

Simulation et modélisation

Cours de DEA/DESS

Bernard Delyon¹

20 septembre 2002

¹IRMAR, Université Rennes-I, Campus de Beaulieu, 35042 Rennes cedex, France ; delyon@maths.univ-rennes1.fr

Table des matières

1	Générateur de suites i.i.d.	7
1.1	Problématique	7
1.2	Générateurs pour la loi $\mathcal{U}([0, 1])$	7
1.2.1	Réurrences linéaires simple	7
1.2.2	Réurrences linéaires multiples	9
1.2.3	Décalage de registre	9
1.2.4	Améliorations modernes	10
1.3	Lois non uniformes : méthodes générales	10
1.3.1	Loi discrète	10
1.3.2	Inversion	10
1.3.3	Composition	11
1.3.4	Rejet	11
1.4	Quelques lois usuelles	12
1.4.1	Loi exponentielle	12
1.4.2	Loi normale	12
1.4.3	Vecteur gaussien	13
1.4.4	Vecteur sur la sphère	13
1.4.5	Loi du χ_p^2	13
1.4.6	Autres	13
1.5	Exercices et compléments	13
2	Épreuves empiriques sur les suites pseudo-aléatoires	17
2.1	Épreuves de distribution instantannée	17
2.1.1	Test de Kolmogorov	17
2.1.2	Test du χ^2	18
2.2	Épreuves d'indépendance	18
2.2.1	Test des corrélations	19
2.2.2	Test des séries	19
2.2.3	Test des points proches	19
2.2.4	Birthday spacings	19
2.2.5	Test des lacunes (gap test)	20
2.3	Exercices	20
3	Monte Carlo : Exemples de base	21
3.1	Estimation du volume	21
3.2	Intégration	21
3.3	Perte de connexion dans un graphe	22

3.4	Décompte	22
3.4.1	Premier cas	23
3.4.2	Deuxième cas	23
3.5	Exemples applicatifs simples	25
3.5.1	Temps d'échappement d'une comète du système solaire	25
3.5.2	Files d'attente	26
3.5.3	Options	26
3.6	Exercices	27
4	Monte Carlo : réduction de variance	29
4.1	Échantillonnage préférentiel (importance sampling)	29
4.2	Variables de contrôle	31
4.3	Échantillonnage corrélé	32
4.4	Variables antithétiques	32
4.5	Échantillonnage stratifié	33
4.6	Conditionnement	34
4.7	Quasi-Monte Carlo : les suites à discrétance faible	35
4.8	Exercices	36
5	Les échantillonneurs de Metropolis et de Gibbs	39
5.1	Rappels : chaînes indécomposables et apériodiques	39
5.2	Algorithme de Metropolis	41
5.2.1	Principes	41
5.2.2	Simulation de graphes.	42
5.2.3	Marche aléatoire sans recoupement	42
5.3	L'échantillonneur de Gibbs	43
5.4	Applications	43
5.4.1	Maximum de vraisemblance	44
5.4.2	L'algorithme EM pour les données manquantes	44
5.5	Exercices, exemples	46
5.5.1	Chaînes de Markov	46
5.5.2	Échantillonnage de Gibbs	46
5.5.3	Algorithme de Metropolis	48
6	Modèles markoviens	49
6.1	Généralités	49
6.2	Simulation de processus à événements discrets	49
6.2.1	Simulation des processus de saut	50
6.2.2	Simulation d'une file à un serveur	51
6.2.3	Simulation d'une file à deux serveurs en série	52
6.2.4	Simulation à état non borné	52
6.2.5	Gestion d'inventaire	52
6.3	Champs de Markov	53
6.3.1	Définition	53
6.3.2	Échantillonnage de Gibbs	55
6.3.3	Algorithme de Metropolis	55
6.3.4	Exemples où \mathcal{X} est infini	55
6.4	Modèles spatiaux dynamiques	57

6.5	Exercices	58
6.5.1	Processus à événements discrets	58
6.5.2	Champs Markoviens	59
6.5.3	Modèles spatiaux dynamiques	60
7	Simulation de processus : convergence	61
7.1	Algorithme des répétitions	61
7.2	Élimination du transitoire	62
7.3	Régénération	62
7.3.1	Renouvellement	62
7.3.2	Théorie générale	64
7.3.3	Méthode atomique	65
7.4	Échantillonnage parfait : couplage sur le passé	66
7.4.1	Cas général	67
7.4.2	Cas monotone	68
7.5	Réduction de variance	68
7.6	Exercices et compléments	69
8	Optimisation par Monte-Carlo	71
8.1	Approximation stochastique	71
8.1.1	Deux exemples simples	71
8.1.2	Forme générale et la trajectoire moyenne	72
8.1.3	La régression linéaire	73
8.1.4	Exemple : Optimisation d'une chaîne de Markov contrôlée	74
8.1.5	Exemple : Données manquantes	75
8.1.6	Exemple : Algorithme de Kiefer-Wolfowitz	75
8.1.7	Cadre général à dynamique markovienne	76
8.1.8	Algorithmes de poursuite	76
8.2	Recuit simulé	77
8.3	Exercices	78
9	Simulation d'équations différentielles stochastiques	79
9.1	Introduction	79
9.2	Le cas déterministe : $\sigma = 0$	79
9.3	Les deux mesures de vitesse de convergence	81
9.4	Schémas d'intégration pour les eds	81
9.4.1	Construction du schéma	81
9.4.2	Schéma de Milstein	81
9.4.3	Convergence forte du schéma d'Euler	82
9.4.4	Convergence faible du schéma d'Euler	82
9.5	Résumé	83
9.6	Exercices de simulation. Exemples	84
9.6.1	Martingale exponentielle.	84
9.6.2	Pont brownien.	84
9.6.3	Oscillateur de Van der Pol.	84
9.6.4	Dynamique des populations.	85

10 Bootstrap	87
10.1 Introduction	87
10.2 Estimation par répétitions	87
10.3 Principes de base du bootstrap	88
10.4 Exemples	89
10.5 Échec du bootstrap dans un cas non-régulier	91
10.6 Variantes et détails pratiques	91
10.6.1 Bootstrap régularisé.	91
10.6.2 Bootstrap semi-paramétrique.	92
10.6.3 Normalisation pivotale.	92
10.6.4 Bootstrap par sous-échantillonnage	92
10.7 Exercices et compléments	93

Chapitre 1

Générateur de suites i.i.d.

1.1 Problématique

Soit une distribution sur \mathbb{R} ou \mathbb{R}^d , il s'agit dans ce chapitre de simuler une suite de variables i.i.d. Y_n possédant cette loi. Il y a donc deux contraintes à satisfaire ;

- ▶ Les Y_n doivent avoir la bonne distribution
- ▶ Les Y_n doivent être indépendantes.

Nous ne donnons ici qu'un aperçu culturel rapide des méthodes utilisées. Une bonne référence est le livre de Knuth [29]. L'article [32] contient des informations très intéressantes également, avec des références plus actuelles.

Une compréhension plus approfondie des aspects théoriques de la question passe par le concept de complexité de Kolmogorov, point qui ne sera pas abordé ici.

1.2 Générateurs pour la loi $\mathcal{U}([0, 1])$

Les nombres machine compris entre 0 et 1 étant tous de la forme $x = n2^{-p}$ $n \in \{0, \dots, 2^p\}$ il n'y a pas plus de 2^p nombres différents sur $[0, 1]$, et comme ces nombres sont également espacés, le problème se réduit à tirer des entiers uniformément sur $\{0, 1, \dots, 2^p - 1\}$. Typiquement $p = 32$.

On verra que les algorithmes de génération de suites $(U_n)_{n \geq 0}$ uniformes auront toujours plus ou moins la forme $X_n = f(X_{n-1})$, $X_n \in \{0, 1, \dots, m\}$, $U_n = X_n/m$. Il y a donc deux conséquences importantes

- ▶ Les U_n ne seront pas indépendantes : la suite est pseudo-aléatoire.
- ▶ La suite U_n sera périodique de période au plus m , et parfois moins.

1.2.1 Récurrences linéaires simple

Il s'agit de générateurs de la forme¹ :

$$X_i = aX_{i-1} + c \pmod{m}. \tag{1.1}$$

avec $U_i = X_i/m$. La période d'un tel générateur est bien entendu inférieure à m . Si $c = 0$ sa période est même inférieure ou égale à $m - 1$ car 0 est point fixe. On ne considérera que trois cas, qui sont les plus importants dans la pratique.

¹Dans toute le suite on note $x = y \pmod{m}$ pour définir x à partir de y par les conditions $x \equiv y \pmod{m}$ et $0 \leq x < m$.

Théorème 1 Si $c \neq 0$ la période du générateur (1.1) est égale à m si et seulement si

1. c et m sont premiers entre eux
2. tout diviseur premier de m divise $a - 1$
3. si $4|m$ alors $4|(a - 1)$.

Théorème 2 Si $c = 0, m = 2^\beta$ et $\beta \geq 4$, le générateur (1.1) a une période (maximale) égale à $m/4$ si et seulement si x_0 est impair et $a \equiv 3$ ou $5 \pmod{8}$.

Ce théorème est une conséquence du précédent (cf exercice 1.5.2).

Théorème 3 Si $c = 0$ et m premier le générateur (1.1) a une période (maximale) égale à $m - 1$ ssi $x_0 \neq 0$ et a est une racine primitive de m :

$$a \neq 0 \text{ et pour tout } p \text{ diviseur premier de } m - 1 : a^{(m-1)/p} \not\equiv 1 \pmod{m}.$$

Notons que $m = 2^{31} - 1$ est premier. Voici quelques exemples (voir aussi [13]) :

Lehmer (1948)	$a = 23, m = 10^8 + 1$	assez moyen
RANDU (IBM 1968)	$a = 65539, m = 2^{31}$	mauvais
Marsaglia (1972)	$a = 69069, m = 2^{32}$	raisonnable
SURAND (1968)	$a = 16807, m = 2^{31} - 1$	raisonnable
INMOS [29]	$a = 1664525, m = 2^{32}$	le meilleur a connu pour ce m
L'Ecuyer (1993)	$a = 41358, m = 2^{31} - 1$	bon
CDC	$a = 5^{15}, m = 2^{47}$	longue période

TAB. 1.1 – Quelques valeurs utilisées pour m et a , ($c = 0$) [25]

Critère spectral. La période n'est pas tout. Par exemple le générateur avec $a = c = 1$ a une période maximale mais fournit une suite U_n loin d'être indépendante. Une façon de le voir est d'observer que la distribution de (U_n, U_{n+1}) n'est pas du tout uniforme sur le carré, puisque $U_{n+1} - U_n \equiv m^{-1} \pmod{1}$.

De même, dans le cas du générateur RANDU des vieux IBM, associé à $a = 65539, m = 2^{31}$, on observe que $U_{n+2} - 6U_{n+1} + 9U_n \equiv 0 \pmod{1}$; ceci vient de ce que $65539 = 2^{16} + 3$.

Les chercheurs ont observé que les générateurs linéaires ont tendance à placer les vecteurs $V_i = (U_{i+1}, \dots, U_{i+d})$ sur la réunion des hyperplans parallèles de la forme $H_n = \{v \in \mathbb{R}^d : \langle \xi, v \rangle = n\}$ pour un certain vecteur $\xi = (\xi_1, \dots, \xi_d)$, ce qui signifie que $\langle \xi, V_i \rangle \equiv 0 \pmod{1}$ pour tout i . Par exemple, si $d = 2$, on voit tout les points (U_n, U_{n+1}) sur des droites parallèles équi-espacées. Il y a même de nombreux vecteurs ξ ayant cette propriété. Plus les H_n sont espacés, plus le générateur est de mauvaise qualité, car il y a de grande zones de l'espace non visitées. La distance entre deux hyperplans successifs étant la distance de 0 à H_1 qui vaut $\|\xi\|^{-1}$, on considérera le critère spectral

$$h_d = \max_{\xi \neq 0} \{ \|\xi\|^{-1} : \sum_{j=1}^d \xi_j U_{i+j} \equiv 0 \pmod{1}, i = 1, 2, \dots \}.$$

On calcule alors

$$\mu_d = \frac{\pi^{d/2} h_d^{-d}}{P \Gamma(d/2 + 1)}$$

où P est la période. On considère que pour les bons générateur, μ_d est voisin ou supérieur à 1 et que si μ_d est inférieur à 0,1 le générateur est assez mauvais [35].

1.2.2 Récurrences linéaires multiples

Elles sont définies par la relation :

$$X_i = \sum_{j=1}^k a_j X_{i-j} \pmod{m}.$$

avec bien entendu $U_i = X_i/m$. La période d'un tel générateur est $m^k - 1$ si m est premier et a_1, \dots, a_k bien choisis. Noter que la période est bien plus grande que le nombre de valeurs prises.

Un exemple est le générateur de Marsaglia et Zaman : $X_n = X_{n-r} - X_{n-s} \pmod{b}$. Les valeurs proposées dans [35] sont $b = 2^{24}$, $r = 24$, $s = 10$.

1.2.3 Décalage de registre

On opère directement sur la représentation binaire des réels de $[0, 1]$:

$$u = 0 + \frac{b_1}{2} + \frac{b_2}{4} + \dots + \frac{b_l}{2^l} = \langle 0, b_1, \dots, b_l \rangle.$$

On réalise une suite de $b_i \in \{0, 1\}$ par la récurrence

$$b_i = a_1 b_{i-1} + \dots + a_q b_{i-q} \pmod{2}, \quad a_i \in \{0, 1\}$$

puis $u_i = \langle 0, b_{il+1}, b_{il+2}, \dots, b_{il+l} \rangle$, soit :

$$\underbrace{b_0 \dots b_{l-1}}_{u_0} \underbrace{b_l \dots b_{2l-1}}_{u_1} \underbrace{b_{2l} \dots b_{3l-1}}_{u_2} \dots$$

Il y a 2^q suites $(b_{i-1}, \dots, b_{i-q})$ différentes possibles, et comme zéro est point fixe, la période des b_i est au plus de $2^q - 1$; on choisira toujours un générateur ayant cette période. Pour que la période de la suite u_i soit également $2^q - 1$, il faudra utiliser effectivement toutes les suites $(b_{i+1}, \dots, b_{i+l})$ apparaissant (et non pas une sur l), et pour cela on prendra un l tel que $\text{pgcd}(l, 2^q - 1) = 1$.

Exemples. On a beaucoup étudié les générateurs avec des récurrences à deux termes

$$b_i = b_{i-p} + b_{i-q} \pmod{2} \tag{1.2}$$

($p < q$) pour lesquels la période est de $2^q - 1$. Par exemple les paires

$$q = 31, \quad p = 3, 6, 7, 13, 18, 24, 25, 28$$

$$q = 607, \quad p = 273.$$

Un choix recommandé est $q = 31, p = 13, l = 32$.

Critère d'équirépartition. En plus de la période, on peut vérifier théoriquement la bonne répartition des vecteurs :

On calcule pour chaque cellule dyadique de $[0, 1]^d$ de côté $2^{-\lambda}$, avec $\lambda \leq l$ et $d < q$ le nombre de vecteurs de la forme $(U_{i+1}, \dots, U_{i+d})$ qui sont tombés dedans, sur une période entière du générateur (on ajoute le vecteur $(0, \dots, 0)$, ce qui fait en tout exactement 2^q vecteurs).

Si $\lambda \leq [q/d]$ et toutes les cellules ont exactement $2^{q-d\lambda}$, alors on dit qu'il y a *équirépartition maximale pour λ et d* (on vérifie simplement que c'est impossible pour l plus grand). Si $\lambda > [q/d]$ et chaque cellule ne contient pas plus d'un point, on dit qu'il y a *absence de collision*. Il se trouve que de nombreux générateurs à décalage de registre fournissent des suites ayant ces deux propriétés pour diverses valeurs de (d, λ) [32].

1.2.4 Améliorations modernes

Il existe deux façons d'améliorer un générateur. La manière courante est de coupler des générateurs différents. Par exemple le générateur suivant a de bonnes propriétés :

$$\begin{aligned}X_{n+1} &= 40014X_n \pmod{2^{31} - 85} \\Y_{n+1} &= 40692Y_n \pmod{2^{31} - 249} \\Z_{n+1} &= X_n - Y_n \pmod{2^{31} - 86}.\end{aligned}$$

Il s'agit du premier générateur parmi les 32 proposés par L'Écuyer [30] (générateur "grand" de Scilab). L'intérêt est principalement de rallonger la période sans trop compliquer les algorithmes, car, sous certaines hypothèses, les périodes se multiplient (on a donc ici une période d'ordre 10^{18}). On voit se dessiner une forme plus générale qui est une forme d'état [33] :

$$\begin{aligned}X_{n+1} &= AX_n \pmod{m} \\U_n &= C^T X_n \pmod{1}\end{aligned}$$

où X_n , m et C sont des vecteurs et A est une matrice. On arrive ainsi à réaliser des générateurs de période $2^{19937} - 1$ où X est de longueur $623 \times 32 + 1 = 19937$ bits, $m = 2$ et U_n est formé à l'aide des 32 derniers bits de X_n [38].

Générateurs non-linéaires. Un exemple récent d'un tel générateur est [32]

$$\begin{aligned}X_{n+1} &= aX_n^3 + 1 \pmod{m} \\X'_{n+1} &= a'X_n'^3 + 1 \pmod{m'} \\U_n &= (X_n/m + X'_n/m') \pmod{1}\end{aligned}$$

La période vaut au mieux mm' . Par exemple : $m = 65519$, $m' = 65447$, $a = 512$, $a' = 27076$.

1.3 Lois non uniformes : méthodes générales

1.3.1 Loi discrète

Soit une variable à générer qui prend les valeurs $1, 2, \dots, r$ avec probabilité p_1, \dots, p_r : $P(X = i) = p_i$. On commence par générer une suite uniforme indépendante U_n puis on pose

$$X_n = i \text{ si } p_1 + \dots + p_{i-1} < U_n \leq p_1 + \dots + p_i.$$

Si les U_n sont i.i.d $\mathcal{U}([0, 1])$, les X_n sont clairement indépendantes avec la loi voulue.

En pratique on testera successivement $p_1 > U_n$ puis, $p_1 + p_2 > U_n$, etc... Pour que la simulation soit la plus rapide possible, il faudra donc présenter les p_i par ordre décroissant.

1.3.2 Inversion

Si la loi à générer possède une fonction de répartition $F(x) = P(X \leq x)$ continue strictement croissante et si Q est la fonction inverse de F (fonction quantile) alors la variable $Q(U)$, avec $U \sim \mathcal{U}([0, 1])$, suit la loi F :

$$P(Q(U) \leq x) = P(U \leq F(x)) = F(x).$$

Pour un F général, cette méthode fonctionne encore en définissant Q continue à gauche par (exercice 1.5.16)

$$Q(u) = \min\{y : F(y) \geq u\}. \quad (1.3)$$

Cette méthode ne s'étend pas simplement aux variables vectorielles. Pour simuler un vecteur $X_n = (X_{n1}, \dots, X_{nd})$ de variables non-indépendantes avec la méthode d'inversion, il faut se ramener à des variables réelles, par exemple : simuler d'abord X_{n1} selon sa loi marginale, puis X_{n2} selon sa loi conditionnelle à X_{n1} , puis X_{n3} selon sa loi conditionnelle à X_{n1} et X_{n2} , etc...

1.3.3 Composition

C'est le cas où l'on a un mélange de lois, c'est-à-dire que la distribution est de la forme $P_X(dx) = \sum_{i=1}^r p_i P_i(dx)$, ce qui revient à dire que la densité satisfait (si elle existe) $f(x) = \sum_{i=1}^r p_i f_i(x)$. Par exemple X est la taille d'un individu pris au hasard dans r pays différents, P_i est la distribution de la taille dans le i -ième pays et p_i est la population relative de ce pays.

ALGORITHME

1. Générer une v.a. J à valeurs dans $\{1, 2, \dots, r\}$ avec probabilités p_1, \dots, p_r
2. Générer ensuite X selon P_J .

En effet : $P(X \in A) = \sum_i P(X \in A | J = i) P(J = i) = \sum_i P_i(A) p_i = P_X(A)$.

1.3.4 Rejet

On va montrer qu'une approximation, même grossière, de la loi à simuler par une loi facilement simulable peut suffire à générer facilement des échantillons.

Soit à simuler la variable X de loi de densité $f(x)$ par rapport à une certaine mesure $\mu(dx)$ (typiquement $\mu(dx) = dx$), avec la représentation

$$f(x) = C^{-1} g(x) f_0(x), \quad C = \int g(x) f_0(x) \mu(dx)$$

avec les propriétés

- $P_0(dx) = f_0(x) \mu(dx)$ est une loi facilement simulable
- $0 \leq g \leq 1$.

