{n=1}^{T} Q_{X_{n-1},X_n}$ avec les nouvelles probabilités. En faisant la somme sur toutes les valeurs de T et toutes les trajectoires possibles pour chaque T, on obtient

$$\mu = \sum_{T,X_1,\ldots,X_T} \mathbb{I}[X_T = K] \prod_{n=1}^{T} P_{X_{n-1},X_n} = \sum_{T,X_1,\ldots,X_T} \mathbb{I}[X_T = K] \prod_{n=1}^{T} \frac{P_{X_{n-1},X_n}}{Q_{X_{n-1},X_n}} \prod_{n=1}^{T} Q_{X_{n-1},X_n}$$

$$= \sum_{T,X_1,\ldots,X_T} \mathbb{I}[X_T = K] L \prod_{n=1}^{T} Q_{X_{n-1},X_n} = \mathbb{E}_Q[\mathbb{I}[X_T = K]L] = \mathbb{E}_Q[X_{\text{is}}].$$

De plus, si on a toujours $L \leq 1$ lorsque $X_T = K$, alors $\mathrm{Var}_Q[\mathbb{I}[X_T = K]L] \leq \mathrm{Var}_P[\mathbb{I}[X_T = K]$.

Comment choisir les $Q_{i,j}$?

Idée simpliste: bloquer les retours à 0 en posant $Q_{i,0} = 0$ pour tout $i > 0$, et renormaliser les autres probabilités: $Q_{i,j} = P_{i,j}/(1 - P_{i,0})$ pour $i, j > 0$, et $Q_{0,j} = P_{0,j}$ pour tout j.

Comment choisir les $Q_{i,j}$?

Idée simpliste: bloquer les retours à 0 en posant $Q_{i,0} = 0$ pour tout $i > 0$, et renormaliser les autres probabilités: $Q_{i,j} = P_{i,j}/(1 - P_{i,0})$ pour $i, j > 0$, et $Q_{0,j} = P_{0,j}$ pour tout j.
On a alors $\mathbb{P}[X_T = K] = 1$ et le nouvel estimateur est

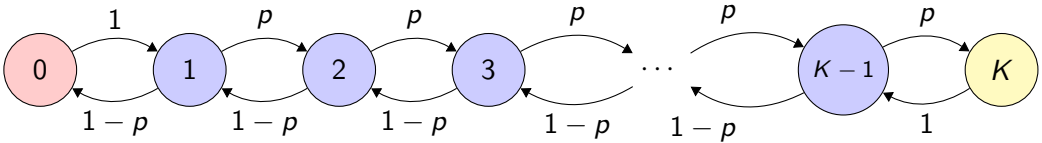
$$L = L(X_1, \ldots, X_T) = \prod_{n=1}^{T} \frac{P_{X_{n-1},X_n}}{Q_{X_{n-1},X_n}} = \prod_{n=2}^{T} (1 - P_{X_{n-1},0}) < 1.$$

La variance est toujours réduite car $L < 1$ quand $X_T = K$.

Mais: chaque simulation risque d'être très longue (on peut de se promener autour de l'état 0 très longtemps). L'efficacité n'est pas nécessairement améliorée. On doit faire mieux.

Une marche aléatoire unidimensionnelle sur $\{0, 1, \ldots, K\}$, avec $x_0 = 1$.

Soient $P_{i,i+1} = p$ et $P_{i,i-1} = 1 - p$ pour $1 \leq i \leq K - 1$, et $P_{0,1} = P_{K,K-1} = 1$.

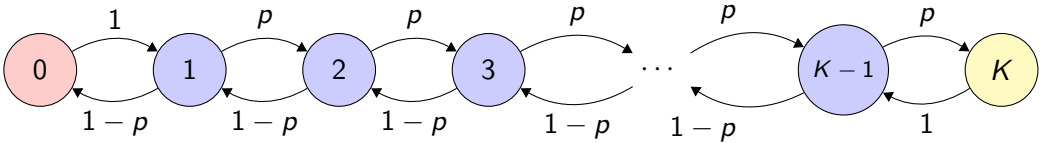


Si $p < 1/2$ et K est grand, la chaine est attirée vers 0.

Couper l'accès à 0 ne suffit pas, il faut augmenter l'attirance vers K.

Une marche aléatoire unidimensionnelle sur $\{0, 1, \ldots, K\}$, avec $x_0 = 1$.

Soient $P_{i,i+1} = p$ et $P_{i,i-1} = 1 - p$ pour $1 \leq i \leq K - 1$, et $P_{0,1} = P_{K,K-1} = 1$.



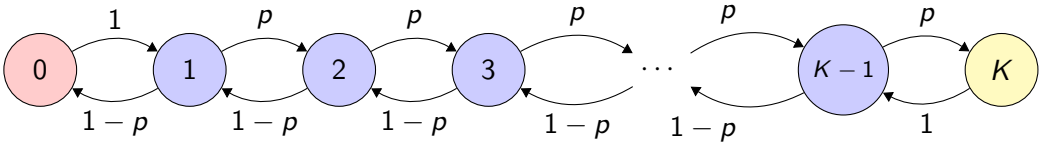
Si $p < 1/2$ et K est grand, la chaine est attirée vers 0.

Couper l'accès à 0 ne suffit pas, il faut augmenter l'attirance vers K.

On va modifier les probabilités $P_{i,j}$ pour $Q_{i,i+1} = q$ et $Q_{i,i-1} = 1 - q$, pour $q > p$.

Une marche aléatoire unidimensionnelle sur $\{0, 1, \ldots, K\}$, avec $x_0 = 1$.

Soient $P_{i,i+1} = p$ et $P_{i,i-1} = 1 - p$ pour $1 \leq i \leq K - 1$, et $P_{0,1} = P_{K,K-1} = 1$.



Si $p < 1/2$ et K est grand, la chaine est attirée vers 0.

Couper l'accès à 0 ne suffit pas, il faut augmenter l'attirance vers K.

On va modifier les probabilités $P_{i,j}$ pour $Q_{i,i+1} = q$ et $Q_{i,i-1} = 1 - q$, pour $q > p$.

Examinons le rapport de vraisemblance lorsque $X_T = K$. Pour chaque paire d'états $(i, i+1)$ on va de i à $i+1$ une fois de plus que de $i+1$ à i. Donc

$$L = \prod_{n=1}^{T} \frac{P_{X_{n-1},X_n}}{Q_{X_{n-1},X_n}} = \left(\frac{p}{q}\right)^{K-1} \left(\frac{p(1-p)}{q(1-q)}\right)^{(T-K)/2}.$$

$$L = \left(\frac{p}{q}\right)^{K-1} \left(\frac{p(1-p)}{q(1-q)}\right)^{(T-K)/2}.$$

Pour s'assurer que $L < 1$, prenons $q > p$ et $q(1-q) \geq p(1-p)$, i.e., $p < q \leq 1-p$.

$$L = \left(\frac{p}{q}\right)^{K-1} \left(\frac{p(1-p)}{q(1-q)}\right)^{(T-K)/2}.$$

Pour s'assurer que $L < 1$, prenons $q > p$ et $q(1-q) \geq p(1-p)$, i.e., $p < q \leq 1-p$.

En maximisant q sous cette contrainte (pour aller à K le plus vite et le plus souvent possible), on obtient $q = 1-p$. Le RV se simplifie alors et devient une constante:

$$L = \left(\frac{p}{q}\right)^{K-1} = \left(\frac{p}{1-p}\right)^{K-1} < 1.$$

Avant, l'estimateur était 1 avec prob. μ, et 0 sinon. Variance de $\mu(1-\mu)$.
Maintenant il est L avec prob. μ/L, et 0 sinon.
Nouvelle variance: $L^2\mu/L - \mu^2 = \mu(L-\mu) < L\mu(1-\mu)$.
Avec IS, la variance est inférieure à L fois l'ancienne variance.

$$L = \left(\frac{p}{q}\right)^{K-1} \left(\frac{p(1-p)}{q(1-q)}\right)^{(T-K)/2}.$$

Pour s'assurer que $L < 1$, prenons $q > p$ et $q(1-q) \geq p(1-p)$, i.e., $p < q \leq 1-p$.