Par exemple, $f(x) = c_0 e^{-|x| - \sqrt{|x|+1}}$, $f_0(x) = e^{-|x|}/2$, $g(x) = e^{1 - \sqrt{|x|+1}}$.

ALGORITHME

1. Tirer U et Y indépendantes selon $\mathcal{U}([0, 1])$ et P_0
2. Si $U \leq g(Y)$ alors on a un nouvel échantillon $X = Y$ sinon recommencer.

Théorème 4 *L'algorithme simule bien la loi de X .*

DÉMONSTRATION: Si l'on applique l'algorithme n fois et que l'on génère k_n échantillons de X (k_n est une variable aléatoire), on a

$$\frac{1}{k_n} \sum_{j=1}^{k_n} \varphi(X_j) = \frac{\frac{1}{n} \sum_{i=1}^n \varphi(Y_i) 1_{U_i \leq g(Y_i)}}{\frac{1}{n} \sum_{i=1}^n 1_{U_i \leq g(Y_i)}}.$$

Par application de la loi des grands nombres le numérateur converge vers

$$\int \varphi(y) 1_{0 \leq u \leq g(y)} du f_0(y) \mu(dy) = \int \varphi(y) g(y) f_0(y) \mu(dy).$$

Le dénominateur correspondant à $\varphi = 1$, la limite du rapport est

$$\frac{\int \varphi(y) g(y) f_0(y) \mu(dy)}{\int g(y) f_0(y) \mu(dy)} = \int \varphi(y) f(y) \mu(dy).$$

■

Dans le cas où g est la fonction indicatrice d'un ensemble B , l'algorithme se simplifie en une forme dont la signification intuitive est claire : *Tirer des échantillon selon P_0 et ne garder que ceux qui appartiennent à B .*

Le cas où P_0 est uniforme sur une partie B de \mathbb{R}^d correspond à la méthode du rejet traditionnelle :

1. Tirer Y uniformément sur B et $U \sim \mathcal{U}([0, 1])$
2. Si $U \leq g(Y)$ alors on a un nouvel échantillon $X = Y$ sinon recommencer.

Aspects pratiques. Les rejets seront nombreux si g est petite sur un ensemble de grande probabilité sous P_0 . On vérifie que la probabilité de rejet est $1 - C$ (exercice 1.5.11).

En général, on part de f , on cherche f_0 simulable qui ressemble à f , puis g d'en déduit (proportionnelle à f/f_0 et inférieure à 1).

Si g est difficile à calculer mais peut être approximée par deux suites $g_n^- \leq g \leq g_n^+$ respectivement croissantes et décroissantes convergeant vers g , le test $U \geq g(Y)$ peut se faire en cherchant un n assez grand pour que $g_n^-(Y)$ et $g_n^+(Y)$ soient du même côté de U , ce qui permet de réaliser un algorithme en temps fini [11].

1.4 Quelques lois usuelles

1.4.1 Loi exponentielle

L'application de la méthode d'inversion donne immédiatement

$$X = -\log(U), \quad U \sim \mathcal{U}([0, 1]).$$

1.4.2 Loi normale

Une méthode classique consiste à générer les variables par paires.

1. Générer $U \sim \mathcal{U}([0, 1])$ et V exponentielle de paramètre 1
2. On obtient deux v.a. normales indépendantes en posant

$$X_1 = \sqrt{2V} \cos(2\pi U), \quad X_2 = \sqrt{2V} \sin(2\pi U).$$

En effet pour toute fonction continue bornée f

$$\begin{aligned} E[f(X_1, X_2)] &= \int_0^{+\infty} \int_0^1 f(\sqrt{2v} \cos(2\pi u), \sqrt{2v} \sin(2\pi u)) e^{-v} dv du \\ &= \int_0^{+\infty} \int_0^{2\pi} f(\rho \cos \theta, \rho \sin \theta) e^{-\rho^2/2} \rho d\rho \frac{d\theta}{2\pi} \\ &= \int f(x, y) e^{-(x^2+y^2)/2} \frac{dx dy}{2\pi} \end{aligned}$$

En particulier on a vu que pour $(X_1, X_2) \sim \mathcal{N}(0, I)$, la loi de $X_1^2 + X_2^2$ est exponentielle de paramètre 1/2.

1.4.3 Vecteur gaussien

Soit à simuler la loi $\mathcal{N}(\mu, R)$. Alors simuler $X_0 = \mathcal{N}(0, I)$ à l'aide de l'algorithme précédent et poser $X = \mu + MX_0$, où M est une racine carrée de $R : MM^T = R$. Clairement X est un vecteur gaussien de moyenne et variance correctes.

1.4.4 Vecteur sur la sphère

L'invariance de la loi gaussienne par rotation implique que si $X \sim \mathcal{N}(0, I)$, alors $X/\|X\|$ est uniformément réparti sur la sphère.

1.4.5 Loi du χ_p^2

C'est la loi de $X_1^2 + \dots + X_p^2$ où $(X_1, \dots, X_p) \sim \mathcal{N}(0, I)$. Si p est pair c'est donc la loi de la somme de $p/2$ variables exponentielles de paramètre 1/2, sinon, ajouter le carré d'une normale au χ_{p-1}^2 .

1.4.6 Autres

Il existe quantité de méthodes astucieuses pour simuler les différentes lois. Voir par exemple [11]. Voir aussi les exercices 1.5.15 et 1.5.18.

1.5 Exercices et compléments

Exercice 1.5.1 Montrer qu'un générateur de la forme $X_n = f(X_{n-1}, X_{n-2})$, où f est à valeurs dans $\{0, 1, \dots, m-1\}$ est périodique avec une période au plus de m^2 . Que peut-on dire si $f(0, 0) = 0$? si $f(x, 0) = f(0, x)$ pour tout x ?

Exercice 1.5.2 On considère le théorème 2.

1. Montrer que si x_0 est impair et $a \equiv 5 \pmod{8}$, alors la période vaut $m/4$. On remarquera qu'en posant $b = x_0 \pmod{4}$, les $x_i - b$ sont multiples de 4 et la suite $y_i = (x_i - b)/4$ satisfait une certaine récurrence.
2. Si $a \equiv 3 \pmod{8}$, faire de même en posant $y_i = (x_i - x_0 + 2)/4$.

Exercice 1.5.3 On considère l'algorithme du § 1.3.1. Quel est le nombre moyen de tests faits pour générer une variable aléatoire ?

Exercice 1.5.4 Soient U et V i.i.d $\mathcal{U}([0, 1])$; quelle est la répartition (simple) de la variable $(\sqrt{U} \cos(2\pi V), \sqrt{U} \sin(2\pi V))$?

Exercice 1.5.5 On considère la densité $f(x, y) = xe^{-xy}1_{y>0}1_{0<x<1}$ et une paire de v.a. (X, Y) suivant cette loi.

1. Quelle est la loi de Y sachant $X = x$?
2. Quelle est la loi de X ?
3. Proposer un procédé de simulation de (X, Y) .

Exercice 1.5.6 On considère la densité $f(x, y) = yx^{y-1}e^{-y}1_{y>0}1_{0<x<1}$ et une paire de v.a. (X, Y) suivant cette loi.

1. Quelle est la loi de Y ?
2. Que vaut $P(X \leq x|Y = y)$?
3. Proposer un procédé de simulation de (X, Y) .

Exercice 1.5.7 Soit la densité $f(x, y) = \frac{1}{\sqrt{8\pi}} e^{-y^2 x/2} e^{-\sqrt{x}} 1_{x>0}$ et la paire (X, Y) de variables aléatoires suivant cette loi.

1. Quelle est la loi de Y sachant $X = x$?
2. Quelle est la loi de \sqrt{X} ?
3. Proposer un procédé de simulation de (X, Y) .

Exercice 1.5.8 On sait générer une variable X de fonction de répartition F_X ainsi que Y de fonction de répartition F_Y . Quelle est la fonction de répartition de $\max(X, Y)$? En déduire une façon de générer une v.a. de fonction de répartition $\min(x, 1)(1 - e^{-x})1_{x>0}$. Tracer la densité de cette variable.

Exercice 1.5.9 Proposer une méthode de rejet pour fabriquer des variables uniformes sur le disque unité avec des $\mathcal{U}([0, 1])$ sans utiliser de fonction trigonométrique.

Exercice 1.5.10 Inversion approchée [11]. On veut simuler une variable aléatoire X de densité f , ayant à sa disposition une approximation ψ croissante de la fonction quantile Q (cf § 1.3.2).

1. Quelle doit être la densité h de la loi de U pour que $\psi(U)$ suive la loi de X . Vérifier la solution obtenue si $\psi = Q$.
2. Proposer une méthode de rejet pour simuler de telles variables U .

Exercice 1.5.11 On considère la méthode de rejet du § 1.3.4.

1. Démontrer que la probabilité de rejet est $1 - C$.
2. On veut simuler une variable $\mathcal{N}(0, 1)$ avec la méthode de rejet en utilisant pour f_0 la loi de Laplace de paramètre λ . Exprimer g et C (en normalisant de sorte que $\sup g = 1$) et en déduire la valeur (simple) de λ qui va minimiser la probabilité de rejet ; vérifier que cette dernière vaut alors approximativement 0,24.

Exercice 1.5.12 Quelle est la distribution des échantillons rejetés dans la méthode de rejet ?

Exercice 1.5.13 Soit $\lambda < 1$. Utiliser la méthode du rejet pour simuler des variables de loi de Poisson de paramètre λ ($P(X = n) = \lambda^n e^{-\lambda}/n!$) à partir de variables de probabilité $P(Y = n) = \lambda^n(1 - \lambda)$. Calculer la probabilité de rejet.

Exercice 1.5.14 (Mélange avec poids négatifs ; méthode de Bignami et de Matteis). Soit une densité de la forme $f(x) = \sum_i p_i f_i(x)$ où chaque f_i est une densité de probabilité et $\sum p_i = 1$. Les p_i ne sont pas tous positifs (ceci arrive en estimation de densité), et l'on pose $p_i = p_i^+ - p_i^-$ où $p_i^+ = p_i$ si $p_i > 0$ et 0 sinon. Expliquer pourquoi la simulation de la loi $\sum_i p_i^+ f_i(x) / \sum p_i^+$ est bien plus simple que celle de la loi $f(x)$. En déduire une méthode de rejet pour simuler f , dont la probabilité de rejet est $\sum p_i^- / \sum p_i^+$.

Exercice 1.5.15 Soit E_1, E_2, \dots une suite de v.a. exponentielles de paramètre 1, vérifier que la variable aléatoire dont la valeur est l'indice n tel que

$$E_1 + E_2 + \dots + E_n \leq \lambda < E_1 + E_2 + \dots + E_{n+1}$$

suit une loi de Poisson de paramètre λ ($p_n = e^{-\lambda} \lambda^n / n!$). Pour cela on pourra démontrer et utiliser que $\int_{\mathbb{R}_+^n} 1_{x_1 + \dots + x_n < 1} dx_1 \dots dx_n = 1/n!$ (la démonstration se fait par changement de variables).

En déduire une méthode pour simuler une loi de Poisson. Cette méthode n'est utilisée que pour λ relativement petit ; pourquoi ?

Exercice 1.5.16 Montrer que le min de (1.3) est atteint. En déduire que $Q(u) \leq x$ équivaut à $F(x) \geq u$, $0 \leq u \leq 1$, puis que $Q(U)$ a pour fonction de répartition F si $U \sim ([0, 1])$.

Exercice 1.5.17 Soit $F_-(x)$ la limite à gauche de F . Montrer que la v.a. U obtenue en tirant X selon F , puis U uniformément sur $[F_-(X), F(X)]$ est uniforme sur $[0, 1]$.

Exercice 1.5.18 Soient $\lambda, \mu > 0$. Montrer que le procédé suivant simule l'inverse gaussienne (loi de densité $\sqrt{\lambda} \exp \left\{ -\frac{\lambda(x-\mu)^2}{2\mu^2 x} \right\} \frac{dx}{\sqrt{2\pi x^3}}$ [11]) :

1. Simuler $Y \sim \mathcal{N}(0, 1)$.
2. Calculer les deux solutions X_- et X_+ de l'équation $y^2 = \lambda(x - \mu)^2 / (\mu^2 x)$ pour $y = Y$.
3. Choisir $X = X_-$ avec probabilité $p = (1 + X_- / \mu)^{-1}$ et $X = X_+$ avec probabilité $1 - p$.

Indications : Noter que

$$E[f(X)] = E[f(x_-(Y))p(Y)] + E[f(x_+(Y))(1 - p(Y))]$$

où $x_-(y)$ est la plus petite racine de l'équation du second degré. Pour effectuer les calculs, se garder de résoudre cette équation, mais noter que la fonction $x_-(y)$ est bijective de \mathbb{R}_+ dans $]0, \mu]$ et que l'équation donne tout de suite une relation entre dx_- et dy , qui permet d'effectuer simplement le changement de variable. Tout se passe de la même façon pour x_+ (noter que $x_+ x_- = \mu^2$ et donc $1 - p = (1 + x_+ / \mu)^{-1}$).

Chapitre 2

Épreuves empiriques sur les suites pseudo-aléatoires

Les épreuves servent à vérifier la qualité des suites pseudo-aléatoires. Elles sont faites pour être passées avec succès pour les suites i.i.d de loi F désirée. Ces épreuves s'ajoutent à d'éventuels critères, plus spécifiques au type de générateur, comme les deux vus au § 1.2.1 ou au § 1.2.3.

Il y a deux types extrêmes d'épreuves : le premier se centre sur la vérification du fait que la distribution est bien celle attendue et l'autre sur la vérification de l'indépendance des variables. C'est le deuxième point qui est de loin le plus délicat, aussi bien du point de vue de la réalisation des suites que de celui la vérification par des épreuves.

Noter toutefois qu'une épreuve conçue pour vérifier que la loi de trois variables simulées consécutives est bien la loi de trois répliques indépendantes de la loi F participe de ces deux types.

On trouve des applications de certains des tests qui suivent dans l'article [31].

2.1 Épreuves de distribution instantannée

L'idée est de supposer que la suite simulée est bien une suite i.i.d et d'utiliser des tests statistiques classiques pour s'assurer que la loi est bien la bonne.

2.1.1 Test de Kolmogorov

Soit la fonction de répartition empirique

$$F_n(x) = \frac{1}{n} \sum_{i=1}^n 1_{X_i \leq x}$$

et la statistique

$$d_n = \sqrt{n} \sup_x |F_n(x) - F(x)|$$

alors on a le théorème de Kolmogorov et Smirnov

Théorème 5 *Si les X_i sont i.i.d de loi donnée par F supposée continue, alors la loi de d_n est indépendante de F avec asymptotiquement :*

$$\lim_{n \rightarrow \infty} P(d_n \leq x) = \sum_{k=-\infty}^{k=+\infty} (-1)^k e^{-2k^2 x^2} = H(x)$$

Il s'ensuit que $H(x)$ est la probabilité de confiance du test décidant la loi significativement non-conforme si $d_n > x$. En pratique on compare le niveau $\alpha = 1 - H(d_n)$ à un seuil donné à l'avance toujours pris inférieur à 5%.

On montre en exercice 2.3.2 que si F n'est pas continue, le test basé sur d_n aura un niveau supérieur ; on peut donc très bien l'utiliser dans ce contexte.

Notons que le fait que la statistique est indépendante de F est lié aux résultats du chapitre précédent ; en effet, posons $U_i = F(X_i)$, alors les U_i sont $\mathcal{U}([0, 1])$ et

$$F_n(x) = n^{-1} \sum_1^n 1_{U_i \leq F(x)}$$

et donc

$$d_n = \sqrt{n} \sup_x |n^{-1} \sum_1^n 1_{U_i \leq F(x)} - F(x)| = \sqrt{n} \sup_u |n^{-1} \sum_1^n 1_{U_i \leq u} - u|$$

dont la loi est fixe.

La statistique de Kuiper est parfois préférée :

$$d'_n = \sqrt{n} \left(\sup_x (F_n(x) - F(x)) - \min_x (F_n(x) - F(x)) \right).$$

2.1.2 Test du χ^2

Ce test est utile si l'on cherche à simuler des variables aléatoires pour lesquelles le test de Kolmogorov est inadapté (à valeurs non-réelles, discrètes,...).

On découpe l'espace de valeurs de X en ν parties différentes, S_1, \dots, S_ν , de probabilité p_1, \dots, p_ν . On considère alors les probabilités empiriques de tomber sur ces ensembles

$$\hat{p}_k = n^{-1} \sum_{i=1}^n 1_{X_i \in S_k}$$

et l'on considère la statistique de test

$$T_n = n \sum_{k=1}^{\nu} \frac{(\hat{p}_k - p_k)^2}{p_k}.$$

Théorème 6 *Si les X_i sont i.i.d de loi donnée par F alors*

$$T_n \longrightarrow \chi_{\nu-1}^2 \text{ en loi.}$$

Donc le test qui décide que la loi est significativement non-conforme si $T_n > x$, est de niveau asymptotique $1 - F_{\nu-1}(x)$ (probabilité de confiance de $F_{\nu-1}(x)$) où F_ν est la fonction de répartition du χ_ν^2 . Les ν ensembles seront typiquement choisis de probabilité égale.

2.2 Épreuves d'indépendance

On supposera ici souvent pour simplifier que la distribution à générer est $\mathcal{U}([0, 1])$ et la suite sera notée U_n au lieu de X_n . Ces épreuves se généralisent aux autres distributions sans difficulté.

On va voir qu'une bonne partie des épreuves d'indépendance consistera à vérifier que la suite $(U_{j+1}, \dots, U_{j+d})_j$ a bien une distribution uniforme sur l'hypercube $[0, 1]^d$.

Notations. On posera pour tous entiers d et K fixés

$$V_j = (U_{jd+1}, \dots, U_{jd+d}), \quad Y_i = [KU_i], \quad Z_i = (Y_{id+1}, \dots, Y_{id+d}). \quad (2.1)$$

2.2.1 Test des corrélations

Cette épreuve a la particularité de ne pas nécessiter la connaissance de F_X .

On calcule simplement les corrélations empiriques de la suite X_i produite. Ce test peut s'appuyer sur le théorème suivant

Théorème 7 *Si la suite X_i est i.i.d de variance finie, alors pour tout $k > 0$, la corrélation empirique \hat{r}_n entre deux échantillons à distance k , basée sur n échantillons, satisfait*

$$\sqrt{n} \hat{r}_n \longrightarrow \mathcal{N}(0, 1).$$

Il s'ensuit que le test qui décide la dépendance si $\sqrt{n} |\hat{r}_n| > x$ a un niveau égal à $2(1 - F_g(x))$ où F_g est la fonction de répartition de la gaussienne (probabilité de confiance $2F_g(x) - 1$).

2.2.2 Test des séries

Les Z_i , $i = 1, \dots, n$ ne peuvent prendre que $q = K^d$ valeurs possibles distinctes.

- Méthode χ^2 : Ici $n \gg q$ (p.ex $d = 5$, $K = 2$) et l'on fait un test du χ^2 .
- Méthode des collisions : q est grand (par exemple de l'ordre de la période) et l'on compte $C = n - n_v$ où n_v est le nombre de valeurs prises parmi les q valeurs possibles ; C est donc le nombre de répétitions (collisions). On a sous hypothèse d'uniformité [29]

$$P(C = c) = \frac{q!}{q^n (q - n + c)!} C_n^c$$

qui permet de construire un intervalle de confiance pour C (numériquement parlant, il faut un algorithme spécial) ; il sera de la forme $[0, a]$ ou $[b, +\infty[$ selon qu'on s'attend à observer un fort nombre de collisions (concentrations de points dans certaines régions) ou un faible nombre de collisions (générateur trop régulier), ou $[a, b]$ pour détecter les deux types d'anomalie. On vérifie qu'il faut dans le deuxième cas $n \gg \sqrt{q}$ car $P(C = 0) \simeq \exp(-n^2/q)$.

2.2.3 Test des points proches

Soient V_1, \dots, V_n des variables $\mathcal{U}([0, 1]^d)$ et D la distance minimale entre deux de ces points (on prend la distance sur le tore, c'est-à-dire que les différences sont calculées modulo 1, comprises entre $-1/2$ et $1/2$). Soit $v = (\pi)^{d/2} / \Gamma(d/2 + 1)$ le volume de la sphère unité en dimension d , alors la variable

$$W = \exp(-\lambda D^d), \quad \lambda = \frac{n(n-1)}{2} v$$

suit approximativement une loi $\mathcal{U}([0, 1])$.

On peut donc simuler de telles suites de n vecteurs et tester si la loi de W est bien uniforme. Pour les détails de mise-en-œuvre, voir [32].

2.2.4 Birthday spacings

Prendre la suite Y_1, \dots, Y_n , et considérer la suite réordonnée $Y_{(i)}$, puis les écarts $S_i = Y_{(i)} - Y_{(i-1)}$, $i = 1, \dots, n$ ($Y_{(0)} = 0$). On calcule alors le nombre C de collisions de la suite (S_i) , c-à-d n moins le nombre de valeurs prises.

Sous l'hypothèse d'uniformité, C possède une certaine loi que l'on peut calculer [29]. Il ne reste alors qu'à répéter l'expérience et à faire un test d'adéquation de distribution. Typiquement K sera la période mais ce n'est pas obligé.

2.2.5 Test des lacunes (gap test)

Soit $I = [a, b] \subset [0, 1]$ un intervalle, typiquement $I = [0, 1/2]$. On dit que la suite U_i admet une lacune de longueur l après j si :

$$U_j \in I, \quad U_{j+1}, \dots, U_{j+l} \notin I, \quad U_{j+l+1} \in I$$

Les longueurs successives L_k de ces lacunes sont des variables i.i.d dont la loi est $p_l = p(1-p)^l$ où $p = b - a$. Il ne reste plus qu'à faire un test d'adéquation de distribution par exemple un test de Kolmogorov (exercice 2.3.2), ou un χ^2_ν sur la variable $\tilde{L}_k = \min(L_k, \nu)$.

2.3 Exercices

Exercice 2.3.1 (Épreuve du maximum) On choisit d et l'on pose $Y_j = \max(X_{jd+1}, \dots, X_{j(d+d)})$. Expliciter le test de Kolmogorov.

Exercice 2.3.2 On reprend le test de Kolmogorov. Utiliser les résultats de l'exercice 1.5.16 pour montrer que si F est discontinue :

$$P(d_n > \lambda) \leq P(d_n^c > \lambda)$$

où d_n^c est la statistique dans le cas continu. En déduire qu'à seuil fixe, le test qui réfute F si $d_n > \lambda$ a un niveau supérieur si F est discontinue.

Exercice 2.3.3 (Points proches) On reprend l'idée du § 2.2.3 mais en considérant la variable δ

$$\delta = \min_{1 \leq i \leq n} \{\|V_i - V_0\|_\infty, i = 1, \dots, n\}, \quad (\|x\|_\infty = \max_k |x_k|)$$

où les différences sont calculées modulo 1 de sorte que $0 \leq V_i - V_0 < 1$.

1. En remarquant que $P(\delta > h | V_0 = v)$ est indépendant de v , calculer la loi de δ .
2. On réalise P échantillons de δ . Quelle est la statistique du test de Kolmogorov et quel niveau de signification obtient-on ?
3. Pourquoi peut-il sembler raisonnable de choisir $d \simeq \log(n)$?

Exercice 2.3.4 (Programmation) On considère la suite

$$U_n = a(U_{n-1}U_{n-2})^{U_{n-3}} \pmod{1}$$

1. Programmer cette suite avec $a = 2$ et tracer sa fonction de répartition empirique.
2. Montrer, en utilisant les corrélations d'ordre 1, que c'est une mauvaise suite aléatoire.
3. Ressayer avec $a = 100000$ (ou d'autres valeurs de votre choix).

Chapitre 3

Monte Carlo : Exemples de base

3.1 Estimation du volume

On cherche à estimer le volume V d'un objet géométrique E inclus dans $[0, 1]^d$. On suppose que l'on a un moyen simple de tester si un point x appartient à E ou non.

ALGORITHME

1. Tirer n points dans $[0, 1]^d$
2. Compter le nombre N de points appartenant à E
3. $\hat{V} = N/n$
4. $\hat{\sigma}(\hat{V})^2 = \hat{V}(1 - \hat{V})/n$
5. $I_c = \hat{V} \pm \hat{\sigma} Q_g(1 - \alpha/2)$

Q_g désigne la fonction quantile de la gaussienne centrée réduite. N est une somme de variables de Bernoulli d'espérance V , la variance de \hat{V} vaut donc $\sigma^2 = V(1-V)/n$, et $\hat{\sigma}(\hat{V})^2$ en est une estimation raisonnable. I_c est un intervalle de confiance de niveau asymptotique α , basé sur l'approximation gaussienne (théorème-limite central $\hat{V} \simeq V + \sigma\mathcal{N}(0, 1)$).

Si l'on sait que E est compris entre deux ensembles E_- et E_+ , $E_- \subset E \subset E_+$, de volume V_- et V_+ , on a toujours intérêt, si c'est possible et peu coûteux, à tirer uniformément dans l'ensemble $E_+ \setminus E_-$; la proportion de points tombant dans E donnera une estimation \hat{R} de $R = (V - V_-)/(V_+ - V_-)$ avec variance $R(1 - R)/n$, d'où une variance $(V - V_-)(V_+ - V)/n$ pour \hat{V} .

3.2 Intégration

Au paragraphe précédent on calculait $E[1_E(X)]$ où X est $\mathcal{U}([0, 1]^d)$. La problématique est ici sensiblement la même puisqu'il s'agit de calculer $\theta = E[f(X)]$ pour une v.a. X que l'on sait simuler. L'estimateur naturel est

$$\hat{\theta} = \frac{1}{n} \sum_{i=1}^n f(X_i).$$

Sa variance vaut

$$Var(\hat{\theta}) = \frac{1}{n^2} Var(f(X)).$$

Comme précédemment, cette variance permet d'obtenir, par approximation gaussienne, un intervalle de confiance basé sur

$$\hat{\sigma}(\hat{\theta})^2 = \frac{1}{n^2} \sum_{i=1}^n (f(X_i) - \hat{\theta})^2.$$

Cette méthode permet en particulier de calculer des intégrales sur $[0, 1]^d$, et par extension sur toute partie de \mathbb{R}^d .

Noter la vitesse de convergence en $1/\sqrt{n}$. Les méthodes (déterministes) d'intégration par quadrature, c-à-d de la forme

$$\int f(x) dx \simeq \sum_{i=1}^n w_i f(x_i)$$

commettent une erreur en $O(n^{-s/d})$ où s est la régularité de la fonction (les dérivées d'ordre inférieur ou égal à s sont bornées) ; les suites (x_i, w_i) dépendent du s que l'on considère. On voit donc qu'en grande dimension, l'algorithme de Monte Carlo est compétitif.

3.3 Perte de connexion dans un graphe

On se donne un graphe totalement connecté (réseau de communication) pour lequel la probabilité de perte du i -ième arc est p_i . On veut savoir la probabilité p de coupure de connexion entre deux nœuds s et t (ou deux ensembles de nœuds).

La solution la plus simple est de répéter n fois l'expérience suivante (tirage d'un graphe aléatoire) :

1. Tirer au hasard les pertes d'arc : pour chaque arc tirer une variable Bernoulli de probabilité p_i dont la valeur dira si l'arc est perdu.
2. Voir si la communication entre s et t est perdue.

et enfin de calculer la probabilité empirique \hat{p}_n de coupure de communication entre ces deux nœuds. La variance d'estimation est $p(1-p)/n$.

En pratique on essaiera de tester la perte de communication au fur et à mesure de la génération des arcs : par exemple si deux arcs partent de s , on peut les simuler en premier et s'ils sont perdus, rien ne sert de poursuivre la simulation du graphe.

Pour des méthodes plus subtiles, voir [14].

3.4 Décompte

Soit E_1, \dots, E_K une suite de sous-ensembles d'un ensemble E ; il s'agit ici de calculer le cardinal de leur réunion ou de leur intersection. La méthode déterministe consiste en général à utiliser une formule

$$|\cup E_i| = \sum_i |E_i| - \sum_{i \neq j} |E_i \cap E_j| + \dots$$

Cette formule est très vite inutilisable. On va proposer des méthodes de Monte Carlo pour approximer de telles quantités. Par défaut, tous les tirages seront faits uniformément dans les ensembles concernés.

3.4.1 Premier cas

On est ici dans la situation où l'on peut tirer au hasard des éléments dans E et que l'on peut tester à bas prix leur appartenance aux différents sous-ensembles.

Exemple. On se donne un système d'équations de la forme $Ax \leq b$, où A est une matrice $m \times p$, x est un vecteur de 0 ou 1 de dimension p , et b est un vecteur de dimension m . Cette équation signifie une inégalité pour chaque coordonnée. Le calcul du nombre de solutions x à A et b fixé est très difficile. On est ici dans la situation où

$$E = \{0, 1\}^p, \quad E_i = \{x \in E : \sum_j a_{ij}x_j \leq b_i\}$$

et l'on cherche $\theta = |\cap_i E_i|$ ($|F|$ désigne le cardinal de F).

Comme $\theta/|E|$ est la probabilité qu'un point de E tiré au hasard tombe dans $\cap_i E_i$, l'algorithme consiste à calculer empiriquement cette probabilité et à la multiplier par $|E|$:

ALGORITHME

1. Tirer au hasard $e_1, \dots, e_n \in E$
2. $\hat{\theta} = |E| \cdot |\{j : e_j \in \cap_i E_i\}|/n$, $\hat{\sigma}(\hat{\theta})^2 = (|E| - \hat{\theta})\hat{\theta}/n$.

La variance de $\hat{\theta}$ est $(|E| - \theta)\theta/n$. Pour la réunion l'algorithme est bien entendu similaire, puisque le complémentaire de la réunion est l'intersection des complémentaires.

3.4.2 Deuxième cas

On est ici dans la situation où l'on peut tirer au hasard des éléments dans chacun des E_i et où les $|E_i|$ sont connus. Karp et Luby [27] ont proposé un algorithme pour calculer $\theta = |\cup_i E_i|$ dans ces circonstances.

ALGORITHME

1. Tirer au hasard $\alpha \in \{1, \dots, K\}$ avec probabilité $P(\alpha = i) = p_i = |E_i|/\sum |E_j|$, puis tirer $\varepsilon \in E_\alpha$
2. Tirer au hasard des $\beta \in \{1, \dots, K\}$ jusqu'à ce que $\varepsilon \in E_\beta$ et enregistrer le nombre τ de tirages ($\tau = 1$ si $\varepsilon \in E_\beta$ le premier coup)
Autre option : $\tau = K/l(\varepsilon)$ où la fonction $l(e)$ est le nombre de i tels que $e \in E_i$.
3. Itérer N_{it} fois les deux points précédents. $\hat{\theta} = \bar{\tau} \sum |E_i|/K$ où $\bar{\tau}$ est la moyenne empirique de τ
4. $\hat{\sigma}(\hat{\theta})^2 = N_{it}^{-1} K^{-2} (\sum |E_i|)^2 Var_{emp}(\tau)$ (Var_{emp} =variance empirique).

Remarques. Dans cet énoncé, on a choisi de désigner les points et indices aléatoires par des lettres grecques afin de les distinguer des variables muettes.

On va voir que la variance est bornée par

$$Var(\hat{\theta}) \leq 2N_{it}^{-1}\theta \sum |E_i|.$$

Le calcul montre que la seconde option du point 2 fait gagner un facteur 2 dans cette borne.

Cette borne ne prend pas en compte le nombre effectif de tirages effectués, ce qui la rend difficile à comparer à celle du paragraphe précédent. Ce dernier est la variable aléatoire $N_{it}\bar{\tau}$ qui est asymptotiquement équivalente à $n = \theta K N_{it} / \sum |E_i|$. On a donc

$$\text{Var}(\hat{\theta}) \leq \frac{2K}{n} \theta^2$$

nettement inférieure ($|E| - \theta)/n$ si $\theta \ll |E|$.

DÉMONSTRATION: Soit F_q l'ensemble des points qui appartiennent à exactement q des ensembles E_i :

$$F_q = \{e \in E : l(e) = q\}.$$

On a alors

$$\begin{aligned} P(\varepsilon \in F_q) &= \sum_j P(\varepsilon \in F_q | \alpha = j) p_j \\ &= \sum_j \frac{|F_q \cap E_j|}{|E_j|} \frac{|E_j|}{\sum |E_k|} \\ &= \frac{1}{\sum |E_k|} \sum_j \sum_x 1_{x \in F_q \cap E_j} \\ &= \frac{1}{\sum |E_k|} \sum_x \left(\sum_j 1_{x \in E_j} \right) 1_{x \in F_q} \\ &= \frac{q|F_q|}{\sum |E_k|}. \end{aligned}$$

D'où le calcul de l'espérance de τ :

$$E[\tau] = \sum_e E[\tau | \varepsilon = e] P(\varepsilon = e)$$

mais comme $P(\tau = k | \varepsilon) = (1 - p)^{k-1} p$, $p = l(\varepsilon)/K$, il vient

$$E[\tau | \varepsilon] = \frac{K}{l(\varepsilon)}$$

d'où finalement

$$E[\tau] = \sum_e \frac{K}{l(e)} P(\varepsilon = e) = \sum_q \frac{K}{q} P(\varepsilon \in F_q) = \frac{K|\cup F_q|}{\sum |E_k|} = \frac{K|\cup E_i|}{\sum |E_k|}.$$

Pour la variance de $\hat{\theta}$, on a (en notant N_{it} le nombre d'itérations de l'algorithme)

$$\begin{aligned} \text{Var}(\hat{\theta}) &= K^{-2} \left(\sum |E_i| \right)^2 \text{Var}(\bar{\tau}) \\ &= N_{it}^{-1} K^{-2} \left(\sum |E_i| \right)^2 \text{Var}(\tau) \\ &\leq N_{it}^{-1} K^{-2} \left(\sum |E_i| \right)^2 E[\tau^2]. \end{aligned}$$

Le calcul de $E[\tau^2]$ nécessite de remarquer que $E[\tau^2 | \varepsilon] = 2p^{-2} - p^{-1}$, $p = l(\varepsilon)/K$; on a donc $E[\tau^2 | \varepsilon] \leq 2K^2 l(\varepsilon)^{-2}$. On en déduit comme précédemment la borne (un peu grossière)

$$E[\tau^2] \leq \sum_q 2K^2 q^{-2} \frac{q|F_q|}{\sum |E_k|} \leq \frac{2K^2 |\cup E_i|}{\sum |E_i|}$$

qui conduit bien à la borne annoncée pour la variance. ■

Exemple : formules booléennes. On se place dans $\{0, 1\}^d$. Chaque E_i est l'ensemble des suites de variables booléennes $b \in \{0, 1\}^d$ qui satisfont une formule conjonctive (des “et”) :

$$\prod_{j \in J_i} b_j = 1$$

soit encore en langage logique

$$\bigwedge_{j \in J_i} b_j = \text{vrai.}$$

E_i est donc simplement caractérisé par l'ensemble d'indices J_i . L'ensemble $\cup E_i$ est une disjonction (des “ou”) de telles formules :

$$\bigvee_i \bigwedge_{j \in J_i} b_j = \text{vrai}$$

ce qui correspond à la “forme normale disjonctive” d'une formule booléenne.

Noter qu'il n'y a aucune difficulté à tirer un point au hasard dans E_i et que $|E_i| = 2^{d-|J_i|}$.

Extension au cas probabilisé. On vérifie sans peine que si l'on veut maintenant calculer $P(\cup E_i)$ pour une certaine mesure de probabilité P l'algorithme devient :

ALGORITHME

1. Tirer au hasard $\alpha \in \{1, \dots, K\}$ avec probabilité $P(\alpha = i) = p_i = P_0(E_i) / \sum P_0(E_j)$, puis tirer $\varepsilon \in E_\alpha$ avec probabilité $P(\varepsilon = e) = P_0(e) / P_0(E_\alpha)$
2. (inchangé) Tirer au hasard des $\beta \in \{1, \dots, K\}$ jusqu'à ce que $\varepsilon \in E_\beta$ et enregistrer le nombre $\tau \geq 1$ de tirages (ou faire $\tau = K/l(\varepsilon)$)
3. Itérer les deux points précédents. L'estimateur est $\bar{\tau} \sum P_0(E_i)$ où $\bar{\tau}$ est la moyenne empirique de τ .

3.5 Exemples applicatifs simples

3.5.1 Temps d'échappement d'une comète du système solaire

Soit $-z < 0$ l'énergie d'une comète (l'énergie est négative, calculée de sorte qu'elle sera nulle une fois la comète sortie du système solaire). La durée d'une période de sa trajectoire est $z^{-3/2}$. On suppose que cette énergie varie aléatoirement selon un modèle où elle est considérée constante sur chaque période :

$$z_{i+1} = z_i + u_i.$$

Les u_i sont des gaussiennes centrées réduites indépendantes. Soit M le premier instant où $z_M = 0$, la durée de vie de cette comète est

$$T = \sum_{i=0}^{M-1} z_i^{-3/2}.$$

Des simulations de Monte Carlo permettent d'étudier la distribution de T [21].

3.5.2 Files d'attente

Une station service est ouverte h heures par jours. Il y a un seul pompiste et les clients font la queue pour se faire servir dès que leur tour arrive. Les inter-arrivées des clients sont des variables aléatoires exponentielles indépendantes de paramètre λ_a . Le temps de service d'un client suit une loi d'Erlang(2, λ_s) (somme de 2 exponentielles indépendantes de paramètre λ_s).

Les clients suivent la stratégie suivante à leur arrivée en fonction de la longueur l de la file d'attente

- rester faire la queue si $l < 5$
- partir si $l > 10$
- si $5 \leq l \leq 10$ il y a une probabilité $(11 - l)/7$ que le client reste.

Un client achète pour P francs d'essence, où P suit une loi inverse gaussienne (cf exercice 1.5.18).

Estimer le chiffre d'affaire moyen d'une journée, et le temps d'attente moyen d'un client.

Pour ne pas perdre de temps, s'il n'y a aucun client, le pompiste se lance dans une tâche annexe qui lui prend un temps exponentiel de paramètre λ_t à réaliser, qui rapporte une somme S de loi inverse gaussienne. Quel est maintenant le chiffre d'affaire moyen d'une journée?

3.5.3 Options

On se donne le modèle suivant pour l'évolution de la valeur d'une action :

$$V_n \stackrel{\text{def}}{=} V_0 e^{\sum_{i=1}^n X_i} \sim \mathcal{N}(\mu, \sigma^2).$$

On achète au temps 0 une option, c'est-à-dire le droit d'acheter une action à un prix K fixé à l'avance et à une date inférieure à N . On cherche alors la meilleure politique d'achat, en supposant que l'on revend immédiatement après l'achat : comment décider au vu du cours du moment le meilleur instant ν pour acheter ; bien entendu on n'achète que si $V_\nu > K$, et le bénéfice est donc de $(V_\nu - K)_+$. Mathématiquement, comme on ne peut prendre de décision que sur la base de l'information passée, ν est un temps d'arrêt, c'est-à-dire que l'événement $\{\nu = n\}$ ne dépend que de l'information précédant n ; cet événement est donc ici une fonction des variables V_1, \dots, V_n . On cherche donc à maximiser sur les stratégies (temps d'arrêts) ν le bénéfice moyen

$$E[(V_\nu - K)_+].$$

Une première stratégie consiste à prendre $\nu = N$. On montre que c'est la meilleure si $\mu + \sigma^2/2 > 0$.

Si $\mu + \sigma^2/2 < 0$, c'est un problème très difficile. Une stratégie proposée [51] est d'acheter au premier temps ν tel que le gain est supérieur à ce que l'on peut espérer avoir dans le futur, soit le premier ν tel que

$$V_\nu - K \geq E[(V_{\nu+i} - K)_+ | V_\nu], \quad i = 1, \dots, N - \nu.$$

Le membre de gauche se calcule sans problème et l'on obtient

$$V_\nu - K \geq V_\nu e^{i(\mu + \sigma^2/2)} \Phi(\sigma\sqrt{i} + b_i) - K \Phi(b_i), \quad b_i = \frac{i\mu - \log(K/V_\nu)}{\sigma\sqrt{i}}$$

où Φ est la fonction de répartition de la gaussienne.

Des simulations permettent de comparer différentes stratégies, sur des critères arbitraires (gain moyen, variance du gain...).

3.6 Exercices

Exercice 3.6.1 On sait que E contient un ensemble E_- de volume V_- . On sait tester si un point appartient à E_- mais en revanche on n'est pas capable de tirer uniformément dans le complémentaire de E_- . On choisit donc d'estimer le volume de $E \setminus E_-$ par la méthode de base, pour en déduire ensuite le volume de E .

À quelle condition cette méthode est-elle plus rapide que celle consistant à estimer le volume de E directement ? Proposer également une condition suffisante simple sur le volume de E .

Exercice 3.6.2 On est dans la situation de l'exercice 3.6.1 avec $E_- = \{x \in [0, 1]^d : 0.1 \leq x_1 \leq 0.9\}$. Comparer les performances de l'algorithme de base, de celui de l'exercice 3.6.1, et de l'algorithme accéléré du cours. On explicitera comment on fait le tirage pour l'algorithme accéléré.