En maximisant q sous cette contrainte (pour aller à K le plus vite et le plus souvent possible), on obtient $q = 1-p$. Le RV se simplifie alors et devient une constante:

$$L = \left(\frac{p}{q}\right)^{K-1} = \left(\frac{p}{1-p}\right)^{K-1} < 1.$$

Avant, l'estimateur était 1 avec prob. μ, et 0 sinon. Variance de $\mu(1-\mu)$.
Maintenant il est L avec prob. μ/L, et 0 sinon.
Nouvelle variance: $L^2\mu/L - \mu^2 = \mu(L-\mu) < L\mu(1-\mu)$.
Avec IS, la variance est inférieure à L fois l'ancienne variance.

Example: si $p = 1/3$ et $K = 101$, on a $L = 2^{-100} \approx 0.833333 \times 10^{-30}$.
Dans ce cas, la variance est divisée par presque 10^{-30} !

Monte Carlo Conditionnel (conditional Monte Carlo (CMC))

D'abord un exemple très simple.

Supposons que l'on veut estimer $\mu = \mathbb{P}[Y_1 + \cdots + Y_t > x]$ où $Y_1, \ldots, Y_t$ sont indépendants.

Estimateur MC évident: $X = \mathbb{I}[Y_1 + \cdots + Y_t > x]$.

Monte Carlo Conditionnel (conditional Monte Carlo (CMC))

D'abord un exemple très simple.

Supposons que l'on veut estimer $\mu = \mathbb{P}[Y_1 + \cdots + Y_t > x]$ où $Y_1, \ldots, Y_t$ sont indépendants.

Estimateur MC évident: $X = \mathbb{I}[Y_1 + \cdots + Y_t > x]$.

Estimateur CMC: on ne génère pas Y_t. On estime μ par $X_{e,t-1} = \mathbb{P}[Y_1 + \cdots + Y_t > x \mid Y_1, \ldots, Y_{t-1}] = \mathbb{P}[Y_t > x - Y_1 - \cdots - Y_{t-1} \mid Y_1, \ldots, Y_{t-1}] = 1 - F_t[x - Y_1 - \cdots - Y_{t-1}]$.

Monte Carlo Conditionnel (conditional Monte Carlo (CMC))

D'abord un exemple très simple.

Supposons que l'on veut estimer $\mu = \mathbb{P}[Y_1 + \cdots + Y_t > x]$ où $Y_1, \ldots, Y_t$ sont indépendants.

Estimateur MC évident: $X = \mathbb{I}[Y_1 + \cdots + Y_t > x]$.

Estimateur CMC: on ne génère pas Y_t. On estime μ par $X_{\mathrm{e},t-1} = \mathbb{P}[Y_1 + \cdots + Y_t > x \mid Y_1, \ldots, Y_{t-1}] = \mathbb{P}[Y_t > x - Y_1 - \cdots - Y_{t-1} \mid Y_1, \ldots, Y_{t-1}] = 1 - F_t[x - Y_1 - \cdots - Y_{t-1}]$.

Estimateur CMC plus général: $X_{\mathrm{e},s} = \mathbb{P}[Y_1 + \cdots + Y_t > x \mid Y_1, \ldots, Y_s]$, pour $s \leq t$.

Monte Carlo Conditionnel (conditional Monte Carlo (CMC))

D'abord un exemple très simple.

Supposons que l'on veut estimer $\mu = \mathbb{P}[Y_1 + \cdots + Y_t > x]$ où $Y_1, \ldots, Y_t$ sont indépendants.

Estimateur MC évident: $X = \mathbb{I}[Y_1 + \cdots + Y_t > x]$.

Estimateur CMC: on ne génère pas Y_t. On estime μ par $X_{e,t-1} = \mathbb{P}[Y_1 + \cdots + Y_t > x \mid Y_1, \ldots, Y_{t-1}] = \mathbb{P}[Y_t > x - Y_1 - \cdots - Y_{t-1} \mid Y_1, \ldots, Y_{t-1}] = 1 - F_t[x - Y_1 - \cdots - Y_{t-1}]$.

Estimateur CMC plus général: $X_{e,s} = \mathbb{P}[Y_1 + \cdots + Y_t > x \mid Y_1, \ldots, Y_s]$, pour $s \leq t$.

Pour $s = t$: aucun changement.

Pour $s = t - 1$, c'est le cas précédent.

Pour $s = 0$: $X_{e,0} = \mathbb{P}[Y_1 + \cdots + Y_t > x] = \mu$ (variance réduite à zéro).

Plus s est petit, plus la variance est réduite, mais plus l'estimateur est difficile à calculer.

Cadre général

Idée: remplacer l'estimateur X par $\mathbb{E}[X \mid Z]$ où Z est une autre v.a., ou plus généralement par $\mathbb{E}[X \mid \mathcal{G}]$, où $\mathcal{G}$ représente une information partielle sur X.

Cadre général

Idée: remplacer l'estimateur X par $\mathbb{E}[X \mid Z]$ où Z est une autre v.a., ou plus généralement par $\mathbb{E}[X \mid \mathcal{G}]$, où $\mathcal{G}$ représente une information partielle sur X. L'estimateur CMC est

$$X_e \stackrel{\text{def}}{=} \mathbb{E}[X \mid \mathcal{G}].$$

On a

$$\mathbb{E}[X_e] = \mathbb{E}[\mathbb{E}[X \mid \mathcal{G}]] = \mathbb{E}[X].$$

Cadre général

Idée: remplacer l'estimateur X par $\mathbb{E}[X \mid Z]$ où Z est une autre v.a., ou plus généralement par $\mathbb{E}[X \mid \mathcal{G}]$, où $\mathcal{G}$ représente une information partielle sur X. L'estimateur CMC est

$$X_e \stackrel{\text{def}}{=} \mathbb{E}[X \mid \mathcal{G}].$$

On a

$$\mathbb{E}[X_e] = \mathbb{E}[\mathbb{E}[X \mid \mathcal{G}]] = \mathbb{E}[X].$$

De plus,

$$\mathrm{Var}[X] = \mathbb{E}[\underbrace{\mathrm{Var}[X \mid \mathcal{G}]}_{\substack{\text{Var. résiduelle} \\ \text{pour } \mathcal{G} \text{ connu} \\ \text{(éliminé par CMC)}}}] + \underbrace{\mathrm{Var}[\mathbb{E}[X \mid \mathcal{G}]]}_{\substack{\text{Var. due à la} \\ \text{variation de } \mathcal{G}}} = \mathbb{E}[\mathrm{Var}[X \mid \mathcal{G}]] + \mathrm{Var}[X_e],$$

et donc

$$\mathrm{Var}[X_e] \;=\; \mathrm{Var}[X] - \mathbb{E}[\mathrm{Var}[X \mid \mathcal{G}]] \;\leq\; \mathrm{Var}[X].$$

Pour minimiser la variance, on doit maximiser $\mathbb{E}[\mathrm{Var}[X \mid \mathcal{G}]]$, i.e., $\mathcal{G}$ doit contenir le moins d'information possible.