Exercice 3.6.3 Comparer $\sigma(\hat{V}_n)$ et l'erreur relative $\sigma(\hat{V}_n)/V$.

Exercice 3.6.4 On considère la méthode suivante pour calculer $\theta = \int_0^1 f(x) dx$:

1. Tirer X_i selon $\mathcal{U}([\frac{i-1}{n}, \frac{i}{n}])$, $i = 1, \dots, n$
2. $\hat{\theta} = \frac{1}{n} \sum_i f(X_i)$.

Montrer que l'erreur d'estimation est d'ordre $n^{-3/2}$ si f est deux fois dérivable (on utilisera la formule de Taylor pour calculer $\sigma(\hat{\theta})$).

Exercice 3.6.5 (Programmation) Estimer la proportion de suites de $\{0, 1\}^{20}$ qui contiennent au moins une suite de quatre 1 consécutifs. On utilisera la méthode de Karp et Luby.

Exercice 3.6.6 (Programmation) Estimer π avec trois décimales.

Exercice 3.6.7 (Programmation) Tester la méthode d'intégration sur $f(x) = \prod_{i=1}^d \frac{k\pi}{2} \sin(k\pi x_i)$ avec par exemple $d = 8$ et différents k [57]. On calculera la valeur exacte de l'intégrale et de $\sigma(\hat{\theta})$. Comparer $k = 1$ et $k = 2$.

Exercice 3.6.8 (Programmation) On se propose de simuler un polytope convexe en dimension d puis de calculer le volume de son intersection avec le cube unité.

1. Générer $d + 1$ vecteurs gaussiens Z_i et poser $a_i = Z_i / \|Z_i\|$. Le i -ième demi-espace a pour équation

$$\langle a_i, x - c \rangle \leq 1/2$$

où c est le centre du cube. Quelle est la distance de c au plan frontière (c-à-d d'équation $\langle a_i, x - c \rangle = 1/2$) ? Le convexe sera choisi comme l'intersection de ces demi-espaces.

2. Calculer le volume et l'écart-type de l'estimée.

Chapitre 4

Monte Carlo : réduction de variance

Les méthodes de Monte Carlo sont l'application de la simulation à l'estimation. À partir du principe de base généralement très simple (simuler et moyenner), les chercheurs ont imaginé de nombreuses techniques pour accélérer la vitesse de convergence, qui restera toutefois quasiment toujours en $1/\sqrt{n}$. Ces méthodes font l'objet de ce chapitre.

Notons que l'usage de ces méthodes représente souvent un surcoût en calcul qu'il faudrait prendre en compte dans l'étude des performances. Pour un algorithme donné, ce surcoût varie énormément d'une situation concrète à une autre, c'est pourquoi on ne prendra en compte essentiellement dans la suite que la diminution de variance d'estimation, ce qui n'est pas forcément très objectif.

4.1 Échantillonnage préférentiel (importance sampling)

C'est une méthode de calcul de $\theta = E[f(X)]$ qui consiste à tirer les variables selon une distribution erronée, et à compenser numériquement le résultat à postériori.

Pour en donner une idée, revenons au calcul d'une intégrale sur $[0, 1]$. Si p est une densité sur $[0, 1]$ (fonction positive d'intégrale 1), on peut écrire

$$\int_0^1 f(x)dx = \int_0^1 \frac{f(x)}{p(x)}p(x)dx$$

ce qui suggère la possibilité d'un autre algorithme :

$$\hat{\theta}' = \frac{1}{n} \sum_{i=1}^n \frac{f(Y_i)}{p(Y_i)}$$

où les Y_i sont i.i.d sur $[0, 1]$ de loi $p(x)dx$. De manière plus générale on peut proposer :

ALGORITHME Estimation de $\theta = E[f(X)]$.

1. Tirer n points indépendants sur Y_i selon la loi avec densité p par rapport à la loi de X
2. $\hat{\theta} = \frac{1}{n} \sum_{i=1}^n \frac{f(Y_n)}{p(Y_n)}$ (convention $\frac{0}{0} = 0$)
3. Variance estimée : $\hat{\sigma}(\hat{\theta})^2 = \frac{1}{n^2} \sum_{i=1}^n \left(\frac{f(Y_n)^2}{p(Y_n)^2} - \hat{\theta}^2 \right)$

Rappelons que si X a une densité $f_X(x)$ et Y une densité f_Y , alors $p = f_Y/f_X$.

Théorème 8 $\hat{\theta}$ est non-biaisé avec pour variance :

$$n\text{Var}(\hat{\theta}) = E[f(X)^2 p(X)^{-1}] - E[f(X)]^2$$

Elle est minimale pour $p = p^*$, $p^*(y) = |f(y)|/E[|f(X)|]$ auquel cas elle vaut 0 si $f \geq 0$.

DÉMONSTRATION: Pour toute fonction φ , on a : $E[\varphi(Y)] = E[\varphi(X)p(X)]$. Par conséquent

$$E[\hat{\theta}] = E\left[\frac{f(Y)}{p(Y)}\right] = E[f(X)] = \theta.$$

Pour la variance :

$$n\text{Var}(\hat{\theta}) = \text{Var}(f(Y)/p(Y)) = E[f(Y)^2/p(Y)^2] - \theta^2 = E[f(X)^2/p(X)] - \theta^2.$$

Pour trouver le meilleur p , il faut minimiser $E[f(X)^2/p(X)]$. Noter que si $p = p^*$ cette quantité vaut $E[|f(X)|]^2$ qui est la plus petite valeur possible car pour tout choix de p

$$E[|f(X)|]^2 = E\left[\frac{|f(X)|}{\sqrt{p(X)}}\sqrt{p(X)}\right]^2 \leq E\left[\frac{f(X)^2}{p(X)}\right] E[p(X)] = E[f(X)^2/p(X)].$$

■

Aspects pratiques. Comme p^* est inconnu (si $f \geq 0$, connaître p^* implique connaître θ), on ne pourra pas employer le meilleur algorithme, toutefois il sera bon d'essayer de s'en approcher, ce qui signifie de **tirer les points dans une région en quantité proportionnelle à la valeur de f** . Cette méthode peut être très avantageuse en grande dimension si f est petite sur une bonne partie de l'espace.

Événements rares. C'est l'exemple typique d'application de la méthode : on cherche à estimer $P(A)$ pour un événement A de probabilité très faible. Si l'on utilise la méthode habituelle, il faudra tirer un très grand nombre d'échantillons pour tomber dans A de temps en temps. L'échantillonnage préférentiel conseille de tirer avec densité proche de $p^*(y) = 1_A/P(A)$, c'est-à-dire de la loi de X conditionnelle à $X \in A$. On tire donc selon une loi qui favorise A davantage.

Exemple. Reprenons l'exemple du § 3.5.2 et supposons que l'on cherche la probabilité d'avoir trois fois plus de clients servis que d'habitude, sur une journée : $\theta = P(N_s \geq 3\mu)$ où $\mu = E[N_s]$. On suppose ici que le temps de service est également exponentiel. Il s'agit de deviner la loi des variables sachant que $N_s \geq 3\mu$. Il s'agit de deviner la loi du processus sachant $N_s \geq 3\mu$. Plusieurs cas peuvent se présenter :

- On pense que le pompiste est sous-employé et qu'une grosse arrivée de clients suffit à expliquer le phénomène. Dans ce cas on pourra simuler les arrivées avec un taux trois fois supérieur et choisir l'estimateur

$$\hat{\theta} = \text{Moyenne empirique}(1_{N_s \geq 3\mu} 3^{-N_t} e^{\sum 2\lambda_a T_i})$$

où les T_i sont les inter-arrivées entre deux clients, N_s le nombre de clients servis dans la simulation, N_t le nombre total de clients apparus. Le facteur se trouve par le raisonnement suivant : on a simulé avec la densité $3\lambda_a e^{-3\lambda_a T_i}$ pour chaque inter-arrivée au lieu de $\lambda_a e^{-\lambda_a T_i}$, il faut donc compenser en multipliant par $e^{2\lambda_a T_i}/3$.

Noter que θ est maintenant rendu petit par le terme $3^{-N_t} e^{\sum 2\lambda_a T_i} \sim 3^{-N_t} e^{2N_t/3}$.

Le choix de tripler le taux a été fait par simplicité, on aurait pu faire un autre choix de loi d'arrivée qui en gros triple le nombre de clients.

- On pense qu'au contraire les clients arrivent toujours en grand nombre et qu'un jour exceptionnel s'expliquera par de temps de services plus courts, auquel cas on simulera avec un taux trois fois supérieur puis

$$\hat{\theta} = \text{Moyenne empirique}(1_{N_s \geq 3\mu} 3^{-N_s} e^{\sum 2\lambda_s S_i})$$
où les S_i sont les temps de service.
- On pense que les deux effets interviennent et l'on combine les deux méthodes, par exemple en multipliant par trois les deux taux.

4.2 Variables de contrôle

Supposons que l'on cherche à calculer $\theta = E[f(X)]$ par simulations de X . Soit Y_i une v.a. a priori proche de $f(X_i)$ dont on connaît l'espérance (typiquement $Y_i = g(X_i)$ avec g proche de f), on a alors l'estimateur

$$\hat{\theta} = E[Y] + \frac{1}{n} \sum_{i=1}^n f(X_i) - Y_i.$$

Il a une variance de

$$n \text{Var}(\hat{\theta}) = \text{Var}(f(X)) + \text{Var}(Y) - 2 \text{Cov}(f(X), Y)$$

et l'on y gagne si

$$\text{Cov}(f(X), Y) \geq \frac{1}{2} \text{Var}(Y).$$

Exemple : retranchement d'un estimateur intermédiaire. Ayant à notre disposition une famille paramétrique de fonctions g_β pour laquelle on sait calculer $E[g_\beta(X)]$, on peut de choisir pour g un bon approximant de f se trouvant dans cette famille (par exemple en estimant β).

Le cas non-paramétrique est similaire. Si f est régulière, l'estimation de $\int_{[0,1]^d} f(x) dx$ par $n^{-1} \sum f(X_i)$ peut sembler rustique. Une façon d'améliorer est de commencer par faire une estimation non paramétrique de f à l'aide de points Z_1, \dots, Z_p

$$g(x) = \frac{1}{p^d h} \sum_{i=1}^p f(Z_i) K\left(\frac{x - Z_i}{h}\right)$$

avec, si s est la régularité de f , une erreur d'approximation dans L_2 de $p^{-s/(2s+d)}$ pour h d'ordre $p^{-1/(2s+d)}$. Z et g sont désormais fixés. On estime alors l'intégrale de $f - g$ avec $n - p$ points et l'on obtient alors pour $\hat{\theta}$ une variance de

$$\text{Var}(\hat{\theta}) = (n - p)^{-1} p^{-2s/(2s+d)} O(1) = n^{-1-2s/(2s+d)} O(1)$$

si $p = n/2$. On n'a pas pris en compte ici le fait que le temps de calcul de $f(x)$ n'est pas forcément analogue au temps de calcul de $g(x)$. De plus le support de g dépasse $[0, 1]^d$, il faut donc calculer l'intégrale de $f - g$ sur un cube un peu plus grand.

4.3 Échantillonnage corrélé

On cherche à estimer la différence $\theta = \theta_2 - \theta_1$ deux paramètres $\theta_1 = E[f(X)]$ et $\theta_2 = E[g(Y)]$; c'est-à-dire la moyenne de deux fonctions sous deux distributions différentes. Souvent, $f = g$ et les distributions sont proches. La méthode usuelle consiste à générer des X_i et des Y_i indépendants et à poser

$$\hat{\theta} = \hat{\theta}_1 - \hat{\theta}_2 = \frac{1}{n} \sum_{i=1}^n f(X_i) - g(Y_i)$$

et l'on trouve

$$\text{Var}(\hat{\theta}) = \text{Var}(\hat{\theta}_1) + \text{Var}(\hat{\theta}_2) - 2 \text{Cov}(\hat{\theta}_1, \hat{\theta}_2) = \text{Var}(\hat{\theta}_1) + \text{Var}(\hat{\theta}_2).$$

On aurait bien entendu un meilleur résultat si l'on arrivait à corrélérer positivement les deux estimateurs. Par exemple dans le cas extrême où X et Y ont même loi et $f = g$, le choix $\hat{\theta}_2 = \hat{\theta}_1$ conduirait à une variance nulle ; on peut donc penser à juste titre que le gain peut être considérable.

Si l'on utilise la méthode d'inversion (cf § 1.3.2), une possibilité est de générer X_i et Y_i à partir de la même variable uniforme : $X_i = Q_X(U_i)$ et $Y_i = Q_Y(U_i)$; il suffit alors que f et g soient croissantes l'une de l'autre (par exemple si $f = g$) pour réduire la variance (cf exercice 4.8.8).

4.4 Variables antithétiques

Considérons deux estimateurs $\hat{\theta}_1$ et $\hat{\theta}_2$. Leur moyenne $\hat{\theta}_3$ est encore un estimateur dont la variance est

$$\text{Var}\left(\frac{\hat{\theta}_1 + \hat{\theta}_2}{2}\right) = \frac{\text{Var}(\hat{\theta}_1)}{4} + \frac{\text{Var}(\hat{\theta}_2)}{4} + \frac{\text{Cov}(\hat{\theta}_1, \hat{\theta}_2)}{2}.$$

On voit que si ces deux estimateurs sont négativement corrélés, on fait mieux que s'ils étaient indépendants. On a par exemple

Théorème 9 Soient U_i des v.a.i.i.d $\mathcal{U}([0, 1])$ et φ une fonction monotone de chacune de ses variables, alors les variables

$$X = \frac{1}{2}(\varphi(U_1, \dots, U_n) + \varphi(U_{n+1}, \dots, U_{2n})) \quad Y = \frac{1}{2}(\varphi(U_1, \dots, U_n) + \varphi(1 - U_1, \dots, 1 - U_n))$$

vérifient $\text{Var}(Y) \leq \text{Var}(X)$.

Pour la démonstration voir l'exercice 4.8.6. En particulier, si f une fonction monotone, alors les estimateurs

$$\hat{\theta}_i = \frac{1}{2n} \sum_{i=1}^{2n} f(Q_X(U_i)), \quad \hat{\theta}_a = \frac{1}{2n} \sum_{i=1}^n f(Q_X(U_i)) + f(Q_X(1 - U_i))$$

où Q_X est la fonction quantile de X du § 1.3.2, vérifient $\text{Var}(\hat{\theta}_a) \leq \text{Var}(\hat{\theta}_i)$. Un exemple d'application est donné au § 4.6 (calcul de π).

On peut aussi envisager cette méthode pour des situations plus compliquées lorsque les simulations utilisent la méthode d'inversion, comme par exemple la pompe à essence (cf § 3.5.2) : l'estimateur prend la forme $\hat{\theta} = \varphi(U_1, \dots, U_n)$, où les variables U_i sont $\mathcal{U}([0, 1])$.

Rien ne garantit à coup sûr que $\hat{\theta}_a$ soit meilleur que $\hat{\theta}_i$ si φ n'est pas monotone ; l'on peut toutefois estimer facilement la corrélation entre les deux termes et vérifier son signe. Plus précisément, si $\hat{\theta} = n^{-1} \sum f(X_i)$ et que X_i est fonction de p variables uniformes (U_1, \dots, U_p) , il s'agit de vérifier que la corrélation entre $f(X_i)$ et $f(\tilde{X}_i)$, où \tilde{X}_i est calculé avec $1 - U_1, \dots, 1 - U_p$, est bien négative.

4.5 Échantillonnage stratifié

Exemple introductif. Supposons que les clients de l'exemple du § 3.5.2 arrivent avec des taux qui changent chaque jour. Il y a trois taux possibles $\lambda_{a1}, \lambda_{a2}, \lambda_{a3}$, qui sont aléatoires avec probabilités p_1, p_2, p_3 . On cherche toujours à calculer le chiffre d'affaire moyen sur une journée.

Le mode habituel de simulation consiste à répéter la procédure suivante : simuler d'abord le choix d'un des trois taux, puis simuler la journée.

L'option d'échantillonnage stratifié consiste à faire trois séries de simulations, une pour chaque taux, puis à faire la moyenne des trois résultats avec les poids p_1, p_2, p_3 .

Méthode. On cherche à calculer $\theta = E[f(X)]$ par simulations de X . On suppose que l'on est capable de séparer l'espace de probabilité Ω en J sous-ensembles disjoints Ω_j de probabilité connue :

$$P(\cup_j \Omega_j) = 1, \quad P(\Omega_j \cap \Omega_k) = 0, \quad j \neq k$$

et que l'on peut simuler sur chacun de ces ensembles avec la probabilité $P(\cdot|\Omega_j)$. Tout se base alors sur la formule

$$E[f(X)] = \sum_{j=1}^J P(\Omega_j) E[f(X)|\Omega_j].$$

ALGORITHME Estimation de $\theta = E[f(X)]$.

1. Simuler n_j points $(X_{1j}, \dots, X_{n_j j})$ dans chaque Ω_j avec probabilité $P(\cdot|\Omega_j)$

2. Estimateur :

$$\hat{\theta} = \sum_{j=1}^J \frac{P(\Omega_j)}{n_j} \sum_{i=1}^{n_j} f(X_{ij})$$

3. Variance estimée :

$$\hat{\sigma}(\hat{\theta})^2 = \sum_{j=1}^J \frac{P(\Omega_j)^2 \hat{\sigma}_j^2}{n_j}, \quad \hat{\sigma}_j^2 = \frac{1}{n_j} \sum_{i=1}^{n_j} f(X_{ij})^2 - \left(\frac{1}{n_j} \sum_{i=1}^{n_j} f(X_{ij}) \right)^2$$

La démonstration de la validité de la procédure est laissée en exercice. La variance de $\hat{\theta}$ est

$$\text{Var}(\hat{\theta}) = \sum_{j=1}^J \frac{P(\Omega_j)^2}{n_j} \sigma_j^2, \quad \sigma_j^2 = E[f(X)^2|\Omega_j] - E[f(X)|\Omega_j]^2.$$

Choix des Ω_j . Au vu de la formule précédente, on a intérêt à choisir ses Ω_j de sorte que f y ait la plus petite variance.

Choix des n_j . Si $n_j = nP(\Omega_j)$, on obtient toujours une réduction de variance d'un terme qui est la variance inter-classe :

$$n \text{Var}(\hat{\theta}) = \sum_{j=1}^J P(\Omega_j) \sigma_j^2 = \text{Var}(f(X)) - \sum_{j=1}^J P(\Omega_j) (E[f(X)|\Omega_j] - \theta)^2. \quad (4.1)$$

Cependant le meilleur choix (sous la contrainte $\sum n_j = n$) est

$$n_j^* = nP(\Omega_j)\sigma_j / \sum_k P(\Omega_k)\sigma_k$$

avec une variance de

$$nVar(\hat{\theta}) = \left(\sum_{j=1}^J P(\Omega_j)\sigma_j \right)^2, \quad n_j = n_j^*.$$

On peut estimer également en cours d'algorithme les proportions idéales $\pi_j = n_j^*/n$:

$$\hat{\pi}_j^* = P(\Omega_j)\hat{\sigma}_j / \sum_k P(\Omega_k)\hat{\sigma}_k.$$

Choix de J . Éventuellement grand : on vérifie simplement que pour les deux choix de n_j proposés ci-dessus, la variance diminue à chaque fois que l'on scinde une classe Ω_j en deux. Rien n'interdit de prendre $J = n$.

Extension. Si l'on ne sait pas tirer aléatoirement dans les Ω_j mais que l'on sait tester si $x \in \Omega_j$, on peut alors tirer les X_i aléatoirement dans Ω et prendre ensuite l'estimateur $\hat{\theta}$, en distinguant a posteriori ceux qui sont tombés dans chaque ensemble. Noter que si l'on approxime dans cet estimateur $P(\Omega_j)$ par n_j/n , on retombe sur l'estimateur habituel !

4.6 Conditionnement

Exemple. Considérons la station essence du § 3.5.2. Si l'on s'intéresse au chiffre d'affaire moyen sur une journée, la méthode de base consiste à simuler une réalisation P_k du prix payé pour chaque client, et à calculer la somme des P_k sur chaque journée simulée :

$$\hat{\theta} = \frac{1}{n} \sum_{i=1}^n \sum_{k=1}^{N_i} P_{ik}$$

où N_i est le nombre de client de la i -ième journée. la méthode avec conditionnement consiste à affecter la moyenne \bar{P} de P_k à chaque transaction :

$$\hat{\theta}' = \frac{\bar{P}}{n} \sum_{i=1}^n N_i$$

ce qui est intuitivement meilleur.

De manière générale, lorsqu'on cherche à calculer $E[X]$, on simule des réalisations de X mais aussi d'autres variables annexes, Y, Z, \dots . Il arrive que l'on sache simplement calculer $E[X|Y] = g(Y)$, auquel cas l'estimateur

$$\hat{\theta} = \frac{1}{n} \sum X_i, \quad Var(\hat{\theta}) = \frac{1}{n} Var(X)$$

peut être remplacé par

$$\hat{\theta}' = \frac{1}{n} \sum g(Y_i), \quad Var(\hat{\theta}') = \frac{1}{n} Var(E[X|Y]).$$

La variance d'estimation a diminué. Dans l'exemple précédent, Y est le nombre N de transactions de la journée, et $E[\sum_{k=1}^N P_k | N] = N\bar{P}$.

Exemple : calcul de π . Pour calculer π , on tire des v.a. dans $[0, 1]^2$ et l'on compte la proportion de points qui sont tombés dans le cercle centré en 0 de rayon 1 ; ici $\theta = \pi/4$:

$$\begin{aligned} X_i &= 1_{U_i^2 + V_i^2 < 1}, \quad U_i, V_i \sim \mathcal{U}([0, 1]) \\ \theta &= E[X] \\ \hat{\theta} &= \frac{1}{n} \sum_{i=1}^n X_i. \end{aligned}$$

La variance de $\hat{\theta}$ est $\theta(1 - \theta)/n = 0,17/n$. On a aussi

$$E[X|U = u] = P(u^2 + V^2 < 1) = P(V < \sqrt{1 - u^2}) = \sqrt{1 - u^2}.$$

D'où l'estimateur

$$\hat{\theta}' = \frac{1}{n} \sum_i \sqrt{1 - U_i^2}.$$

La variance est très simple à calculer et vaut, au facteur n près, $2/3 - (\pi/4)^2 \simeq 0.05$. On peut encore améliorer en utilisant les variables antithétiques :

$$\hat{\theta}'' = \frac{1}{2n} \sum_i \left(\sqrt{1 - U_i^2} + \sqrt{1 - (1 - U_i)^2} \right).$$

On trouve cette fois-ci, par intégration numérique, une variance de $0,007/n$.

Comparaison avec l'échantillonnage stratifié. Dans la formule

$$E[f(X)] = \int f(x) p_{X|Y=y}(dx) p_Y(dy)$$

l'échantillonnage stratifié calcule l'intégrale en x par simulation et l'intégrale en y explicitement, tandis que la méthode de conditionnement fait exactement le contraire.

4.7 Quasi-Monte Carlo : les suites à discrédance faible

Il s'agit de trouver des suites qui permettent une vitesse de convergence plus rapide que les suites aléatoires pour le problème de l'intégration.

Définition 10 Soit $(\xi_i)_{i>0}$ une suite de points de $[0, 1]^d$; pour tout ensemble $B \subset [0, 1]^d$, on désigne par $\lambda_n(B)$ la proportion des n premiers points de cette suite appartenant à B . La *discrédance* est la suite :

$$D_n(\xi) = \sup_B |\lambda_n(B) - \lambda(B)|, \quad \lambda_n(B) = \frac{1}{n} \sum_{i=1}^n 1_B(\xi_i)$$

où λ est la mesure de Lebesgue et le sup est pris tous les B de la forme $B = \prod_{i=1}^d [0, u_i]$, $u_j \leq 1$.

La discrédance compare donc la fonction de répartition empirique à celle de la mesure de Lebesgue. On ne parlera pas des autres mesures de discrédance.

La variation totale de f au sens de Hardy et Krause est

$$V(f) = \sum_{k=1}^d \sum_{i_1 < i_2, \dots, i_k} \int_{[0,1]^k} \left| \frac{\partial^k f}{\partial x_{i_1} \dots \partial x_{i_k}} \right| dx_{i_1} \dots dx_{i_k}$$

où f est calculée en $x_j = 1$ pour les j différents de i_1, \dots, i_k . On a le théorème

Théorème 11 Pour toute fonction f :

$$\left| \frac{1}{n} \sum_{i=1}^n f(\xi_i) - \int_{[0,1]^d} f(u) du \right| \leq V(f) D_n(\xi).$$

Il s'agit maintenant de trouver des suites ξ telles que $D_n(\xi)$ décroisse le plus vite possible. Il a été démontré qu'au mieux $D_n(\xi) = O(n^{-1} \log(n)^{d/2})$ pour $d > 1$ et $O(n^{-1} \log(n))$ pour $d = 1$. On sait par ailleurs fabriquer par des procédés arithmétiques des suites telles que (suites de Halton, de Hamersley, de Lapeyre-Pagès, de Faure, de Sobol, de Niederreiter...)

$$D_n(\xi) = O(n^{-1} \log(n)^d).$$

Une telle suite est appelée suite à discrédance faible ; la suite $\xi_k = (kz_1, \dots, kz_d) \pmod{1}$, où les z_i sont des irrationnels indépendants sur \mathbb{Q} , satisfait $D_n(\xi) = O(n^{-1} \log(n)^{d+1+\varepsilon})$ pour tout $\varepsilon > 0$ [43].

La convergence est donc plus rapide qu'avec le Monte Carlo ordinaire, au moins pour des fonctions f telles que $V(f) < \infty$. Le problème principal est que la borne $V(f) D_n(\xi)$ est rarement très bonne, et difficile à calculer. L'idée suivante [9] est de coupler les deux méthodes :

ALGORITHME (Répétitions)

1. Tirer R variables aléatoires uniformes X_r sur $[0, 1]^d$

2. $\hat{\theta}_r = n^{-1} \sum_{i=1}^n$

$$f(X_r + \xi_i), \quad \hat{\theta} = R^{-1} \sum_{r=1}^R \hat{\theta}_r,$$

En effet les $\hat{\theta}_r$ sont i.i.d et $\text{Var}(\hat{\theta}) = R^{-1} \text{Var}(\hat{\theta}_1)$. On voit en particulier que l'estimation de $\sigma(\hat{\theta})$ est raisonnable, même avec des R modestes. Notons que son ordre de grandeur théorique est au plus $R^{-1/2} n^{-1} \log(n)^d$.

Contrairement aux suites i.i.d, les suites à discrédance faible sous-échantillonnées ne sont plus à discrédance faible. De même, la suite de vecteurs $((\xi_1, \xi_2), (\xi_3, \xi_4), \dots)$ n'est pas à discrédance faible.

De manière plus générale, rien n'interdit de remplacer les suites aléatoires par des suites à discrédance faible dans une simulation qui n'utilise que des $\mathcal{U}([0, 1])$. Mais il faut bien prendre garde à rester dans le contexte du théorème (d devra être le nombre maximum de points nécessaire à construire un X_i , ce qui dans les cas de la station service ferait un vecteur par jour !).

Pour des références et des compléments concernant ce paragraphe nous renvoyons [43]. Des algorithmes sont disponibles à l'adresse www.netlib.no/netlib/toms/738.

4.8 Exercices

Exercice 4.8.1 Motiver et proposer une méthode d'échantillonnage préférentiel pour l'exemple du § 3.3 dans le cas où la probabilité de coupure est très faible (les p_i sont petits et les arcs nombreux).

Exercice 4.8.2 Soit E une variable exponentielle, $X = \cos(E) + E$, et $Y = \cos(E) + E^2$.

1. Quelle est la loi de E sachant $E > A$?
2. Proposer une méthode d'échantillonnage préférentiel pour calculer $P(X > B)$ avec B grand.
3. Adapter la méthode précédente au calcul $P(Y > B)$.

Exercice 4.8.3 On veut estimer $P(X + Y > 3)$ où X et Y sont exponentielles indépendantes de paramètres λ et μ . Proposer une méthode de variables antithétiques, puis une méthode qui combine le conditionnement et les variables antithétiques.

Exercice 4.8.4 On considère la méthode de variables de contrôle avec une fonction de contrôle g et un paramètre α :

$$\hat{\theta} = \alpha E[g(X)] + \frac{1}{n} \sum_{i=1}^n f(X_i) - \alpha g(X_i).$$

Exprimer la variance de $\hat{\theta}$. Pour quelle valeur de α est-elle minimale ? De quel facteur a-t-elle diminué ?

Exercice 4.8.5 Soit $(X_i)_{1 \leq i \leq 2n}$ une suite i.i.d. et les deux estimateurs

$$\hat{\theta}_1 = \frac{1}{n} \sum_{i=1}^n f(X_i) - g(X_i), \quad \hat{\theta}_2 = \frac{1}{n} \sum_{i=1}^n f(X_i) - \frac{1}{n} \sum_{i=n+1}^{2n} g(X_i).$$

À quelle condition sur f et g la variance de θ_1 est-elle inférieure à celle de θ_2 ?

Exercice 4.8.6 Démontrer le théorème 9 à l'aide du théorème suivant :

Théorème 12 Soient $\varphi(x)$ et $\psi(x)$ deux fonctions de \mathbb{R}^d dans \mathbb{R} , croissantes l'une de l'autre séparément pour chacune de leurs variables au sens où pour tout x et tout vecteur δ ayant une seule composante non-nulle :

$$(\varphi(x + \delta) - \varphi(x))(\psi(x + \delta) - \psi(x)) \geq 0$$

et soit X une v.a. sur \mathbb{R}^d à composantes **indépendantes**, alors

$$\text{Cov}(\varphi(X), \psi(X)) \geq 0.$$

REMARQUE : Pour la démonstration de ce théorème, faire d'abord le cas $d = 1$ en prenant l'espérance dans $(\varphi(Y) - \varphi(X))(\psi(Y) - \psi(X)) \geq 0$ où X et Y sont indépendantes de même loi, puis étendre par récurrence aux dimensions supérieures.

Exercice 4.8.7 Démontrer la formule (4.1).

Exercice 4.8.8 Montrer à l'aide du théorème 12 que si f et g sont croissantes l'une de l'autre (c-à-d $(f(y) - f(x))(g(y) - g(x)) \geq 0$), l'estimateur par échantillonnage corrélé

$$\hat{\theta} = \hat{\theta}_1 - \hat{\theta}_2 = \frac{1}{n} \sum_{i=1}^n f(X_i) - g(Y_i), \quad X_i = Q_X(U_i), \quad Y_i = Q_Y(U_i)$$

a une moindre variance que l'estimateur à échantillonnage indépendant.

Exercice 4.8.9 Réinterpréter l'exercice 3.6.4 en fonction de ce qui a été vu dans ce chapitre.

Exercice 4.8.10 On cherche la probabilité qu'une variable exponentielle de paramètre λ soit supérieure à A . On propose un échantillonnage préférentiel en tirant les n variables selon une loi de paramètre μ différent.

Étudier la variance de l'estimateur obtenu : On s'intéressera à l'asymptotique lorsque $A \rightarrow +\infty$ et l'on estimera la variance relative $\text{Var}(\hat{\theta})/\theta^2$ dans les deux situations "échantillonnage normal" et "échantillonnage préférentiel avec μ optimal" ; on donnera également, dans ces deux cas, l'ordre de grandeur du n nécessaire à l'obtention d'une estimée raisonnablement précise.

Exercice 4.8.11 Proposer une méthode d'échantillonnage préférentiel pour calculer la probabilité que le chiffre d'affaire du pompiste du § 3.5.2, sur une journée, dépasse une certaine valeur C

(grande). On supposera qu'une telle augmentation est due principalement à une abondance exceptionnelle de clients.

Proposer une méthode combinant échantillonnage préférentiel et conditionnement pour calculer l'espérance du temps d'interarrivée entre deux clients sachant que le chiffre d'affaire est supérieur à C .

Exercice 4.8.12 (Programmation) Reprendre l'exercice 3.6.7 avec une suite à discrétance faible.

Chapitre 5

Les échantillonneurs de Metropolis et de Gibbs

Soit un ensemble discret E et une fonction f de E dans \mathbb{R}_+ . On veut générer des variables de distribution

$$\pi_i = C f_i$$

où C est la constante de normalisation. La difficulté que l'on rencontre est qu'en général, soit C n'est pas calculable car E est trop grand, soit E est un sous-ensemble difficilement atteignable d'un ensemble connu (p.ex. si π est la mesure uniforme sur l'ensemble des coloriages des départements, avec 5 couleurs, tels que deux départements voisins soient de couleur *différente*). L'algorithme de Metropolis donne une méthode pour simuler une chaîne de Markov de mesure invariante π .

L'inconvénient de la méthode est que d'une part on ne simule plus des variables indépendantes et que d'autre part les variables n'ont pas la loi désirée dès le début car il faut attendre la convergence vers la mesure invariante. Ce deux problèmes seront étudiés au chapitre 7.

L'échantillonneur de Gibbs quant à lui simule des réalisations d'un vecteur aléatoire lorsqu'on sait réaliser facilement la loi de chaque composante sachant les autres. Ces réalisations se font également par une dynamique markovienne.

Les algorithmes présentés dans ce chapitre dans le cadre d'espace d'états finis s'étendent sans difficulté au cas continu (voir les exercices).

5.1 Rappels : chaînes indécomposables et apériodiques

On considère ici une chaîne de Markov X_n sur un ensemble fini E , et l'on note $P = (p_{ij})$ sa matrice probabilités de transitions (attention P désigne également la probabilité des événements). Pour toute probabilité initiale μ (vecteur ligne), μP^n représente la loi de X_n si $X_0 \sim \mu$, et pour tous n, i, j , et toute fonction f (vecteur colonne) sur E :

$$(P^n)_{ij} = P(X_n = j | X_0 = i), \quad (P^n f)_i = E[f(X_n) | X_0 = i], \quad \mu P^n f = E_\mu[f(X_n)].$$

P a toutes ses valeurs propres de module inférieur à 1 et 1 est toujours valeur propre de P . La chaîne est dite apériodique si 1 est la seule valeur propre de module 1 de P , et indécomposable si l'ordre de multiplicité de 1 est 1.

Théorème 13 P est indécomposable ssi il existe un point i qui peut être atteint de tout autre par une suite de transitions non-nulles ¹ :

$$\exists i \in E, \forall j \in E, \max_{n>0} P(X_n = i | X_0 = j) > 0.$$

Une chaîne indécomposable est apériodique ssi pour un i et un $n \geq 0$

$$P(X_n = i | X_0 = i) > 0, \text{ et } P(X_{n+1} = i | X_0 = i) > 0.$$

Il existe toujours une mesure invariante π , $\pi P = \pi$; si P est indécomposable, elle est unique et pour toute fonction g

$$\frac{1}{n} \sum_{k=1}^n g(X_k) \longrightarrow \pi(g) = \sum_i \pi_i g(i) \quad \text{avec probabilité 1}$$

$$\frac{1}{\sqrt{n}} \left(\sum_{k=1}^n g(X_k) - \pi(g) \right) \longrightarrow N(0, \sigma^2) \quad \text{en loi}$$

où σ dépend de g .

Si de plus P est apériodique, alors X_n converge en loi vers son unique mesure invariante $\pi = (\pi_i)_{i \in E}$ et il y a convergence à vitesse géométrique de P^n vers la matrice Π dont toutes les lignes sont égales à π : soit λ_2 la deuxième valeur propre de P et r sa multiplicité, il existe un C tel que

$$\|P^n - \Pi\| \leq C n^{r-1} |\lambda_2|^n. \quad (5.1)$$

Deux exemples types sont donnés dans les exercices 5.5.1.5 et 5.5.1.4. L'irréductibilité est une condition plus forte que l'indécomposabilité, qui impose que tous les points i soient atteignables, ce qui signifie qu'il n'y a pas d'état transient, ou encore que P a une puissance ayant tous ses coefficients > 0 .

Définition 14 P est dite réversible² (ou f -réversible) s'il existe une fonction $f_i \geq 0$ non identiquement nulle telle que pour tous i et j

$$f_i p_{ij} = f_j p_{ji}.$$

Théorème 15 Si P est réversible, alors la mesure

$$\pi_i = \frac{f_i}{\sum_{j \in E} f_j}$$

est invariante. Si de plus P est indécomposable et apériodique, alors P est diagonalisable, et le théorème 13 s'applique avec $r = 1$.

DÉMONSTRATION: L'identification de π est immédiate. La propriété de réversibilité implique que

$$DP = P^T D$$

où D est la diagonale des f_i , ce qui signifie que la matrice $D^{1/2} P D^{-1/2}$, est symétrique ; elle est diagonalisable, et donc P aussi : $P = Q^{-1} \Delta Q$. 1 est unique valeur propre de module 1 de Δ , et donc Δ s'écrit $\Delta = \Delta_1 + \Delta_0$, avec Δ_1 nulle à l'exception d'un 1 sur la diagonale et Δ_0 diagonale avec tous ses coefficients de module inférieur à $|\lambda_2|$. Clairement $P^n = Q^{-1} \Delta^n Q$ converge vers $\Pi = Q^{-1} \Delta_1 Q$ et $P^n - \Pi = Q^{-1} \Delta_0^n Q$ satisfait (5.1) avec $r = 1$ et $C = \|Q^{-1}\| \|Q\|$. ■

¹Dans le langage des chaînes de Markov à valeurs dans un espace quelconque, on dit que la chaîne, ici indécomposable, est ψ -irréductible, avec ici $\psi = \delta_i$ [40].

²Cette propriété est appelée réversibilité de la chaîne car elle implique que, sous sa mesure invariante, la chaîne retournée dans le temps est une chaîne de Markov de mêmes probabilités de transition.

5.2 Algorithme de Metropolis

5.2.1 Principes

Le but est de simuler une loi de probabilité de la forme $\pi_i = f_i / \sum_k f_k$ où les f_i sont connus. Il faut se donner également une matrice p_{ij} telle que :

Tous points i et j tels que $f_i f_j > 0$ peuvent être joints par une chaîne $k_1 = i, k_2, \dots, k_{n-1}, k_n = j$ telle que $p_{k_i k_{i+1}} > 0, p_{k_{i+1} k_i} > 0$ et $f_{k_i} > 0$.

Par exemple $f > 0$ et P est irréductible.

ALGORITHME Tirage de X_n après $X_{n-1} = i$

1. Tirer j dans E avec probabilité p_{ij}
2. Faire aléatoirement $X_n = j$ ou $X_n = i$ avec probabilité q et $1 - q$:

$$q = \min \left(1, \frac{f_j p_{ji}}{f_i p_{ij}} \right).$$

Théorème 16 On se place sous les hypothèses ci-dessus, et l'on suppose de plus que P n'est pas f -réversible. Alors la suite X_n produite par l'algorithme converge en loi vers π , indépendamment de la valeur initiale X_0 , et le théorème 13 s'applique.

DÉMONSTRATION: La matrice de transition de la chaîne (X_n) est

$$\check{p}_{ij} = \min \left(p_{ij}, \frac{f_j p_{ji}}{f_i} \right), \quad i \neq j, \quad \check{p}_{ii} = 1 - \sum_{k \neq i} \check{p}_{ik}$$

et en particulier $f_i \check{p}_{ij} = f_j \check{p}_{ji}$; il y a donc réversibilité et π est bien invariante. Vu l'hypothèse, l'indécomposabilité est immédiate. Pour montrer l'apériodicité, noter que $\check{p}_{ii} > 0$ pour un i . ■

Remarque. Pour que π soit invariante, il est essentiel de garder tous les échantillons, répétitions comprises.

Choix de p . Cet algorithme est utilisé lorsque E est un ensemble très grand, et comme on va le voir, les coefficients p_{ij} seront souvent majoritairement nuls : on ne fait de transition que vers des états proches. Ceci fonctionne bien si π n'est pas trop multimodale, car ce type de transition défavorise les passages d'un mode à l'autre. Dans la majorité des cas, X sera un long vecteur et un voisin X' ne différera de X que par une coordonnée tirée à peu près uniformément.

La détermination de p_{ij} peut résulter d'une approximation de π restreinte au voisinage de i : si $p_{ij} = \pi_j 1_{A_i}(j)$, avec $i \in A_i$ ssi $j \in A_i$ (i et j voisins), alors $q = 1$.

On choisit parfois $p_{ij} = \text{cste}$ (loi uniforme sur E), mais cette option provoquera de très nombreux rejets ($X_n = X_{n-1}$) si la masse de π se concentre sur une petite partie de E , ce qui est fréquent. Les transitions vers des voisins permettent d'éviter ces rejets.

Cas continu. Il faut raisonner avec les densités. Prenons le cas d'une loi de densité $\pi(x) = Cf(x)$ par rapport à la mesure de Lebesgue sur \mathbb{R}^d . On tirera un nouvel état selon une loi de densité $p_x(y)$ et $q = \min \left(1, \frac{f(y)p_y(x)}{f(x)p_x(y)} \right)$. Si $p_x(y)$ consiste à choisir un indice k au hasard et à remplacer x_k par une valeur aléatoire x'_k de loi $p(\xi)$, alors $q = \min \left(1, \frac{f(y)p(x_k)}{f(x)p(x'_k)} \right)$.