On sait en effet que $\mathcal{G}_1 \subset \mathcal{G}_2$ implique que $\mathbb{E}[\mathrm{Var}[X \mid \mathcal{G}_1]] \geq \mathbb{E}[\mathrm{Var}[X \mid \mathcal{G}_2]]$. Preuve:

$$\mathrm{Var}[X \mid \mathcal{G}_1] = \mathbb{E}[\mathrm{Var}[X \mid \mathcal{G}_2] \mid \mathcal{G}_1] + \mathrm{Var}[\mathbb{E}[X \mid \mathcal{G}_2] \mid \mathcal{G}_1],$$

et il suffit de prendre l'espérance des deux cotés:

$$\mathbb{E}[\mathrm{Var}[X \mid \mathcal{G}_1]] \geq \mathbb{E}[\mathbb{E}[\mathrm{Var}[X \mid \mathcal{G}_2] \mid \mathcal{G}_1]] = \mathbb{E}[\mathrm{Var}[X \mid \mathcal{G}_2]].$$

Pour minimiser la variance, on doit maximiser $\mathbb{E}[\mathrm{Var}[X \mid \mathcal{G}]]$, i.e., $\mathcal{G}$ doit contenir le moins d'information possible.

On sait en effet que $\mathcal{G}_1 \subset \mathcal{G}_2$ implique que $\mathbb{E}[\mathrm{Var}[X \mid \mathcal{G}_1]] \geq \mathbb{E}[\mathrm{Var}[X \mid \mathcal{G}_2]]$. Preuve:

$$\mathrm{Var}[X \mid \mathcal{G}_1] = \mathbb{E}[\mathrm{Var}[X \mid \mathcal{G}_2] \mid \mathcal{G}_1] + \mathrm{Var}[\mathbb{E}[X \mid \mathcal{G}_2] \mid \mathcal{G}_1],$$

et il suffit de prendre l'espérance des deux cotés:

$$\mathbb{E}[\mathrm{Var}[X \mid \mathcal{G}_1]] \geq \mathbb{E}[\mathbb{E}[\mathrm{Var}[X \mid \mathcal{G}_2] \mid \mathcal{G}_1]] = \mathbb{E}[\mathrm{Var}[X \mid \mathcal{G}_2]].$$

Mais moins $\mathcal{G}_1$ contient d'information, plus il est difficile de calculer X_e.
On doit donc faire un compromis.
Dans certains cas, X_e peut être moins coûteux à calculer que X.

Cas limites:
Si $\mathcal{G}$ ne contient aucune information pertinente à X: $X_e = \mathbb{E}[X \mid \mathcal{G}] = \mathbb{E}[X] = \mu$.
Si $\mathcal{G}$ permet de calculer X (i.e., X est $\mathcal{G}$-mesurable): $X_e = X$.

Déplacement d'une poutre en porte-à-faux

Déplacement X pour une charge horizontale Y_2 et charge verticale Y_3:

$$X = h(Y_1, Y_2, Y_3) = \frac{\kappa}{Y_1}\sqrt{\frac{{Y_2}^2}{w^4} + \frac{{Y_3}^2}{t^4}}$$

où Y_1, Y_2, Y_3 sont normales indépendantes, $Y_j \sim \mathcal{N}(\mu_j, \sigma_j^2)$.
Supposons que l'on veut estimer $\mu = \mathbb{P}[X \leq x]$ pour x fixé.

(a) MC ordinaire: générer X et calculer $I = \mathbb{I}[X \leq x]$, qui vaut 0 ou 1.

(b) CMC: générer seulement Y_2, Y_3, et calculer $J = \mathbb{P}[X \leq x \mid Y_2, Y_3]$. On a

$$X \leq x \quad \text{ssi} \quad Y_1 \geq \frac{\kappa}{x}\sqrt{\frac{Y_2^2}{w^4} + \frac{Y_3^2}{t^4}} \stackrel{\text{def}}{=} W_1(x).$$

Donc pour $x > 0$,

$$J = \mathbb{P}[X \leq x \mid Y_2, Y_3] = \mathbb{P}[Y_1 \geq W_1(x) \mid W_1(x)] = 1 - \Phi((W_1(x) - \mu_1)/\sigma_1).$$

Déplacement d'une poutre en porte-à-faux

Déplacement X pour une charge horizontale Y_2 et charge verticale Y_3:

$$X = h(Y_1, Y_2, Y_3) = \frac{\kappa}{Y_1}\sqrt{\frac{{Y_2}^2}{w^4} + \frac{{Y_3}^2}{t^4}}$$

où Y_1, Y_2, Y_3 sont normales indépendantes, $Y_j \sim \mathcal{N}(\mu_j, \sigma_j^2)$.
Supposons que l'on veut estimer $\mu = \mathbb{P}[X \leq x]$ pour x fixé.

(a) MC ordinaire: générer X et calculer $I = \mathbb{I}[X \leq x]$, qui vaut 0 ou 1.

(b) CMC: générer seulement Y_2, Y_3, et calculer $J = \mathbb{P}[X \leq x \mid Y_2, Y_3]$. On a

$$X \leq x \quad \text{ssi} \quad Y_1 \geq \frac{\kappa}{x}\sqrt{\frac{Y_2^2}{w^4} + \frac{Y_3^2}{t^4}} \stackrel{\text{def}}{=} W_1(x).$$

Donc pour $x > 0$,

$$J = \mathbb{P}[X \leq x \mid Y_2, Y_3] = \mathbb{P}[Y_1 \geq W_1(x) \mid W_1(x)] = 1 - \Phi((W_1(x) - \mu_1)/\sigma_1).$$

Dans les deux cas, pour I et pour J, on génère n réalisations indépendantes de l'estimateur puis on calcule la moyenne et un intervalle de confiance. Suite dans le devoir 7.

Exemple: Réseau d'activités stochastique

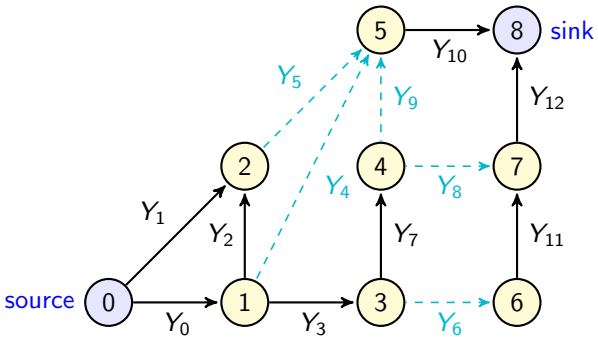
On veut estimer $\mu = \mathbb{P}[T > x]$. Estimateur naïf: $X = \mathbb{I}[T > x]$, vaut 0 ou 1.

Exemple: Réseau d'activités stochastique

On veut estimer $\mu = \mathbb{P}[T > x]$. Estimateur naïf: $X = \mathbb{I}[T > x]$, vaut 0 ou 1.

Soit $\mathcal{L} \subseteq \mathcal{A}$ un ensemble d'activités (arcs) tel que chaque chemin de la source au puits contient exactement un arc de $\mathcal{L}$. ($\mathcal{L}$ est une coupe orientée.)

Soit $\mathcal{G} = \{Y_j, j \notin \mathcal{L}\}$. Exemple: $\mathcal{L} = \{4, 5, 6, 8, 9\}$.



Estimateur CMC:

$$X_e = \mathbb{P}[T > x \mid \mathcal{G}] = \mathbb{P}[T > x \mid \{Y_j, j \notin \mathcal{L}\}] = \mathbb{P}[T > x \mid Y_0, Y_1, Y_2, Y_3, Y_7, Y_{10}, Y_{11}, Y_{12}].$$

On le calcule comme suit.

Pour chaque $l \in \mathcal{L}$, allant disons de a_l à b_l, calculer la longueur α_l du plus long chemin de la source à a_l, puis la longueur β_l du plus long chemin de b_l au puits.

Aucun chemin passant par l est plus long que x ssi $\alpha_l + Y_l + \beta_l \leq x$.

Estimateur CMC:

$$X_e = \mathbb{P}[T > x \mid \mathcal{G}] = \mathbb{P}[T > x \mid \{Y_j, j \notin \mathcal{L}\}] = \mathbb{P}[T > x \mid Y_0, Y_1, Y_2, Y_3, Y_7, Y_{10}, Y_{11}, Y_{12}].$$

On le calcule comme suit.

Pour chaque $l \in \mathcal{L}$, allant disons de a_l à b_l, calculer la longueur α_l du plus long chemin de la source à a_l, puis la longueur β_l du plus long chemin de b_l au puits.

Aucun chemin passant par l est plus long que x ssi $\alpha_l + Y_l + \beta_l \leq x$.

Conditionnellement à $\mathcal{G}$, on a cette condition avec probabilité
$\mathbb{P}[Y_l \leq x - \alpha_l - \beta_l] = F_l[x - \alpha_l - \beta_l]$.

Estimateur CMC:

$$X_e = \mathbb{P}[T > x \mid \mathcal{G}] = \mathbb{P}[T > x \mid \{Y_j, j \notin \mathcal{L}\}] = \mathbb{P}[T > x \mid Y_0, Y_1, Y_2, Y_3, Y_7, Y_{10}, Y_{11}, Y_{12}].$$

On le calcule comme suit.

Pour chaque $I \in \mathcal{L}$, allant disons de a_I à b_I, calculer la longueur α_I du plus long chemin de la source à a_I, puis la longueur β_I du plus long chemin de b_I au puits.

Aucun chemin passant par I est plus long que x ssi $\alpha_I + Y_I + \beta_I \leq x$.

Conditionnellement à $\mathcal{G}$, on a cette condition avec probabilité
$\mathbb{P}[Y_I \leq x - \alpha_I - \beta_I] = F_I[x - \alpha_I - \beta_I]$.

Puisque les Y_I sont indépendants, on obtient

$$X_e = 1 - \mathbb{P}[Y_I \leq x - \alpha_I - \beta_I \text{ pour tout } I] = 1 - \prod_{I \in \mathcal{L}} F_I[x - \alpha_I - \beta_I].$$

Pour l'exemple numérique donné en page 5, cela divise la variance environ par 4.
Et cet estimateur peut être moins coûteux à calculer que X, car moins de Y_j's à générer.

Valeurs aléatoires communes (common random numbers (CRN))

Idée: pour comparer deux (ou plusieurs) systèmes semblables, utiliser les mêmes nombres aléatoires uniformes aux mêmes endroits pour tous les systèmes.

Supposons que $\mu_1 = \mathbb{E}[X_1]$ et $\mu_2 = \mathbb{E}[X_2]$. On veut estimer $\mu_2 - \mu_1 = \mathbb{E}[X_2 - X_1]$.

Valeurs aléatoires communes (common random numbers (CRN))

Idée: pour comparer deux (ou plusieurs) systèmes semblables, utiliser les mêmes nombres aléatoires uniformes aux mêmes endroits pour tous les systèmes.

Supposons que $\mu_1 = \mathbb{E}[X_1]$ et $\mu_2 = \mathbb{E}[X_2]$. On veut estimer $\mu_2 - \mu_1 = \mathbb{E}[X_2 - X_1]$.

On simulant X_1 et X_2 avec les mêmes nombres aléatoires, on ne change pas leurs lois de probabilité individuelles, mais on peut induire une covariance positive entre les deux. On a

$$\mathrm{Var}[X_2 - X_1] = \mathrm{Var}[X_2] + \mathrm{Var}[X_1] - 2\,\mathrm{Cov}[X_1, X_2].$$

Valeurs aléatoires communes (common random numbers (CRN))

Idée: pour comparer deux (ou plusieurs) systèmes semblables, utiliser les mêmes nombres aléatoires uniformes aux mêmes endroits pour tous les systèmes.

Supposons que $\mu_1 = \mathbb{E}[X_1]$ et $\mu_2 = \mathbb{E}[X_2]$. On veut estimer $\mu_2 - \mu_1 = \mathbb{E}[X_2 - X_1]$.

On simulant X_1 et X_2 avec les mêmes nombres aléatoires, on ne change pas leurs lois de probabilité individuelles, mais on peut induire une covariance positive entre les deux. On a

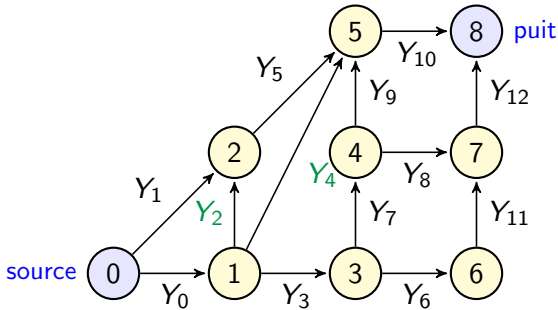
$$\mathrm{Var}[X_2 - X_1] = \mathrm{Var}[X_2] + \mathrm{Var}[X_1] - 2\,\mathrm{Cov}[X_1, X_2].$$

"Independent random numbers" (IRN): X_1 et X_2 sont indépendants.
"Common random numbers" (CRN): On utilise l'inversion avec les mêmes uniformes U_j à la même place pour simuler X_1 et X_2. On espère avoir $\mathrm{Cov}[X_1, X_2] > 0$. Parfois, on peut le prouver.

Réseau d'activités stochastique

Exemple:
On augmente $\mathbb{E}[Y_2]$ de 7.0 à 10.0,
et $\mathbb{E}[Y_4]$ de 16.5 à 18.5.

Impact sur la durée du projet?



X_1 = durée du projet selon les lois originales.
X_2 = durée du projet avec les lois modifiées.
On veut étudier la loi de $\Delta = X_1 - X_2$ et estimer $\mathbb{E}[\Delta]$.

On suppose que $Y_j = F_j^{-1}(U_j)$ pour X_1 et $\tilde{Y}_j = \tilde{F}_j^{-1}(\tilde{U}_j)$ pour X_2.

IRN: Les $\tilde{U}_j$ sont indépendants des U_j.
CRN: $\tilde{U}_j = U_j$ pour chaque j.

Essayons $n = 100,000$ répétitions pour chaque estimateur.

Avec IRN, les réalisations de Δ vont de -223.22 à 280.92.
moyenne $= 1.326$, variance $= 967$.
Intervalle de confiance à 95% pour $\mathbb{E}[\Delta]$: (1.133, 1.519).

Avec CRN, Δ va de 0 à 49.88.
moyenne $= 1.528$, variance $= 9.1$,
IC à 95% pour $\mathbb{E}[\Delta]$: (1.510, 1.547).

CRN réduit la variance par un facteur d'environ **106**.

Essayons $n = 100,000$ répétitions pour chaque estimateur.

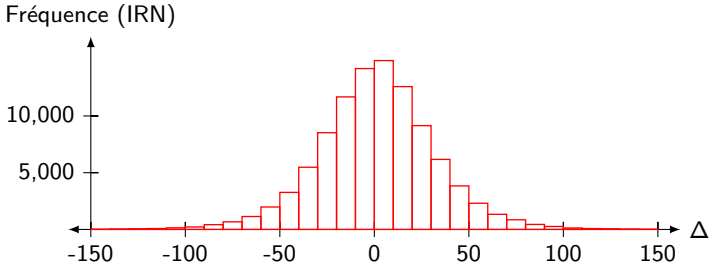
Avec IRN, les réalisations de Δ vont de -223.22 à 280.92.
moyenne $= 1.326$, variance $= 967$.
Intervalle de confiance à 95% pour $\mathbb{E}[\Delta]$: (1.133, 1.519).