5.2.2 Simulation de graphes.

On reprend l'exemple du §3.3. On veut simuler des réseaux ayant la probabilité requise mais en se restreignant à l'ensemble \mathcal{R} de ceux où les points a et b ne sont pas connectés. La solution naturelle est, pour chaque i , de prendre pour p_{ij} la mesure uniforme sur tous les réseaux $j \in \mathcal{R}$ qui ne diffèrent de i que par l'état d'une connexion ; on appelle cet ensemble V_i . On a donc en cas de rétablissement d'une connexion ($f_i = f_j = 1$)

$$q = \min \left(1, \frac{(1-p)|V_i|}{p|V_j|} \right).$$

où p est la probabilité de perte de cette connexion, et en cas de perte p est remplacé par $1-p$ dans cette formule. L'indécomposabilité vient de ce que l'on peut toujours arriver à l'état où toutes les connexions sont faites.

Une méthode plus simple est de prendre pour p_{ij} la mesure uniforme sur tous les réseaux j qui ne diffèrent de i que par l'état d'une connexion, et $f_j = 0$ si $j \notin \mathcal{R}$. Elle semble demander moins de calculs car p_{ij} est constant mais risque de causer beaucoup de rejets, surtout si p est faible.

5.2.3 Marche aléatoire sans recoupement

Dans cet exemple, on réalise un algorithme de Metropolis où f vaut 0 ou 1 et les p_{ij} sont constants et donc $q \in \{0, 1\}$. La marche aléatoire dans \mathbb{Z}^d est le processus

$$W_k = \Delta_1 + \Delta_2 + \dots + \Delta_k, \quad W_0 = 0$$

où Δ_i est tiré uniformément parmi l'ensemble \mathcal{D} des vecteurs de la base canonique et de leur opposés. En somme on se déplace à chaque instant d'une unité dans une des $2d$ directions possibles tirée au hasard. La simulation d'un tel processus ne pose évidemment aucune difficulté. La loi de (W_1, \dots, W_K) est simplement la distribution uniforme sur toutes les trajectoires possibles.

La marche aléatoire sans recoupement de longueur K dans \mathbb{Z}^d est le même processus dont la loi est conditionnée à l'absence de recoupement. Sa distribution est donc uniforme sur l'ensemble

$$E = \{(W_1 \dots W_K) \in \mathbb{Z}^{d \times K} : \|W_i - W_{i-1}\| = 1, \quad \|W_1\| = 1, \quad W_i \neq W_j \text{ si } i \neq j\}.$$

La méthode du rejet pour simuler ces trajectoires marchera très mal car il y a proportionnellement fort peu de trajectoires sans recoupement. On se propose de simuler une chaîne de Markov sur E .

Soit \mathcal{T} l'ensemble des permutations T de \mathcal{D} telles que $T(-v) = -Tv$ (cette condition implique qu'une telle permutation s'étend à une isométrie de l'espace) alors pour tout $1 < k \leq K$ on peut définir

$$\begin{aligned} T_k(W_1, \dots, W_K) &= (W'_1, \dots, W'_K) \\ W'_i &= \Delta'_1 + \Delta'_2 + \dots + \Delta'_i, \\ \Delta'_j &= \begin{cases} \Delta_j & j < k \\ \Delta'_j = T(\Delta_j) & j \geq k. \end{cases} \end{aligned}$$

L'application T_k fait une isométrie sur la fin de la marche sans toucher au début. On tire successivement les X_n :

1. Tirer uniformément k dans $\{1, \dots, K\}$ et T dans \mathcal{T} et poser $X = T_k X_{n-1}$
2. Si X est sans recoupement, alors $X_n = X$, sinon $X_n = X_{n-1}$.

Si $i \neq j$ et $i, j \in E$, alors $p_{ij} = (K|\mathcal{T}|)^{-1}$. La vérification de l'indécomposabilité est un peu difficile [47] (il s'agit de montrer que l'on peut toujours déplier la marche pour la mettre dans l'état où tous les Δ_j sont égaux).

5.3 L'échantillonneur de Gibbs

Soit $X = (X_1, \dots, X_d)$ un vecteur aléatoire de probabilité $\pi(X)$ que l'on veut simuler. On reste, pour simplifier dans le cas où X prend un nombre fini de valeurs bien que ça ne soit pas nécessaire. L'échantillonneur de Gibbs consiste à créer directement une chaîne de Markov réversible indécomposable apériodique de mesure invariante π en tirant chaque coordonnée à son tour selon sa loi conditionnelle aux autres. On notera $X^{k,x} = (X_1, \dots, X_{k-1}, x, X_{k+1}, \dots, X_d)$.

ALGORITHME Tirage de X' après X

1. Tirer k uniformément dans $\{1, \dots, d\}$
2. Tirer x selon la loi de X_k conditionnelle aux autres :

$$Q_k(x|X) = \pi(X_k = x | X_j, j \neq k) = \frac{\pi(X^{k,x})}{\sum_y \pi(X^{k,y})}$$

puis $X' = X^{k,x}$.

La chaîne est réversible avec $f = \pi$ puisque pour $X \neq X'$ et $p_{XX'} \neq 0$ on a $p_{XX'} = d^{-1}Q_k(x|X) = d^{-1}\pi(X')/q_X$ où $q_X = q_{X'}$. On peut montrer qu'il y a indécomposabilité, apériodicité, et donc convergence, si la famille de lois conditionnelles Q_k détermine π de manière unique.

Variantes. Dans le premier point, on peut remplacer la loi uniforme par toute loi qui porte tous les points, ou encore faire un balayage déterministe de tous les points, c-à-d qu'à la n -ième étape $k = n \pmod{d}$.

Il ne faut en revanche pas rafraîchir tous les points simultanément (générer X'_1, \dots, X'_d à partir de (X_1, \dots, X_d) , et considérer (X'_1, \dots, X'_d) comme nouvel échantillon) car on fabrique alors une chaîne dont la loi invariante n'est plus la même [15].

Exemple : formes produit. Dans cet exemple, X_k sera le nombre de communication téléphonique en cours dans la k -ème cellule du réseau UMTS. Si N est le nombre de canaux (fréquences) disponibles, la contrainte de non-interférence sera

$$\sum_{k \in \mathcal{C}} X_k \leq N$$

pour tout ensemble \mathcal{C} de cellules voisines. On notera E_0 l'ensemble des suites (x_1, \dots, x_d) qui satisfont ces contraintes. On montre que typiquement la loi de (X_1, \dots, X_d) a la forme, dite forme produit

$$\pi(x_1, \dots, x_d) = C \pi_1(x_1) \dots \pi_d(x_d) 1_{x \in E_0}$$

où chaque π_k est une mesure de probabilité (cf exercice 5.5.1.6). On a dans cette situation

$$\pi(X_k = x | X_j, j \neq k) = \frac{\pi_k(x)}{\sum_y \pi_k(y)}$$

où la somme est étendue aux y tels que $X^{k,y} \in E_0$. La simulation par échantillonneur de Gibbs est donc très simple (avec éventuellement une méthode de rejet pour tirer selon la loi conditionnelle).

5.4 Applications

Les applications données ici concernent l'estimation ; pour d'autres exemples, voir les exercices ainsi que le chapitre suivant.

5.4.1 Maximum de vraisemblance

Soit une famille de lois de probabilités paramétrées par θ , de la forme

$$p_\theta(X) = \frac{Q(X, \theta)}{Z(\theta)}$$

où la fonction $Q(X, \theta)$ est simple à calculer et la constante de normalisation $Z(\theta)$ est en revanche très difficile à calculer. On a vu des exemples de telles situations (la mesure simulée au § 5.2.2 avec $\theta = p$) et l'on en verra d'autres au chapitre sur les champs de Markov § 6.3. Dans le cas continu, si la mesure de référence est la mesure de Lebesgue, on a $Z(\theta) = \int Q(x, \theta) dx$ où l'intégrale est sur \mathbb{R}^d , d pouvant être très grand, et dans le cas discret, $Z(\theta) = \sum_x Q(x, \theta)$ où la somme s'étend à un ensemble extrêmement grand.

Il est simple de vérifier que l'équation du maximum de vraisemblance se réduit à

$$\nabla_\theta \log Q(X, \theta) = E_\theta[\nabla_\theta \log Q(Y, \theta)].$$

L'algorithme de Metropolis (ou de Gibbs) permet d'estimer le membre de droite pour toute valeur de θ , et donc de faire des algorithmes (itératifs) de calcul de $\hat{\theta}_{MV}$.

5.4.2 L'algorithme EM pour les données manquantes

On dispose d'une famille paramétrique de lois candidates p_θ pour un ensemble de données X . L'algorithme du maximum de vraisemblance consiste à estimer θ par

$$\hat{\theta} = \arg \max_\theta p_\theta(X).$$

Il arrive que certaines données manquent. On note $X = (Y, Z)$, où Y est la partie observée et Z la partie manquante (par exemple X est une matrice dont les lignes sont des vecteurs gaussiens i.i.d. mais certains éléments manquent et θ contient les paramètres de la gaussienne). L'algorithme du maximum de vraisemblance consiste à maximiser la probabilité des données observées :

$$\hat{\theta} = \arg \max_\theta p_\theta(Y), \quad p_\theta(Y) = \int p_\theta(Y, z) dz \quad (5.2)$$

où l'on note également par abus p_θ la loi des données observées. Souvent cette intégrale est incalculable, mais *on est capable de simuler les données manquantes sous leur loi conditionnelle aux observations* $p_\theta(Z|Y)$. La remarque importante est que la solution de l'équation suivante

$$\theta = \arg \max_{\theta'} \int \log(p_{\theta'}(Y, z)) p_\theta(z|Y) dz \quad (5.3)$$

est solution de (5.2) (tout au moins les conditions du premier ordre sont identiques) car une solution de cette équation vérifie en $\theta' = \theta$

$$0 = \nabla_{\theta'} \int \log(p_{\theta'}(Y, z)) p_\theta(z|Y) dz = \int \frac{\nabla p_\theta(Y, z)}{p_\theta(Y, z)} \frac{p_\theta(Y, z)}{p_\theta(Y)} dz = p_\theta(Y)^{-1} \nabla_\theta \int p_\theta(Y, z) dz.$$

L'algorithme EM consiste à fabriquer la suite

$$\hat{\theta}_{n+1} = \arg \max_\theta \int \log(p_\theta(Y, z)) p_{\theta_n}(z|Y) dz.$$

On montre qu'il converge sous des hypothèses assez générales vers une solution de (5.3). Une maximisation exacte à chaque étape est inutilement coûteuse et on pourra en pratique se contenter de chercher $\hat{\theta}_{n+1}$ dans la direction du gradient :

$$\hat{\theta}_{n+1} = \hat{\theta}_n + \gamma_n \delta_n, \quad \delta_n = \int \nabla_{\theta} \log(p_{\theta_n}(Y, z)) p_{\theta_n}(z|Y) dz$$

pour un $\gamma_n > 0$ qui fasse bien augmenter l'intégrale à maximiser (on le calculera à chaque étape, par exemple par une méthode de dichotomie). Le déroulement de l'algorithme impose donc de savoir calculer les fonctions coût et éventuellement gradient (ou une approximation) :

$$f(\theta, \theta') = \int \log(p_{\theta'}(Y, z)) p_{\theta}(z|Y) dz$$

$$g(\theta) = \int \nabla_{\theta} \log(p_{\theta}(Y, z)) p_{\theta}(z|Y) dz$$

ce qui pourra se faire par des simulations de z sous sa loi conditionnelle à Y (algorithme SEM et variantes). C'est ici que l'échantillonneur de Gibbs peut intervenir, comme on va le voir au paragraphe suivant.

Par exemple, si l'on n'utilise pas le gradient, on a l'algorithme (intuitivement clair) suivant

$$\hat{\theta}_{n+1} = \arg \max_{\theta} \sum_{k=1}^K \log(p_{\theta}(Y, z_k))$$

où les z_k sont tous simulés selon la loi $p_{\hat{\theta}_n}(z|Y)$. Pour avoir convergence, vers la bonne valeur, K doit bien entendu tendre vers l'infini avec n . Mentionnons également l'algorithme stochastique du § 8.1.5.

Application. On considère une population de I salamandres femelles et J mâles. On tente d'accoupler chaque femelle avec chaque mâle [36]. Les observations consistent donc en les IJ variables Y_{ij} valant 0 ou 1 selon le succès de l'entremise. On considère le modèle logistique mixte

$$P(Y_{ij} = 1) = \pi_{ij}$$

$$\log \left(\frac{\pi_{ij}}{1 - \pi_{ij}} \right) = \mu + u_i + v_j.$$

Les variables latentes u_i et v_j ne sont pas observées, mais sont i.i.d de loi $\mathcal{N}(0, 1/\theta_u)$ et $\mathcal{N}(0, 1/\theta_v)$. Conditionnellement à u et v les Y_{ij} sont donc indépendants avec la loi donnée ci-dessus.

Notons $\theta = (\mu, \theta_u, \theta_v)$ et $z = (u, v)$. La probabilité de Y, u, v est donc

$$P_{\theta}(Y, z) = P_{\theta}(Y|u, v) P_{\theta}(u, v)$$

$$= c \theta_u^{I/2} \theta_v^{J/2} \prod_{ij} \frac{\exp\{Y_{ij}(\mu + u_i + v_j)\}}{1 + \exp\{\mu + u_i + v_j\}} \exp \left\{ -\frac{\theta_u}{2} \sum_i u_i^2 - \frac{\theta_v}{2} \sum_j v_j^2 \right\}.$$

En revanche la probabilité de Y seul s'obtient en intégrant cette expression par rapport à (u, v) ce qui conduit à une intégrale multiple en dimension élevée sur laquelle il est très difficile de travailler.

Le calcul de f et g se fera en simulant (u, v) selon sa loi sachant Y . Ceci peut se faire par un échantillonneur de Gibbs. La simulation de chaque u_i n'est elle-même pas très évidente et peut se faire par le biais d'une méthode de rejet ; voir aussi [39] pour un algorithme un peu différent.

5.5 Exercices, exemples

5.5.1 Chaînes de Markov

Exercice 5.5.1.1 Deux boîtes à chaussures contiennent des puces qui sautent de l'une à l'autre. Il y a en tout m puces. On considère le nombre X_n de puces dans la première boîte entre le n -ième et le $n + 1$ -ième saut. Donner la matrice de transition de la chaîne de Markov X_n .

Exercice 5.5.1.2 Un volume contient des particules. Sur chaque intervalle $[n, n + 1[$ chacune meurt avec probabilité p , tandis que d'autres arrivent dont le nombre suit une loi de Poisson de paramètre λ . Soit X_n le nombre de particules à l'instant n ; donner la matrice de transition de cette chaîne de Markov (on trouve p_{ij} comme somme de $\min(i, j) + 1$ termes). Montrer qu'une certaine loi de Poisson est invariante.

Exercice 5.5.1.3 Un individu va d'un sommet à un autre d'un graphe en choisissant à chaque instant uniformément parmi les sommets connectés à celui où il se trouve. Vérifier que la chaîne réversible et donner une mesure invariante. À quelle condition est-elle unique ?

Le graphe est un triangle de sommets A, B et C. La probabilité initiale de la chaîne (loi de X_0) donne les poids $1/4$, $1/2$ et $1/4$ à A, B et C. Calculer l'espérance de $f(X_3)$, où f est la fonction qui vaut respectivement 0, 1 et 2 en A, B et C.

Exercice 5.5.1.4 (Décomposabilité et apériodicité) On considère la chaîne de Markov $X_n = \text{signe}(X_{n-1})Z_n$ où les Z_n sont i.i.d uniformes sur $\{1, 2 \dots K\}$ et l'espace d'états est $\{-K, \dots K\}$. Vérifier que X_n est décomposable et apériodique. Montrer comment, si X_0 est connu, on peut se ramener à une chaîne indécomposable.

Exercice 5.5.1.5 (Indécomposabilité et non-apériodicité) On considère la chaîne de Markov $X_n = -\text{signe}(X_{n-1})Z_n$ où les Z_n sont i.i.d uniformes sur $\{1, 2 \dots K\}$ et l'espace d'états est $\{-K, \dots K\}$.

1. Vérifier que X_n est indécomposable et non-apériodique (pour ce dernier point utiliser la caractérisation donnée au théorème 13). Expliquer pourquoi de manière évidente, la chaîne partant d'un état donné ne peut converger vers une mesure invariante.
2. Montrer que de manière générale, lorsqu'il y a un nombre fini d'états, l'existence d'une valeur propre λ pour P équivaut à l'existence d'une fonction f telle que $E[f(X_{n+1})|X_n] = \lambda f(X_n)$. Montrer que -1 est valeur propre de P . En déduire à nouveau la non-apériodicité.
3. Trouver la mesure invariante de la chaîne (elle est unique en raison de l'indécomposabilité).

Exercice 5.5.1.6 (Forme produit) Soient F_1, \dots, F_d , d espaces d'états, E_0 une partie de leur produit, et P_k une matrice de transition réversible sur F_k , de mesure invariante π_k ; on se propose de définir une chaîne de Markov sur E_0 de la façon suivante : si $X = (X_1, \dots, X_d)$, on choisit $1 \leq k \leq d$ uniformément puis x selon la loi $P_k(X_k, \cdot)$; la transition se fait vers $X^{k,x}$ ou X selon que $X^{k,x} \in E_0$ ou non (notation du § 5.3). Vérifier que la matrice de transition globale est réversible et que la mesure invariante est

$$\pi(x_1, \dots, x_K) = C \pi_1(x_1) \dots \pi_K(x_K) 1_{x \in E_0}.$$

5.5.2 Échantillonnage de Gibbs

Exercice 5.5.2.7 Proposer un algorithme d'échantillonnage de Gibbs pour simuler des réalisations du vecteur aléatoire de densité

$$f_{XY}(x, y) = C \exp(-y^2/2 - x^2(1 + y + y^2)/2).$$

C est la constante de normalisation qui garantit une intégrale 1.

Exercice 5.5.2.8 (Un modèle bayésien hiérarchique) On observe des v.a. indépendantes $Y_{ij} \sim \mathcal{N}(\theta_i, \lambda_y^{-1})$, $1 \leq i \leq I$, $1 \leq j \leq J$. Les θ_i sont des effets aléatoires indépendants $\theta_i \sim \mathcal{N}(\mu, \lambda_\theta^{-1})$, puis $\mu \sim \mathcal{N}(0, 1)$ et $\lambda_y, \lambda_\theta \sim \mathcal{E}(1)$.

1. On pose $Y = (Y_{ij})_{ij}$. Écrire la loi de $(Y, \theta, \mu, \lambda_y, \lambda_\theta)$.
2. On pose $\lambda = (\lambda_y, \lambda_\theta)$. Proposer un échantillonneur de Gibbs pour simuler (θ, μ, λ) sous sa loi conditionnelle aux Y_{ij} .
3. Soit $\xi = (\mu, \lambda)$. Proposer une méthode pour calculer l'estimateur bayésien $\hat{\xi}(Y) = E[\xi|Y]$.
4. Proposer alors une méthode simple de simulation pour estimer la variance de $\hat{\xi}(Y)$ sous la loi globale de (Y, θ, ξ) .
5. Dans l'esprit de la question précédente, proposer une méthode de simulation pour estimer la variance de $\hat{\xi}(Y)$ conditionnellement à ξ (il faut supposer que $\hat{\xi}(Y)$ n'est pas trop éloigné de ξ).
6. Comparer heuristiquement ces deux estimateurs de variance.

Exercice 5.5.2.9 Simulation de points avec effet répulsif.

1. Proposer un algorithme d'échantillonnage de Gibbs pour générer des familles de d points (z_1, \dots, z_d) de $[0, 1]$, conditionnés par le fait que les z_k sont au moins à distance ε les uns des autres.
2. Combiner cette méthode avec un rejet dans le cas où chaque z_i appartient à $[0, 1]^2$.
3. Proposer une méthode de Métropolis pour générer des familles de d points (z_1, \dots, z_d) avec une densité par rapport à la loi uniforme sur $([0, 1]^2)^d$ proportionnelle à γ^n où n est le nombre de paires distantes de moins de ε .

Exercice 5.5.2.10 Soient X_1, \dots, X_d d v.a. exponentielles de paramètre λ , et S leur somme. Proposer un algorithme d'échantillonnage de Gibbs pour simuler la loi de (X_1, \dots, X_d) conditionnellement à $S > C$.

Exercice 5.5.2.11 On se donne un graphe G à n sommets. Une configuration consiste à affecter à chaque sommet de G une valeur 0 ou 1 de sorte que deux sommets connectés par un arc ne soient pas à 1. On considère la distribution qui donne à une configuration un poids proportionnel à $p^{n_1}(1-p)^{n-n_1}$ où n_1 est le nombre de 1.