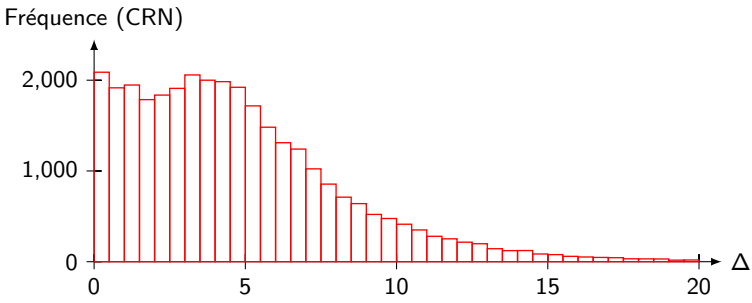
Avec CRN, Δ va de 0 à 49.88.
moyenne $= 1.528$, variance $= 9.1$,
IC à 95% pour $\mathbb{E}[\Delta]$: (1.510, 1.547).

CRN réduit la variance par un facteur d'environ **106**.

Avec CRN, 67,880 réalisations de Δ sont 0 (parce que les Y_j modifiés ne sont pas sur le plus long chemin) et les 32,120 autres réalisations de Δ sont toutes positives. Raison: quand on augmente sa moyenne θ_j, $Y_j = -\theta_j \ln(1 - U_j)$ (loi exponentielle) ne peut pas diminuer pour U_j fixé, car $-\ln(1 - U_j) > 0$. Ainsi la longueur du plus long chemin ne peut pas diminuer.

Avec IRN, on a deux valeurs indépendantes pour chaque Y_j et tout peut arriver.


Fréquence (IRN)
10,000
5,000
-150
-100
-50
0
50
100
150
Δ


Fréquence (CRN)
2,000
1,000
0
0
5
10
15
20
Δ

Un modèle d'inventaire très simple

X_j = niveau d'inventaire au matin du jour j;
D_j = demande durant le jour j, de loi uniforme sur $\{0, 1, \ldots, L\}$;
$\min(D_j, X_j)$ ventes durant le jour j;
$Y_j = \max(0, X_j - D_j)$ niveau d'inventaire à la fin du jour j;

Politique (s, S) de gestion d'inventaire: Si $Y_j < s$, on commande $S - Y_j$ items.
Chaque commande arrive le lendemain matin avec probabilité p.
Si $Y_j < s$, alors $X_{j+1} = S$ avec probabilité p, sinon $X_{j+1} = Y_j$.
On pose aussi $X_0 = S$. $\{X_j, j \geq 0\}$ est une chaîne de Markov.

Un modèle d'inventaire très simple

X_j = niveau d'inventaire au matin du jour j;
D_j = demande durant le jour j, de loi uniforme sur $\{0, 1, \ldots, L\}$;
$\min(D_j, X_j)$ ventes durant le jour j;
$Y_j = \max(0, X_j - D_j)$ niveau d'inventaire à la fin du jour j;

Politique (s, S) de gestion d'inventaire: Si $Y_j < s$, on commande $S - Y_j$ items.
Chaque commande arrive le lendemain matin avec probabilité p.
Si $Y_j < s$, alors $X_{j+1} = S$ avec probabilité p, sinon $X_{j+1} = Y_j$.
On pose aussi $X_0 = S$. $\{X_j, j \geq 0\}$ est une chaîne de Markov.

$$[\text{Revenu pour le jour } j] = [\text{ventes} - \text{coûts d'inventaire} - \text{coûts de commande}]$$
$$= c \cdot \min(D_j, X_j) - h \cdot Y_j - (K + k \cdot (S - Y_j)) \cdot \mathbb{I}[\text{la commande arrive}].$$

On cherche à optimiser les valeurs (s, S). On va les comparer avec les mêmes uniformes.
Deux séquences ("streams") de nombres aléatoires, une sous-séquence pour chacune des n répétitions. Mêmes séquences et sous-séquences pour toutes les politiques (s, S).

Modèle d'inventaire: code Java avec la librairie SSJ pour simuler m jours

```
public double simulateOneRun (int m, int s, int S,
            RandomStream streamDemand, RandomStream streamOrder) {
   // Simulates inventory model for m days, with the (s,S) policy.
   int Xj = S, Yj;          // Stock in morning and in evening.
   double profit = 0.0;     // Cumulated profit.
   for (int j = 0; j < m; j++) {
      Yj = Xj - streamDemand.nextInt (0, L); // Subtract day demand.
      if (Yj < 0) Yj = 0;             // Lost demand.
      profit += c * (Xj - Yj) - h * Yj;
      if ((Yj < s) && (streamOrder.nextDouble() < p)) {
         // We have a successful order.
         profit -= K + k * (S - Yj);
         Xj = S;
      } else
         Xj = Yj;
   }
   return profit / m;     // Average profit per day.
}
```

Comparaison de p politiques avec CRNs

```
// Simulate n runs with CRNs for p policies (s[k], S[k]), k=0,...,p-1.
RandomStream streamDemand = new MRG32k3a();
RandomStream streamOrder  = new MRG32k3a();
for (int k = 0; k < p; k++) {  // Perform n independent runs for policy k.
   for (int i = 0; i < n; i++) {
      stat_profit[k, i] = simulateOneRun (m, s[k], S[k], streamDemand, streamOrder);
            // Advance both streams to next substream, after each run.
      streamDemand.resetNextSubstream();
      streamOrder.resetNextSubstream();
   }
   streamDemand.resetStartStream();
   streamOrder.resetStartStream();
}
// Print and plot results ...
```

Comparaison de p politiques avec CRNs

```
// Simulate n runs with CRNs for p policies (s[k], S[k]), k=0,...,p-1.
RandomStream streamDemand = new MRG32k3a();
RandomStream streamOrder  = new MRG32k3a();
for (int k = 0; k < p; k++) {  // Perform n independent runs for policy k.
   for (int i = 0; i < n; i++) {
      stat_profit[k, i] = simulateOneRun (m, s[k], S[k], streamDemand, streamOrder);
            // Advance both streams to next substream, after each run.
      streamDemand.resetNextSubstream();
      streamOrder.resetNextSubstream();
   }
   streamDemand.resetStartStream();
   streamOrder.resetStartStream();
}
// Print and plot results ...
```

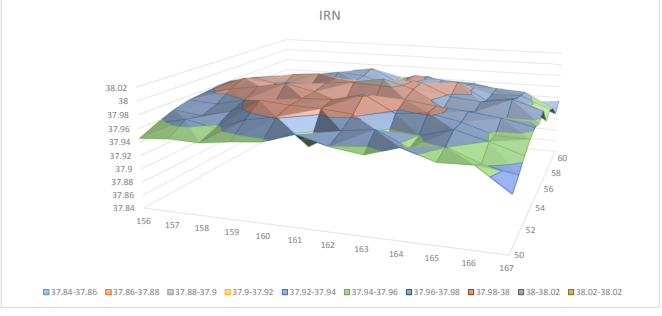
MRG32k3a avec streams et substreams en Python: https://github.com/simopt-admin/mrg32k3a.
En MATLAB: https://www.mathworks.com/help/matlab/ref/randstream.html.

On pourrait faire exécuter ces pn simulations sur des processeurs parallèles et obtenir exactement les mêmes résultats, si on utilise les mêmes séquences et sous-séquences.

Comparison with independent random numbers

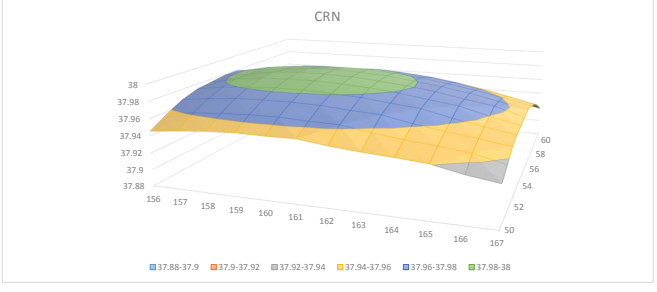
[Nabil Kemerchou made the plots.]

	156	157	158	159	160	161	162	163	164	165	166	167
50	37.94537	37.94888	37.94736	37.95314	37.95718	37.97194	37.95955	37.95281	37.96711	37.95221	37.95325	37.92063
51	37.9574	37.9665	37.95732	37.97337	37.98137	37.94273	37.96965	37.97573	37.95425	37.96074	37.94185	37.93139
52	37.96725	37.96166	37.97192	37.99236	37.98856	37.98708	37.98266	37.94671	37.95961	37.97238	37.95982	37.94465
53	37.97356	37.96999	37.97977	37.97611	37.98929	37.99089	38.00219	37.97693	37.98191	37.97217	37.95713	37.95575
54	37.97593	37.9852	37.99233	38.00043	37.99056	37.9744	37.98008	37.98817	37.98168	37.97703	37.97145	37.96138
55	37.97865	37.9946	37.97297	37.98383	37.99527	38.00068	**38.00826**	37.99519	37.96897	37.96675	37.9577	37.95672
56	37.97871	37.9867	37.97672	37.9744	37.9955	37.9712	37.96967	37.99717	37.97736	37.97275	37.97968	37.96523
57	37.97414	37.97797	37.98816	37.99192	37.9678	37.98415	37.97774	37.97844	37.99203	37.96531	37.97226	37.93934
58	37.96869	37.97435	37.9625	37.96581	37.97331	37.95655	37.98382	37.97144	37.97409	37.96631	37.96764	37.94759
59	37.95772	37.94725	37.9711	37.97905	37.97504	37.96237	37.98182	37.97656	37.97212	37.96762	37.96429	37.93976
60	37.94434	37.95081	37.94275	37.95515	37.98134	37.95863	37.96581	37.95548	37.96573	37.93949	37.93839	37.9203
61	37.922	37.93006	37.92656	37.93281	37.94999	37.95799	37.96368	37.94849	37.954	37.92439	37.90535	37.93375



Comparison with CRNs

	156	157	158	159	160	161	162	163	164	165	166	167
50	37.94537	37.94888	37.95166	37.95319	37.95274	37.95318	37.94887	37.94584	37.94361	37.94074	37.93335	37.92832
51	37.9574	37.96169	37.96379	37.96524	37.96546	37.96379	37.96293	37.95726	37.95295	37.94944	37.94536	37.93685
52	37.96725	37.97117	37.97402	37.97476	37.97492	37.97387	37.971	37.96879	37.96184	37.95627	37.95154	37.94626
53	37.97356	37.97852	37.98098	37.98243	37.98187	37.98079	37.97848	37.97436	37.97088	37.96268	37.95589	37.94995
54	37.97593	37.98241	37.98589	37.98692	37.98703	37.98522	37.9829	37.97931	37.97397	37.96925	37.95986	37.95186
55	37.97865	37.98235	37.9874	37.9894	**37.98909**	37.9879	37.98483	37.98125	37.97641	37.96992	37.96401	37.95343
56	37.97871	37.98269	37.98494	37.98857	37.98917	37.98757	37.98507	37.98073	37.97594	37.96989	37.96227	37.95519
57	37.97414	37.98035	37.98293	37.98377	37.98603	37.98528	37.98239	37.97858	37.97299	37.96703	37.95981	37.95107
58	37.96869	37.97207	37.97825	37.97944	37.97895	37.97987	37.97776	37.97358	37.96848	37.9617	37.95461	37.94622
59	37.95772	37.96302	37.9663	37.97245	37.97234	37.97055	37.9701	37.96664	37.96122	37.95487	37.94695	37.93871
60	37.94434	37.94861	37.95371	37.95691	37.96309	37.96167	37.9586	37.95678	37.95202	37.9454	37.93785	37.92875
61	37.922	37.93169	37.93591	37.94085	37.94401	37.95021	37.94751	37.94312	37.94	37.93398	37.92621	37.91742



Estimation de dérivées (sensibilité)

Supposons que la quantité μ que l'on veut estimer dépend d'un paramètre θ dans le modèle, $\mu = \mu(\theta) = \mathbb{E}_\theta[X(\theta)]$, et on veut estimer

$$\mu'(\theta_1) = \left.\frac{\partial \mu(\theta)}{\partial \theta}\right|_{\theta=\theta_1} = \left.\frac{\partial \mathbb{E}_\theta[X(\theta)]}{\partial \theta}\right|_{\theta=\theta_1} = \lim_{\delta \to 0} \frac{\mathbb{E}_{\theta_1+\delta}[X(\theta_1+\delta)] - \mathbb{E}_{\theta_1}[X(\theta_1)]}{\delta}.$$

Ce θ peut apparaitre directement dans le coût X ou encore dans les lois de probabilité.

Estimation de dérivées (sensibilité)

Supposons que la quantité μ que l'on veut estimer dépend d'un paramètre θ dans le modèle, $\mu = \mu(\theta) = \mathbb{E}_\theta[X(\theta)]$, et on veut estimer

$$\mu'(\theta_1) = \left.\frac{\partial \mu(\theta)}{\partial \theta}\right|_{\theta=\theta_1} = \left.\frac{\partial \mathbb{E}_\theta[X(\theta)]}{\partial \theta}\right|_{\theta=\theta_1} = \lim_{\delta \to 0} \frac{\mathbb{E}_{\theta_1+\delta}[X(\theta_1+\delta)] - \mathbb{E}_{\theta_1}[X(\theta_1)]}{\delta}.$$

Ce θ peut apparaitre directement dans le coût X ou encore dans les lois de probabilité.

Pourquoi estimer les dérivées?

(a) Évaluer l'importance des différents paramètres d'un modèle (par ex. pour construire un méta-modèle).
(b) Calculer un Intervalle de confiance qui tient compte de l'erreur d'estimation des paramètres du modèle.
(c) Évaluer l'effet du changement d'un paramètre de décision.
(d) Un estimateur du gradient est souvent requis dans les algorithmes d'optimisation.

On veut estimer

$$\mu'(\theta_1) = \left.\frac{\partial \mathbb{E}_\theta[X(\theta)]}{\partial \theta}\right|_{\theta=\theta_1} = \lim_{\delta \to 0} \frac{\mathbb{E}_{\theta_1+\delta}[X(\theta_1+\delta)] - \mathbb{E}_{\theta_1}[X(\theta_1)]}{\delta}.$$

Différences finies: On choisit $\delta > 0$ très petit.
On simule à $\theta = \theta_1$ pour obtenir un estimateur $X_1 = X_1(\theta_1)$ de $\mu(\theta_1)$,
puis on simule à $\theta = \theta_2 = \theta_1 + \delta$ pour obtenir un estimateur X_2 de $\mu(\theta_2)$,
et on estime la dérivée $\mu'(\theta_1)$ par $\Delta/\delta = (X_2 - X_1)/\delta$.

On veut estimer

$$\mu'(\theta_1) = \left.\frac{\partial \mathbb{E}_\theta[X(\theta)]}{\partial \theta}\right|_{\theta=\theta_1} = \lim_{\delta \to 0} \frac{\mathbb{E}_{\theta_1+\delta}[X(\theta_1+\delta)] - \mathbb{E}_{\theta_1}[X(\theta_1)]}{\delta}.$$

Différences finies: On choisit $\delta > 0$ très petit.
On simule à $\theta = \theta_1$ pour obtenir un estimateur $X_1 = X_1(\theta_1)$ de $\mu(\theta_1)$,
puis on simule à $\theta = \theta_2 = \theta_1 + \delta$ pour obtenir un estimateur X_2 de $\mu(\theta_2)$,
et on estime la dérivée $\mu'(\theta_1)$ par $\Delta/\delta = (X_2 - X_1)/\delta$.