Proposer un échantillonneur de Gibbs pour simuler des réalisations de ce graphe et une méthode pour estimer le nombre moyen de sommets à 1 sous la loi considérée.

Exercice 5.5.2.12 On veut simuler sous une loi $\pi(x_1, \dots, x_K)$ conditionnellement à l'appartenance à $E_0 \subset E$, et l'on sait simuler chaque X_k sous π conditionnellement aux autres X_j , $j \neq k$. L'algorithme de Gibbs va demander de simuler y sous $\pi(X_k = y|X_j, j \neq k)$ conditionnellement à $X^{k,y} \in E_0$. Si ce n'est pas faisable directement (E_0 est trop compliqué), on peut faire des tirages de y jusqu'à ce que $X^{k,y} \in E_0$. Cette méthode sera longue s'il se trouve que peu de valeurs de y réalisent cette contrainte. On peut alors proposer l'algorithme suivant qui permet de faire changer k à chaque fois :

1. Tirer k uniformément dans $\{1, \dots, d\}$.
2. Tirer y sous $P(y) = \pi(X_k = y|X_j, j \neq k)$.
3. Si $X^{k,y} \in E_0$, faire $X' = X^{k,y}$ sinon $X' = X$.

Vérifier que la chaîne ainsi construite est bien réversible et que la mesure invariante est bien la bonne.

5.5.3 Algorithme de Metropolis

Exercice 5.5.3.13 Proposer une méthode de Métropolis pour générer des permutations σ de $\{1, 2, \dots, n\}$ sous la loi $\pi(\sigma) = C \exp(-\lambda \sum_i i \sigma_i)$ (on rappelle que toute permutation est une composée de transposition). On vérifiera l'indécomposabilité.

Proposer une méthode de Métropolis pour générer des permutations σ telles que $\sum_i i \sigma_i > n^3/10$.

Exercice 5.5.3.14 On désire colorier une carte des départements. Un tel coloriage associe à chaque département un vecteur $g_i \in [0, 1]^3$, correspondant aux proportions de rouge, vert et bleu. Comme il faut que deux départements voisins aient des couleurs assez différentes, on se propose de tirer les coloriations avec une probabilité proportionnelle à $\exp(-\beta \sum \|g_i - g_j\|)$ où la somme est étendue seulement aux paires de départements voisins, et β est une constante à choisir, supposée ici connue.

Proposer un échantillonneur de Gibbs pour réaliser ce projet.

Exercice 5.5.3.15 Vérifier que pour simuler la marche aléatoire sans recoupement, on peut aussi tirer k et T selon une distribution non-uniforme sans changer la réversibilité du moment que T et T^{-1} aient même probabilité.

Exercice 5.5.3.16 Soit une matrice symétrique dont les coefficients hors-diagonaux sont $\mathcal{N}(0, 1)$ ($n(n-1)/2$ v.a.) et les coefficients diagonaux sont $\mathcal{N}(0, 2)$ (n v.a.), tous indépendants. On peut montrer que ses valeurs propres ont pour loi

$$C e^{-\frac{1}{4} \sum_{i=1}^n x_i^2} \prod_{1 \leq i < j \leq n} |x_i - x_j| dx_1 \dots dx_n.$$

Proposer un algorithme de Metropolis pour simuler de tels vecteurs.

Chapitre 6

Modèles markoviens

6.1 Généralités

Les chaînes de Markov à état discret ou scalaire peuvent toujours se mettre sous la forme

$$X_n = Q(X_{n-1}, U_n), \quad U_n \sim \mathcal{U}([0, 1]) \quad (6.1)$$

où $Q(x, u)$ est la fonction quantile de la loi de X_n sachant X_{n-1} :

$$Q(x, u) = \min\{y : P(X_n \leq y | X_{n-1} = x) \geq u\}.$$

C'est la méthode d'inversion pour générer X_n connaissant sa loi, cf §1.3.2. Si le processus est non-stationnaire, Q va dépendre de n .

En pratique, la méthode d'inversion est inutile car en général les chaînes ont naturellement la forme (6.1), avec cependant U non-uniforme. Par exemple pour simuler un processus GARCH on utilisera directement des variables gaussiennes :

$$X_n = \sum_{k=1}^p a_k X_{n-k} + \varepsilon_n \sigma_n + \sum_{k=1}^p b_k \varepsilon_{n-k} \sigma_{n-k},$$
$$\sigma_n^2 = s + \sum_{k=1}^p c_k \sigma_{n-k}^2 + \sum_{k=0}^p d_k \varepsilon_{n-k}^2.$$

Les ε_n sont des $\mathcal{N}(0, 1)$ indépendantes, et σ_n^2 représente donc la variance de l'erreur de prédiction de X_n sachant le passé. Ici $(X_n, \dots, X_{n-p+1}, \sigma_n, \dots, \sigma_{n-p+1}, \varepsilon_n, \dots, \varepsilon_{n-p+1})$ est une chaîne de Markov. Ces modèles sont utilisés en finance pour représenter des processus dont la variance est changeante ("volatility clustering") et les queues de distribution élevées.

Le reste du chapitre est consacré au traitement de situations plus compliquées. Nous renvoyons à [51] et [15] pour des discussions plus approfondies sur ce qui suit.

6.2 Simulation de processus à événements discrets

À chaque instant, un système à événements discrets est caractérisé par son état, qui est le nombre d'individus dans chacune de ses parties. Un événement est un changement d'état. La simulation va calculer l'état à chaque instant d'événement, ce qui suffit à avoir l'état sur tout le temps.

L'état nécessaire à la simulation, c'est-à-dire celui qui rend le système Markovien, contiendra davantage d'informations :

$$\text{État} \left\{ \begin{array}{l} \text{état du système} \\ \text{temps absolu (date du dernier événement)} \\ \text{instants d'événements à venir initialisés dans le passé} \\ \text{autres} \end{array} \right.$$

“autres” contient toutes les variables supplémentaires nécessaires à la simulation des événements (par exemple si un temps de service dépend du temps écoulé depuis la dernière maintenance, ce dernier devra être propagé dans la simulation). Un événement à venir initialisé dans le passé est typiquement la livraison d'une commande déjà passée.

De plus certaines variables doivent être récupérées en cours de simulation pour évaluer les statistiques :

$$\text{Variables} \left\{ \begin{array}{l} \text{simulation : état} \\ \text{statistiques annexes : comptes d'événements...} \end{array} \right.$$

6.2.1 Simulation des processus de saut

Les processus de Poisson ont un rôle très important dans de tels systèmes. Ce sont des processus croissants à valeurs sur les entiers, et donc constants par morceaux. Un tel processus N_t est donc caractérisé simplement par les instants où il saute.

Un saut représente, en théorie des réseaux, l'arrivée d'un événement.

Processus à intensité constante. Pour un tel processus, les interarrivées sont les v.a. $\mathcal{E}(\lambda)$. Il suffit donc de simuler une suite i.i.d $\tau_n \sim \mathcal{E}(\lambda)$ et les événements arrivent aux instants $T_k = \sum_{i=1}^k \tau_i$.

Il se trouve que ce processus se caractérise par propriété suivante

$$P(N_{t+h} - N_t \geq 1 | \mathcal{F}_t) = \lambda h + O(h^2)$$

où \mathcal{F}_t est la tribu du passé. La valeur exacte du membre de gauche est bien entendu $\int_0^h e^{-\lambda s} \lambda ds$.

Processus à intensité variable. On a maintenant

$$P(\text{un saut sur } [t, t+h] | \mathcal{F}_t) = \lambda_t h + O(h^2).$$

On voit bien comment cette propriété suffit en théorie à simuler le processus, en prenant h très petit.

La méthode de simulation la plus simple est d'utiliser que le processus N_t se réécrit également $N_{a(t)}^0$, où $a(t) = \int_0^t \lambda_s ds$ et N_t^0 est un processus de Poisson d'intensité 1. Pour une méthode différente de type rejet, voir [51].

Un processus à sauts Markoviens est caractérisé par sa matrice de taux de transition A d'un état à l'autre. Pour simuler ce processus, on procède comme suit : soit $X(t) = i$ l'état courant, pour chaque état $j \neq i$ on simule une variable exponentielle de paramètre A_{ij} et l'on fait la transition correspondant au minimum m de ces variables. On a alors l'état à l'instant $X(t+m)$, et $X(s) = X(t)$ pour $s \in [t, t+m[$.

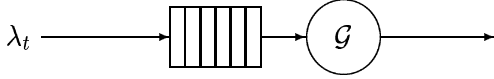
On vérifie que partant de i , j est choisi avec probabilité $A_{ij} / \sum_{k \neq i} A_{ik}$. Si l'on fait abstraction du temps, l'évolution peut donc se faire par une chaîne de Markov. Le temps qu'on reste dans chaque

état peut même être simulé ensuite, sa loi est une v.a. exponentielle indépendante du reste et de paramètre $\sum_{j \neq i} A_{ij}$, si l'état est i .

La matrice $P_t = (P(X_t = j | X_0 = i))_{ij}$ satisfait $P_t = e^{tA}$, avec $A_{ii} = -\sum_{j \neq i} A_{ij}$.

6.2.2 Simulation d'une file à un serveur

On suppose les arrivées poissonniennes d'intensité λ_t et le service de distribution \mathcal{G} :



Les événements qu'on peut considérer sont

$$\text{Événements} \begin{cases} \text{fin d'un service} \\ \text{arrivée d'un client} \end{cases}$$

et un choix naturel de l'état est

$$\text{État} \begin{cases} N = \text{nombre de clients dans le système} \\ t = \text{temps absolu} \\ \tau_D = \text{instant de fin du service en cours (+}\infty \text{ si } N = 0) \\ \tau_A = \text{instant d'arrivée du prochain client} \end{cases}$$

Simulation non-stationnaire en temps absolu. L'état est donc $X = (N, t, \tau_D, \tau_A)$. Le calcul de l'état suivant $X' = (N', t', \tau'_D, \tau'_A)$ se fait en fonction du premier événement qui arrive

- si $\tau_D < \tau_A$ un client libère le serveur ; simuler le temps de service S du suivant dans la file ($S = +\infty$ si $N = 1$), puis $X' = (N - 1, \tau_D, \tau_D + S, \tau_A)$.
- si $\tau_A < \tau_D$, un nouveau client arrive ; si $N = 0$, simuler également son service ; simuler l'instant d'arrivée du suivant τ'_A ; finalement $X' = (N + 1, \tau_A, \tau'_D, \tau'_A)$:
 - $N = 0$: $\tau'_D = \tau_A + S$, $S \sim \mathcal{G}$,
 - $N > 0$: $\tau'_D = \tau_D$.

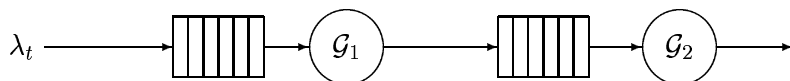
Simulation stationnaire en temps relatif. Si le processus de Poisson a une intensité constante, la simulation peut se faire sans mettre le temps dans l'état, ce qui permet de rendre l'état stationnaire (dans l'exemple précédent $t, \tau_D, \tau_A \rightarrow \infty$.)

$$\text{État} \begin{cases} N = \text{nombre de clients dans le système} \\ \delta_D = \tau_D - t = \text{durée restante pour le service en cours (+}\infty \text{ si } N = 0) \\ \delta_A = \tau_A - t = \text{durée d'ici l'arrivée du prochain client} \end{cases}$$

Alors, on déduit de ce qui précède la simulation du nouvel état $Y' = (N', \delta'_D, \delta'_A)$:

- si $\delta_D < \delta_A$: $Y' = (N - 1, \delta'_D, \delta_A)$, $\delta'_D \sim \mathcal{G}$, $\delta'_D = +\infty$ si $N = 1$.
- si $\delta_A < \delta_D$: $\delta'_A \sim \mathcal{E}(\lambda)$ et $Y' = (N + 1, \delta'_D, \delta'_A)$ avec
 - $N = 0$: $\delta'_D \sim \mathcal{G}$,
 - $N > 0$: $\delta'_D = \delta_D - \delta_A$.

6.2.3 Simulation d'une file à deux serveurs en série



On est dans une situation analogue à la précédente. On appellera système 1 “la première file d'attente et le premier serveur”, et le système 2 est défini de manière analogue. Cette fois :

$$\text{État} \begin{cases} N_1, N_2 = \text{nombre de clients dans les deux systèmes} \\ t = \text{temps absolu} \\ \tau_1, \tau_2 = \text{instant de fin du service en cours de chaque serveur} \\ \tau_A = \text{instant d'arrivée du prochain client} \end{cases}$$

Comme précédemment, on convient que $\tau_i = +\infty$ si $N_i = 0$. On a ici trois cas à considérer selon que τ_1, τ_2 , ou τ_A est le $\min(\tau_1, \tau_2, \tau_A)$. Les détails sont laissés en exercice.

6.2.4 Simulation à état non borné

Il peut arriver que le nombre de variables à prendre en compte dans l'état ne puisse pas être réduit à un nombre fini donné à l'avance. C'est par exemple le cas si dans la file à un serveur du § 6.2.2, les clients s'impatientent et partent si leur service n'a pas commencé après un temps T de distribution donnée : les temps de départ potentiel des clients faisant la queue devront être inclus dans l'état.

On est encore dans cette situation si dans l'on veut savoir, toujours dans l'exemple du § 6.2.2, la fonction répartition du temps de passage d'un client dans le système. Dans ce cas il faut également mettre dans l'état la date d'arrivée de chaque client présent.

6.2.5 Gestion d'inventaire

Ce problème est caractérisé par les points suivants

- Politique de gestion des stocks
 - Renouvellement : soit S la quantité de marchandise en stock, alors dès que $S < s_-$, on achète $s_+ - S$ au prix de $c_m(s_+ - S)$. Le délai de la livraison est d_l . Cet achat est payable à la livraison.
 - Inventaire : Les coûts d'inventaire sont de c_i par unité de temps et de marchandise.
- Dynamique des clients : ils arrivent avec un taux λ pour acheter $\min(Q, S)$ de marchandise au prix unitaire p . Q est une v.a. de loi \mathcal{Q} . La transaction est instantanée.

On pourrait aussi tenir compte de la détérioration des stocks. Le problème concret qui peut se poser est par exemple l'optimisation de s_- , qui doit être assez petit pour éviter trop de commandes et assez grand pour éviter que trop de clients ne partent en achetant moins de marchandise que prévu. On a la structure suivante

$$\text{Événements} \begin{cases} \text{arrivée d'un client} \\ \text{arrivée d'une livraison} \\ \text{passage d'une commande (coïncide avec l'arrivée d'un client)} \end{cases}$$

$$\text{État} \left\{ \begin{array}{l} S = \text{volume du stock} \\ C = \text{volume de la commande en cours} \\ t = \text{temps absolu} \\ \tau_L = \text{instant d'arrivée de la prochaine livraison (+}\infty \text{ si } C = 0) \\ \tau_A = \text{instant d'arrivée du prochain client} \end{array} \right.$$

On s'intéressera également à la variable B représentant le bénéfice. La simulation du nouvel état $X' = (S', C', t', \tau'_L, \tau'_A)$ et de B' se fait selon deux cas :

- si $\tau_A < \tau_L$, on simule la transaction et l'instant d'arrivée du suivant :
 $t' = \tau_A$, $Q \sim \mathcal{Q}$, $S' = S - \min(Q, S)$, $\tau'_A \sim \mathcal{E}(\lambda) + t'$
 et l'on passe une commande en cas de baisse excessive des stocks :
 si $S' < s_-$ et $C = 0$: $C' = s_+ - S'$, $\tau'_L = t' + d_l$
 sinon : $C' = C$, $\tau'_L = \tau_L$
 puis l'évolution du bénéfice : $B' = B + \min(Q, S)p - (t' - t)Sc_i$.
- si $\tau_L < \tau_A$, on a : $X' = (S + C, 0, \tau_L, +\infty, \tau_A)$, $B' = B - c_m(C) - (t' - t)Sc_i$.

6.3 Champs de Markov

Il s'agit ici de simuler des tableaux de variables aléatoires. On ne s'intéressera qu'aux champs de Gibbs qui recouvrent quasiment tous les champs de Markov, et qui représentent ici le modèle adéquat. La question plus générale des données spatiales est extensivement traitée dans [10]. On traite d'abord le cas où le champ prend ses valeurs dans un ensemble fini.

La simulation passera par la création d'une chaîne de Markov dont la mesure invariante correspond à la loi désirée.

6.3.1 Définition

Un des exemples phare pour les champs de Markov est une image numérisée, par exemple sur une grille de 256×256 pixels, chacun ayant un niveau de gris par exemple un entier compris entre 1 et 32.

Appelons S la grille et \mathcal{X} l'ensemble des niveaux de gris. Cette image sera alors considérée comme la réalisation d'un tableau de variables aléatoires indexées par les éléments de S . Le but de la théorie des champs de Markov est de permettre d'étudier les corrélations existantes sur les valeurs entre points proches.

Définition 17 *On appelle structure de voisinage sur un ensemble S une relation symétrique sur S : si a est voisin de b alors b est voisin de a . On convient que a n'est pas voisin de a . On appelle clique un ensemble de points voisins entre eux. On note $\mathcal{V}(a)$ l'ensemble des voisins de a . Les points de S sont généralement appelés des sites.*

En traitement d'image, on considère souvent les 4 ou 8 plus proches voisins. Dans le premier cas les cliques sont les ensembles de deux points à distance minimale et dans le deuxième, ce sont les blocs carrés et leur sous-ensembles (à vérifier !).

Définition 18 *Soit \mathcal{X} un ensemble fini et S un ensemble muni d'une structure de voisinage, on appelle champ de Gibbs une distribution de probabilité sur les tableaux X indexés par les éléments de S et à valeurs dans \mathcal{X} ($X \in \mathcal{X}^S$) ayant la forme suivante :*

$$P(X) = Z^{-1} e^{-\sum_C V_C(X_C)}$$

où la somme est prise sur toutes les cliques, $X_C = (X_a)_{a \in C}$, et les V_C sont des fonctions arbitraires finies. Z est une constante de normalisation.

$P(X)$ désigne ici soit la probabilité de X si \mathcal{X} est fini, soit la densité, si $\mathcal{X} = \mathbb{R}$.

En particulier, toute distribution est un champ de Gibbs pour la structure triviale où tous les points sont voisins. La constante Z est généralement complètement inconnue. Un exemple typique dans le cas des 4-voisinages est :

$$P(X) = Z^{-1} \exp \left\{ - \sum_{ij} \beta_1 (X_{ij} - X_{i-1,j})^2 + \beta_2 (X_{ij} - X_{i,j-1})^2 \right\}.$$

Si $\beta_1 = \beta_2$, cette expression se réécrit

$$P(X) = Z^{-1} e^{-\beta \sum_{\|a-b\|=1} (X_a - X_b)^2}.$$

Dans le cas où $\mathcal{X} = \{-1, 1\}$, on a, les deux expressions :

$$P(X) = Z^{-1} e^{-4\beta_1 N_h - 4\beta_2 N_v} = Z'^{-1} \exp \left\{ \sum_{ij} 2X_{ij} (\beta_1 X_{i-1,j} + \beta_2 X_{i,j-1}) \right\} \quad (6.2)$$

où N_h et N_v sont le nombre de cliques horizontales et verticales avec désaccord. On a ici choisi le même coefficient pour les cliques de même orientation de sorte à avoir une certaine stationnarité. On voit que si les β_i sont positifs, cette distribution a tendance à corrélérer positivement les voisins. Cette corrélation sera d'autant plus forte que le paramètre β_i sera grand.

Théorème 19 *Soit un champ de Gibbs sur S , alors la loi conditionnelle de X_a sachant $(X_b)_{b \neq a}$ est donnée par*

$$P(X_a | X_b, b \neq a) = Z_a^{-1} e^{-\sum_{C \ni a} V_C(X_C)}.$$

En particulier elle ne dépend que de $X_{\mathcal{V}(a)}$.

Si l'on revient à l'exemple (6.2), il vient

$$P(X_{ij} | X_a, a \neq (i, j)) = Z_{ij}^{-1} \exp \{ 2X_{ij} (\beta_1 X_{i-1,j} + \beta_1 X_{i+1,j} + \beta_2 X_{i,j-1} + \beta_2 X_{i,j+1}) \}.$$

Noter que, comme on le voit sur cet exemple, la constante Z_a sera souvent calculable.

Le théorème de Hammersley-Clifford donne une réciproque : sous des conditions très générales, si la loi conditionnelle de X_a sachant les autres ne dépend que de $X_{\mathcal{V}(a)}$, alors le champ est un champ de Gibbs pour cette structure de voisinages.