Cet estimateur est biaisé, mais le biais $\beta \to 0$ quand $\delta \to 0$. On a aussi

$$\mathrm{Var}[\Delta/\delta] = \frac{\mathrm{Var}(X_2 - X_1)}{\delta^2} = \frac{\mathrm{Var}[X_1] + \mathrm{Var}[X_2] - 2\mathrm{Cov}[X_1, X_2]}{\delta^2}.$$

Si X_1 et X_2 sont indépendants, alors
$\mathrm{Var}[\Delta/\delta] \approx 2\mathrm{Var}[X_1]/\delta^2 = \Theta(1/\delta^2) \to \infty$ quand $\delta \to 0$.

On veut estimer

$$\mu'(\theta_1) = \left.\frac{\partial \mathbb{E}_\theta[X(\theta)]}{\partial \theta}\right|_{\theta=\theta_1} = \lim_{\delta \to 0} \frac{\mathbb{E}_{\theta_1+\delta}[X(\theta_1+\delta)] - \mathbb{E}_{\theta_1}[X(\theta_1)]}{\delta}.$$

Différences finies: On choisit $\delta > 0$ très petit.
On simule à $\theta = \theta_1$ pour obtenir un estimateur $X_1 = X_1(\theta_1)$ de $\mu(\theta_1)$,
puis on simule à $\theta = \theta_2 = \theta_1 + \delta$ pour obtenir un estimateur X_2 de $\mu(\theta_2)$,
et on estime la dérivée $\mu'(\theta_1)$ par $\Delta/\delta = (X_2 - X_1)/\delta$.

Cet estimateur est biaisé, mais le biais $\beta \to 0$ quand $\delta \to 0$. On a aussi

$$\mathrm{Var}[\Delta/\delta] = \frac{\mathrm{Var}(X_2 - X_1)}{\delta^2} = \frac{\mathrm{Var}[X_1] + \mathrm{Var}[X_2] - 2\mathrm{Cov}[X_1, X_2]}{\delta^2}.$$

Si X_1 et X_2 sont indépendants, alors
$\mathrm{Var}[\Delta/\delta] \approx 2\mathrm{Var}[X_1]/\delta^2 = \Theta(1/\delta^2) \to \infty$ quand $\delta \to 0$.
Idée: utiliser des valeurs aléatoires communes pour avoir $\mathrm{Cov}[X_1, X_2] > 0$.

Parfois, on peut prendre $\lim_{\delta \to 0}(X_2 - X_1)/\delta$ comme estimateur.

Parfois, on peut prendre $\lim_{\delta \to 0}(X_2 - X_1)/\delta$ comme estimateur.

Détails: Supposons que $X(\theta) = f(\theta, \boldsymbol{U})$ où $\boldsymbol{U} \sim \mathcal{U}(0,1)^s$ et que

$$X'(\theta) = f'(\theta, \boldsymbol{U}) = \frac{\partial f(\theta, \boldsymbol{U})}{\partial \theta} = \lim_{\delta \to 0} \frac{f(\theta + \delta, \boldsymbol{U}) - f(\theta, \boldsymbol{U})}{\delta}$$

existe avec prob.1 à θ_1. Cette dérivée stochastique $f'(\theta, \boldsymbol{U})$ est un estimateur sans biais de $\mu'(\theta)$ ssi

$$\mathbb{E}[f'(\theta, \boldsymbol{U})] \stackrel{\text{def}}{=} \mathbb{E}\left[\frac{\partial f(\theta, \boldsymbol{U})}{\partial \theta}\right] \stackrel{?}{=} \frac{\partial \mathbb{E}[f(\theta, \boldsymbol{U})]}{\partial \theta} \stackrel{\text{def}}{=} \mu'(\theta). \tag{1}$$

Parfois, on peut prendre $\lim_{\delta \to 0}(X_2 - X_1)/\delta$ comme estimateur.

Détails: Supposons que $X(\theta) = f(\theta, \boldsymbol{U})$ où $\boldsymbol{U} \sim \mathcal{U}(0,1)^s$ et que

$$X'(\theta) = f'(\theta, \boldsymbol{U}) = \frac{\partial f(\theta, \boldsymbol{U})}{\partial \theta} = \lim_{\delta \to 0} \frac{f(\theta + \delta, \boldsymbol{U}) - f(\theta, \boldsymbol{U})}{\delta}$$

existe avec prob.1 à θ_1. Cette dérivée stochastique $f'(\theta, \boldsymbol{U})$ est un estimateur sans biais de $\mu'(\theta)$ ssi

$$\mathbb{E}[f'(\theta, \boldsymbol{U})] \stackrel{\text{def}}{=} \mathbb{E}\left[\frac{\partial f(\theta, \boldsymbol{U})}{\partial \theta}\right] \stackrel{?}{=} \frac{\partial \mathbb{E}[f(\theta, \boldsymbol{U})]}{\partial \theta} \stackrel{\text{def}}{=} \mu'(\theta). \tag{1}$$

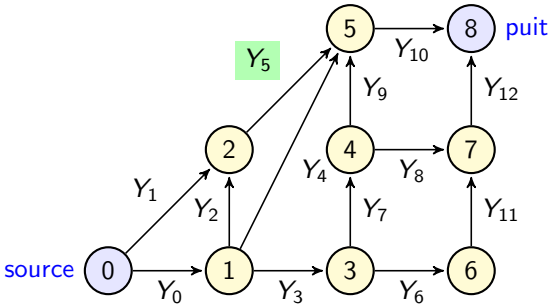
Condition suffisante: théorème de convergence dominée (TCD) de Lebesgue.
S'il existe $\delta_1 > 0$ et une v.a. Y tels que

$$\sup_{\delta \in (0, \delta_1]} \frac{|f(\theta + \delta, \boldsymbol{U}) - f(\theta, \boldsymbol{U})|}{\delta} \leq Y \tag{2}$$

et $\mathbb{E}[Y] < \infty$, alors (1) est valide.

Si plusieurs paramètres, le vecteur des dérivées est le gradient stochastique.

Exemple: réseau d'activités stochastique, $\mathbb{E}[T]$.



$Y_j = F_{j,\theta_j}^{-1}(U_j)$. Voir diapo 5 pour les F_{j,θ_j}.

Certains Y_j suivent une loi exponentielle de moyenne θ_j:

$$Y_j = Y_j(\theta_j) = -\theta_j \ln(1 - U_j).$$

Certains Y_j suivent une loi normale de moyenne θ_j et écart-type $\theta_j/4$:

$$Y_j = Y_j(\theta_j) = \theta_j + (\theta_j/4)Z_j = \theta_j + (\theta_j/4)\Phi^{-1}(U_j).$$

On veut estimer la dérivée de $\mathbb{E}[T]$ p.r. à chaque $\theta_j = \mu_j = \mathbb{E}[Y_j]$.
On considère un θ_j à la fois. On écrit $T = f_j(\theta_j, \boldsymbol{U})$ où $\boldsymbol{U} = (U_1, \ldots, U_{13})$.

On veut estimer la dérivée de $\mathbb{E}[T]$ p.r. à chaque $\theta_j = \mu_j = \mathbb{E}[Y_j]$.
On considère un θ_j à la fois. On écrit $T = f_j(\theta_j, \boldsymbol{U})$ où $\boldsymbol{U} = (U_1, \ldots, U_{13})$.

Différences finies avec CRN: On simule à θ_j et soit X_1 la valeur de T obtenue, puis on simule à $\theta_j + \delta$ et soit X_2 la valeur de T obtenue. L'estimateur de dérivée est

$$\frac{X_2 - X_1}{\delta} = \frac{f_j(\theta_j + \delta, \boldsymbol{U}) - f_j(\theta_j, \boldsymbol{U})}{\delta}.$$

Si on veut estimer tout le gradient (la dérivée pour chaque j), on peut garder le même $X_1 = f_j(\theta_j, \boldsymbol{U})$ pour tous les j, mais on doit simuler un nouveau X_2 pour chaque j. Donc $(d+1)n = 14n$ simulations au total si on répète n fois.

Dérivée stochastique: L'estimateur de dérivée est

$$\lim_{\delta \to 0} \frac{X_2 - X_1}{\delta} = \lim_{\delta \to 0} \frac{f_j(\theta_j + \delta, \boldsymbol{U}) - f_j(\theta_j, \boldsymbol{U})}{\delta} = f_j'(\theta_j, \boldsymbol{U}).$$

Quand δ est suffisamment petit, le plus long chemin demeure le même. Dans ce cas, $X_2 - X_1$ est égal au changement sur Y_j si l'arc j est sur le plus long chemin, et $X_2 - X_1 = 0$ sinon. Par conséquent, $f_j'(\theta_j, \boldsymbol{U}) = Y_j'(\theta_j)$ si l'arc j est sur le plus long chemin, et $f_j'(\theta_j, \boldsymbol{U}) = 0$ sinon.

Dérivée stochastique: L'estimateur de dérivée est

$$\lim_{\delta\to 0} \frac{X_2 - X_1}{\delta} = \lim_{\delta\to 0} \frac{f_j(\theta_j + \delta, \boldsymbol{U}) - f_j(\theta_j, \boldsymbol{U})}{\delta} = f_j'(\theta_j, \boldsymbol{U}).$$

Quand δ est suffisamment petit, le plus long chemin demeure le même. Dans ce cas, $X_2 - X_1$ est égal au changement sur Y_j si l'arc j est sur le plus long chemin, et $X_2 - X_1 = 0$ sinon. Par conséquent, $f_j'(\theta_j, \boldsymbol{U}) = Y_j'(\theta_j)$ si l'arc j est sur le plus long chemin, et $f_j'(\theta_j, \boldsymbol{U}) = 0$ sinon.

Si Y_j suit la loi exponentielle, alors $Y_j = Y_j(\theta_j) = -\theta_j \ln(1 - U_j)$, $Y_j'(\theta_j) = -\ln(1 - U_j)$ et

$$0 \le \frac{f_j(\theta_j + \delta, \boldsymbol{U}) - f_j(\theta_j, \boldsymbol{U})}{\delta} \le \frac{-\delta \ln(1 - U_j)}{\delta} = -\ln(1 - U_j) = E_j,$$

où $E_j \sim \mathrm{Expon}(1)$. Le théorème de convergence dominée s'applique car (2) est vérifiée pour $Y = E_j$. Donc $\mathbb{E}[f_j'(\theta_j, \boldsymbol{U})] = \partial \mathbb{E}[T]/\partial \theta_j$ (sans biais).

Dérivée stochastique: L'estimateur de dérivée est

$$\lim_{\delta\to 0}\frac{X_2 - X_1}{\delta} = \lim_{\delta\to 0}\frac{f_j(\theta_j+\delta, \boldsymbol{U}) - f_j(\theta_j, \boldsymbol{U})}{\delta} = f_j'(\theta_j, \boldsymbol{U}).$$

Quand δ est suffisamment petit, le plus long chemin demeure le même. Dans ce cas, $X_2 - X_1$ est égal au changement sur Y_j si l'arc j est sur le plus long chemin, et $X_2 - X_1 = 0$ sinon. Par conséquent, $f_j'(\theta_j, \boldsymbol{U}) = Y_j'(\theta_j)$ si l'arc j est sur le plus long chemin, et $f_j'(\theta_j, \boldsymbol{U}) = 0$ sinon.

Si Y_j suit la loi exponentielle, alors $Y_j = Y_j(\theta_j) = -\theta_j \ln(1-U_j)$, $Y_j'(\theta_j) = -\ln(1-U_j)$ et

$$0 \le \frac{f_j(\theta_j+\delta, \boldsymbol{U}) - f_j(\theta_j, \boldsymbol{U})}{\delta} \le \frac{-\delta\ln(1-U_j)}{\delta} = -\ln(1-U_j) = E_j,$$

où $E_j \sim \mathrm{Expon}(1)$. Le théorème de convergence dominée s'applique car (2) est vérifiée pour $Y = E_j$. Donc $\mathbb{E}[f_j'(\theta_j, \boldsymbol{U})] = \partial\mathbb{E}[T]/\partial\theta_j$ (sans biais).

Si Y_j suit la loi normale, alors $Y_j = Y_j(\theta_j) = \theta_j + (\theta_j/4)\Phi^{-1}(U_j)$ et $Y_j'(\theta_j) = 1 + \Phi^{-1}(U_j)/4$. Sans biais aussi (facile à vérifier).

Exemple: réseau d'activités stochastique, $\mathbb{P}[T > x]$.

On veut maintenant estimer la dérivée de $\mathbb{P}[T > x]$ p.r. à θ_j.

L'estimateur de $\mathbb{P}[T > x]$ est $f_j(\theta_j, \boldsymbol{U}) = \mathbb{I}[T > x]$. Ne peut prendre que les valeurs 0 et 1.

Exemple: réseau d'activités stochastique, $\mathbb{P}[T > x]$.

On veut maintenant estimer la dérivée de $\mathbb{P}[T > x]$ p.r. à θ_j.
L'estimateur de $\mathbb{P}[T > x]$ est $f_j(\theta_j, \boldsymbol{U}) = \mathbb{I}[T > x]$. Ne peut prendre que les valeurs 0 et 1.

La dérivée $f_j'(\theta_j, \boldsymbol{U})$ est toujours soit 0, soit pas définie (survient avec prob. 0).
On a donc $\mathbb{P}[f_j'(\theta_j, \boldsymbol{U}) = 0] = 1$. Estimateur biaisé de $\mu'(\theta_j) = \partial \mathbb{P}[T > x]/\partial \theta_j$.

Ici le théorème de convergence dominée ne s'applique pas car
$\sup_{\delta>0}[f_j(\theta_j + \delta, \boldsymbol{U}) - f_j(\theta_j, \boldsymbol{U})]/\delta$ n'est pas intégrable.
Le problème vient du fait que f_j est discontinue en θ_j.

Exemple: réseau d'activités stochastique, $\mathbb{P}[T > x]$.

On veut maintenant estimer la dérivée de $\mathbb{P}[T > x]$ p.r. à θ_j.
L'estimateur de $\mathbb{P}[T > x]$ est $f_j(\theta_j, \boldsymbol{U}) = \mathbb{I}[T > x]$. Ne peut prendre que les valeurs 0 et 1.

La dérivée $f_j'(\theta_j, \boldsymbol{U})$ est toujours soit 0, soit pas définie (survient avec prob. 0).
On a donc $\mathbb{P}[f_j'(\theta_j, \boldsymbol{U}) = 0] = 1$. Estimateur biaisé de $\mu'(\theta_j) = \partial \mathbb{P}[T > x]/\partial \theta_j$.

Ici le théorème de convergence dominée ne s'applique pas car
$\sup_{\delta>0}[f_j(\theta_j + \delta, \boldsymbol{U}) - f_j(\theta_j, \boldsymbol{U})]/\delta$ n'est pas intégrable.
Le problème vient du fait que f_j est discontinue en θ_j.

Solution: On peut régler ce problème en utilisant CMC:
On remplace l'indicateur $\mathbb{I}[T > x]$ par la probabilité conditionnelle

$$\mathbb{P}[T > x \mid \{Y_j, j \notin \mathcal{L}\}] = \mathbb{P}[T > x \mid Y_0, Y_1, Y_2, Y_3, Y_7, Y_{10}, Y_{11}, Y_{12}],$$

qui est continue en θ_j. Voir diapos 76–77.
La dérivée stochastique de cet estimateur donne un estimateur sans biais de la dérivée.