Exemple : segmentation d'image. Afin de modéliser la présence de régions homogènes distinctes dans une image, on propose le modèle suivant : Une image est la réalisation d'une suite $(X_a, Y_a)_{a \in S}$ où les X_a (étiquette) sont des v.a. prenant leurs valeurs dans $\{1, \dots, n_r\}$ (numéros de région) et Y_a est le niveau de gris. La loi de X est

$$P(X) = Z^{-1} e^{-\beta \sum_{\|a-b\|=1} 1(X_a \neq X_b)} = Z^{-1} e^{-\beta N}.$$

N est la longueur des frontières entre régions. La loi de Y sachant X est une suite de variables indépendante, Y_i ayant une loi de densité $e^{U(y, X_i)}$ (par exemple et pour simplifier $\mathcal{N}(\mu_i, \sigma^2)$). On a donc finalement la loi liée :

$$P(X, Y) = P(X)P(Y|X) = Z^{-1} \exp \left\{ -\beta \sum_{\|a-b\|=1} 1_{X_a \neq X_b} + \sum U(Y_a, X_a) \right\}. \quad (6.3)$$

L'expression ci-dessus conduit à la loi de X sachant Y (qui est la seule variable observée) ce qui va permettre d'estimer les régions, par exemple en maximisant $P(X, Y)$ par rapport à X . Voir l'exercice 6.5.2.9 pour des compléments.

6.3.2 Échantillonnage de Gibbs

L'échantillonneur de Gibbs du § 5.3 devient ici :

ALGORITHME Tirage de X' après X

1. Tirer a au hasard uniformément dans S .
2. Tirer une v.a selon la loi conditionnelle de X_a sachant les autres
3. On déduit X' de X en remplaçant X_a par cette nouvelle valeur.

Au lieu de tirer a au hasard on peut se donner un ordre à l'avance. Par exemple dans le cas des 4-voisinages, on peut renouveler d'abord les sites blancs (S est vu comme un damier) puis les noirs. Notons que le renouvellement des blancs peut se faire en parallèle puisque la distribution sur un site blanc ne dépend que des valeurs sur des sites noirs.

6.3.3 Algorithme de Metropolis

On est, avec les notations du § 5.2, dans la situation où

$$f(X) = e^{-\sum_C V_C(X_C)}.$$

La transition d'un état X à un autre X' va consister à tirer un site a au hasard et à modifier X_a par choix d'une valeur dans \mathcal{X} , si bien que X et X' coïncident sur $S \setminus \{a\}$; les états voisins d'un état donné sont donc en nombre $|S| \cdot |\mathcal{X}|$ et p_{ij} est pour tous i et j l'inverse de cette valeur. Le passage effectif au nouvel état X' se décidera donc avec probabilité

$$q = \min \left(1, e^{\sum_C V_C(X_C) - \sum_C V_C(X'_C)} \right) = \min \left(1, e^{\sum_{C \ni a} V_C(X_C) - V_C(X'_C)} \right).$$

Cette quantité locale est très facile à calculer ; dans l'exemple des 4-voisinages, on a si $a = (i, j)$ et toujours $\mathcal{X} = \{-1, 1\}$:

$$\sum_{C \ni a} V_C(X_C) - V_C(X'_C) = 4X_{ij}(\beta_1 X_{i-1,j} + \beta_1 X_{i+1,j} + \beta_2 X_{i,j-1} + \beta_2 X_{i,j+1}).$$

6.3.4 Exemples où \mathcal{X} est infini

Dans le cas où $\mathcal{X} = \mathbb{R}$, il suffit de considérer que P est la densité de la loi par rapport, par exemple, à la mesure de Lebesgue : $p(x) = Z^{-1} \exp\{-\sum_C V_C(x_C)\}$.

Champ gaussien autorégressif. On va supposer que $S = \{a_1, \dots, a_n\}$ est une partie finie de l'espace euclidien. Considérons la distribution pour $X = (X_a)_{a \in S}$:

$$\log p(X) = -\frac{1}{2\sigma^2} \sum_{a,b} (X_a - \mu_a) c_{ab} (X_b - \mu_b) + cst, \quad c_{aa} = 1 \tag{6.4}$$

où les μ_i sont des réels et la matrice $C = (c_{ab})_{a,b \in S}$ est définie positive. Il s'agit d'une distribution gaussienne. Il est facile de vérifier que la loi d'un X_a sachant les autres est une gaussienne de variance σ^2 et de moyenne :

$$\hat{X}_a = m_a - \sum_{b \neq a} c_{ab} X_b, \quad m_a = \sum_b c_{ab} \mu_b. \quad (6.5)$$

Les voisins de a sont donc les points b tels que $c_{ab} \neq 0$, et m_a est la moyenne de X_a sachant les autres nuls. Il se trouve que pour m et $C > 0$ donnés, P est la seule distribution satisfaisant cette spécification de loi conditionnelle. Souvent on préfère partir de cette spécification pour en déduire m et C , puis μ . Généralement c_{ab} ne dépendra que de $b - a$ (le champ recentré est stationnaire).

Par exemple dans le cas d'une grille régulière du plan, on pourra partir du modèle local

$$\hat{X}_{ij} = \frac{1}{4}(X_{i-1,j} + X_{i+1,j} + X_{i,j-1} + X_{i,j+1})$$

qui définit les m_a (nuls) et la matrice C (elle est bien positive). Pour les points $a = (i, j)$ du bord, si, par exemple, $(i + 1, j)$ sort du domaine on se donne à l'avance une valeur pour $X_{i+1,j}$ dont l'effet est d'ajouter un terme m_a .

On simule facilement de tels champs avec un échantillonneur de Gibbs.

Ce modèle est utilisé par exemple pour étudier le rendement de parcelles voisines de même dimensions [10] : X_a est la production de la parcelle a .

Attention, les erreurs de prédiction $X_a - \hat{X}_a$ ne sont pas des variables aléatoires indépendantes ; pas plus que dans une série temporelle autorégressive, ne le sont les erreurs de prédictions utilisant passé et futur. En revanche, $X_a - \hat{X}_a$ est indépendante de $X_b, b \neq a$.

Modèle de binomial conditionnel. Considérons les données de mort subite de nouveau-né dans les 100 comtés de Caroline du Nord [10]. On a un tableau à 100 lignes et 2 colonnes : la première colonne contient, pour a variant de 1 à 100, le nombre X_a de morts subites de bébés de moins d'un an entre 1974 et 1978 dans le comté a , et la seconde le nombre n_a de naissances durant cette période. Un modèle possible est le modèle binomial suivant (les n_a sont déterministes)

$$P(X_a | X_b, b \neq a) = Z_a^{-1} \frac{\lambda_a^{X_a}}{X_a! (n_a - X_a)!}, \quad \lambda_a = e^{\alpha + \sum_b c_{ab} X_b} \quad (6.6)$$

avec $c_{aa} = 0$ et $c_{ab} = c_{ba}$. Il existe bien une loi de probabilité satisfaisant cette spécification (exercice 6.5.2.11).

Ce modèle est visiblement surparamétré puisqu'il y a 4950 paramètres c_{ab} en jeu pour ici 100 observations. Une solution est de se donner une forme particulière :

$$c_{ab} = \gamma d(a, b)^{-k}$$

où $d(a, b)$ est la distance entre les comtés a et b . Le modèle n'a plus pour paramètres que α, γ et k . On peut également de faire dépendre α de la région, $\alpha = \alpha_r$, ce qui fait alors 14 paramètres (il y a 12 régions en Caroline du Nord).

En dehors de l'intérêt de l'estimation des paramètres, on peut également considérer ce modèle comme une base pour faire des tests ; par exemple, on pourra tester les hypothèses " $\alpha_r = cst$ " (stationnarité, pas de différence significative entre les régions) ou encore " $\gamma = 0$ " (pas de corrélation significative entre comtés proches).

6.4 Modèles spatiaux dynamiques

Il s'agit d'un processus à sauts markoviens qui vit sur une grille S (partie finie de \mathbb{Z}^2 , ou éventuellement un domaine plus général), et évolue en fonction de ses voisins avec une dynamique de processus de saut. Partons d'un exemple simple de dynamique des épidémies où 3 états pour chaque individu (point de la grille, ou "site") sont possibles :

$$\left\{ \begin{array}{l} 0 : \text{sain} \\ 1 : \text{infecté} \\ 2 : \text{immunisé.} \end{array} \right.$$

Le processus $X(t)$ décrit à chaque instant l'état dans lequel se trouve chaque site : $X_{ij}(t) \in \mathcal{X}$, $(i, j) \in S$. Dans cet exemple les transitions se font de la façon suivante :

transition	$0 \rightarrow 1$	$1 \rightarrow 2$	$2 \rightarrow 0$
taux	$\lambda_0 n_{ij}$	λ_1	λ_2

où n_{ij} est le nombre de voisins infectés (à l'état 1). Le nombre total d'états est 3^N , mais de chaque état ne partent que N transitions possibles, ce qui rend la simulation faisable (cf § 6.2.1).

Le même modèle est utilisé pour les feux de forêt, auquel cas les trois états pour chaque arbre sont "détruit", "en feu" et "vivant" (pour un article récent sur le sujet voir [60]).

Compétition entre espèces et dynamique de populations. En chaque site pousse soit de l'herbe, soit un buisson, soit un arbre ; on a donc trois états, et cinq transitions correspondant à la naissance ou la mort d'un buisson ou d'un arbre :

	transition	taux
	$0 \rightarrow 1$	$\lambda_1 n_{ij}(1)$
0 : herbe	$0 \rightarrow 2$	$\lambda_2 n_{ij}(2)$
1 : buisson	$1 \rightarrow 0$	λ_3
2 : arbre	$1 \rightarrow 2$	$\lambda_4 n_{ij}(2)$
	$2 \rightarrow 0$	λ_5

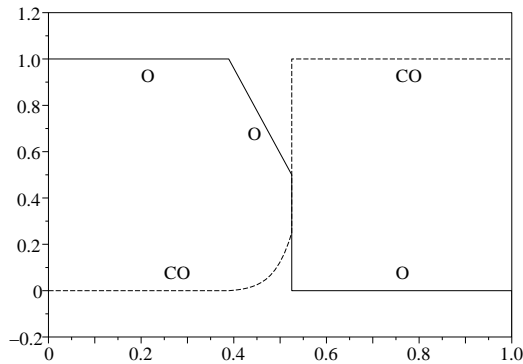
$n_{ij}(1)$ est bien entendu le nombre de buissons dans les sites voisins. Ici le nombre de transitions possibles partant d'un état est $2n_0 + 2n_1 + n_2$ où n_i est le nombre de sites dans l'état i . Ce type de modèle s'étend immédiatement à davantage d'espèces, et peut faire apparaître des comportements très différents selon les valeurs des paramètres (disparitions d'espèces, coexistence, oscillations...) ; voir p.ex. [56].

Catalyse. S représente maintenant la surface d'un catalyseur. L'air ambiant contient du monoxyde de carbone CO et de l'oxygène O₂. Le catalyseur transformera le CO en CO₂ au travers des trois transitions suivantes [6] :

1. une molécule de CO atterit sur un site libre
2. une molécule d'O₂ arrive sur deux sites libres voisins et se scinde en deux avec un atome par site

3. une molécule de CO et un atome d'oxygène se trouvant sur deux sites voisins reagissent formant une molécule de CO₂ qui s'échappe.

Chacune de ces transitions se fait avec son propre taux. Dans l'article original de Ziff, Gulari et Barshad (1986), la dernière transition est instantanée (taux infini), et p désigne $\lambda_1/(\lambda_1 + \lambda_2)$ (λ_1 et λ_2 sont les taux des deux transitions restantes). p peut s'interpréter comme la proportion de CO dans le mélange ambiant CO/O₂. Ils ont obtenu par simulation les résultats de la figure 6.4 pour les concentrations de CO et d'O sur la surface en régime stationnaire. Les résultats font apparaître trois régimes :



$p \in [0, p_1[$: empoisonnement à l'oxygène
 $p \in]p_1, p_2[$: coexistence
 $p \in]p_2, 1]$: empoisonnement au CO
 $p_1 \simeq 0,389$ et $p_2 \simeq 0,525$.

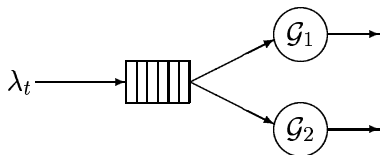
FIG. 6.1 – Concentrations en fonction de p .

Pour la simulation, on réalise cette dynamique sur une grille, et pour éviter d'avoir un empoisonnement à long terme (inévitables sur une grille finie) on interdit à toute molécule de s'installer sur le bord (une molécule ne peut venir au bord que si cela occasionne une transition instantanée de type 3).

6.5 Exercices

6.5.1 Processus à événements discrets

Exercice 6.5.1.1 On considère une file à deux serveurs en parallèle :



Le client en tête de file occupe le premier serveur qui se libère, et le serveur 1 si les deux sont libres. Quel état choisira-t-on ? Expliciter le schéma de simulation.

Exercice 6.5.1.2 Un système subit des chocs à des instants poissonniens. À chaque fois, les dégâts ont une loi \mathcal{G} donnée, et s'amortissent exponentiellement avec le temps : si x_i est la gravité d'un choc intervenu à l'instant t_i , le dégât résiduel à t sera $x_i e^{-\alpha(t-t_i)}$. Les dégâts s'ajoutent et l'on suppose que le système tombe en panne lorsque leur somme dépasse un certain seuil C .

Proposer une méthode pour simuler les instants de panne (il faut tirer profit de la forme de l'amortissement qui permet de simuler avec un état particulièrement réduit).

Exercice 6.5.1.3 On considère une file d'attente à serveur unique. Montrer la relation exprimant l'attente W_i du i -ième client à celle du précédent, au temps de service du précédent S_{i-1} et à l'inter-arrivée A_i :

$$W_i = (W_{i-1} + S_{i-1} - A_i)_+, \quad W_0 = 0.$$

On pourra commencer par faire un schéma, puis faire ensuite la démonstration mathématique.

Exercice 6.5.1.4 Expliciter le schéma de simulation de l'exemple du pompiste du § 3.5.2.

Exercice 6.5.1.5 [51] Un atelier possède $m + n$ machines et a besoin de n machines pour être opérationnel. Un employé est chargé de la réparation des machines en panne. On suppose qu'en sortie de réparation, une machine repart à neuf. S'il y a $m + 1$ machines en panne, tout s'arrête.

Proposer un modèle pour ce système et expliciter l'algorithme de simulation.

Indication : L'état est de dimension un peu supérieure à n .

Exercice 6.5.1.6 (Importance splitting [24]) On cherche à calculer la probabilité d'un événement A assez rare pour un processus demandant une simulation assez longue (probabilité de blocage dans un réseau téléphonique...). L'idée est la suivante : lorsqu'au cours d'une simulation on est proche du blocage (charge du réseau supérieure à un seuil...), on fait repartir de cette situation plusieurs simulations indépendantes, afin d'augmenter ses chances d'observer A ; si l'on s'approche encore (charge supérieure à un nouveau seuil), on multiplie à nouveau les simulations... Le problème est que bien entendu, si l'on observera A plus souvent, les simulations ne sont plus tirées selon la même loi, et ne sont plus indépendantes.

Le cadre théorique adéquat est le suivant. Soit une suite d'événements $B_i, i = 1 \dots k$ ("dépassement du i -ième seuil") $A \subset B_i$ et une suite de tribus $\mathcal{F}_1, \dots, \mathcal{F}_k$ telles que $B_i \in \mathcal{F}_{i+j}, j \geq 0$ (en pratique $\mathcal{F}_1 \subset \dots \subset \mathcal{F}_k$ et $B_i \in \mathcal{F}_i$, et très souvent $A \subset B_k \subset \dots \subset B_1$). On part de N_0 simulations indépendantes. Si pour une simulation B_1 est atteint, on multiplie les simulations en faisant N_1 (choisi à l'avance) tirages conditionnellement à \mathcal{F}_1 , puis on fait de même avec B_2, \dots multipliant ainsi les simulations à partir du contingent de départ.

Montrer qu'un estimateur sans biais de $P(A)$ est

$$\hat{p} = \frac{1}{N_0 N_1 \dots N_k} \sum 1_A$$

où la dernière somme comptabilise tous les succès observés (indication : noter qu'on peut également multiplier les simulations si B_i n'est pas atteint ; bien voir le rôle de l'hypothèse d'appartenance aux tribus). Dans l'exemple \mathcal{F}_i est la tribu du passé de la simulation avant l'atteinte du i -ième seuil (toute la simulation si le seuil n'est pas atteint).

Noter que $N_0 N_1 \dots N_k$ est bien plus grand que le nombre de simulations effectivement réalisées ; son inverse est le pendant du terme $3^{-N_i} e^{-\sum 2\lambda_a T_i}$ de l'exemple du § 4.1.

6.5.2 Champs Markoviens

Exercice 6.5.2.7 Soit un champ de Gibbs sur S et $A \subset S$ montrer que la loi conditionnelle de X_A sachant $(X_b)_{b \notin A}$ est de la forme

$$P(X_A | X_b, b \notin A) = Z^{-1} e^{-\sum_{A \cap C \neq \emptyset} V_C(X_C)}.$$

où Z ne dépend que des valeurs de X en dehors de A .

Exercice 6.5.2.8 [15] On considère le champ de Gibbs binaire sur $S = \{1, \dots, 2n\}$ de loi

$$\pi(x) = Z^{-1} e^{U(x)}, \quad U(x) = \alpha \sum_i x_i + \beta \sum_{i=1}^{2n-1} x_i x_{i+1} + \gamma \sum_{i=1}^{2n-2} x_i x_{i+2}, \quad x_i = \pm 1.$$

1. Quels sont les points voisins d'un point $i \in S$?
2. Soit P (resp. I) l'ensemble des indices pair (resp. impairs). Quelle est la loi de X_I sachant X_P ? Que se passe-t-il si $\gamma = 0$?
3. Désormais $\gamma = 0$. On observe un vecteur $(Y_i)_{i \in P}$ de variables indépendantes conditionnellement à X telles que $P(Y_i = -X_i) = 1 - P(Y_i = X_i) = \varepsilon$.
 - (a) Montrer qu'on a $P(Y_i|X) = e^{\delta X_i Y_i} / (2 \cosh(\delta))$ pour un δ bien choisi (on calculera $P(Y_i|X)$ dans les deux cas $X_i Y_i = 1$ et $X_i Y_i = -1$).
 - (b) En déduire la loi de X sachant Y .
 - (c) En déduire une méthode basée sur un échantillonneur de Gibbs pour estimer $E[X|Y]$.

Exercice 6.5.2.9 On reprend l'exemple de la segmentation d'image du § 6.3. On suppose que l'on sait réaliser la maximisation de (6.3) en X (c'est en fait un problème difficile). Le paramètre β et ceux de la loi de Y sachant X (les μ_i et σ) ne sont pas connus. Il faut donc les estimer.

1. En s'inspirant du § 5.4, proposer un algorithme pour réaliser cette estimation, préalablement à celle de X .
2. Proposer un algorithme itératif, moins rigoureux mais plus simple, pour estimer simultanément X , les μ_i et σ , si β est connu. Expliquer pourquoi l'estimation de β par cette méthode est irréaliste.

Exercice 6.5.2.10 Vérifier la formule (6.5).

Exercice 6.5.2.11 On s'intéresse au modèle binomial du § 6.3.4.

1. En s'inspirant des modèles de champs gaussiens autorégressifs, prouver l'existence d'une loi de probabilité satisfaisant (6.6) si $c_{ab} = c_{ba}$, $c_{aa} = 0$.
2. Trouver de même une loi de probabilité satisfaisant la contrainte (loi de Poisson tronquée) :

$$P(X_a | X_b, b \neq a) = Z_a^{-1} \frac{\lambda_a^{X_a}}{X_a!} 1_{X_a \leq n_a}, \quad \lambda_a = e^{\alpha + \sum_b c_{ab} X_b}.$$

3. Même question avec cette fois le modèle de Poisson :

$$P(X_a | X_b, b \neq a) = Z_a^{-1} \frac{\lambda_a^{X_a}}{X_a!}, \quad \lambda_a = e^{\alpha + \sum_b c_{ab} X_b}.$$

Pourquoi est-il nécessaire ici que $c_{ab} \leq 0$ pour tous a et b ? (Commencer par étudier le cas où il n'y a que 2 sites $S = \{a, b\}$).

6.5.3 Modèles spatiaux dynamiques

Exercice 6.5.3.12 (Simulation) Simuler l'épidémie avec $\lambda_2 = 1$, $\lambda_1 = 0.1$ et $\lambda_0 = 0.04$. Réessayer avec $\lambda_0 = 0.01$.

Exercice 6.5.3.13 Proposer un modèle spatial proie/prédateur qui tienne compte en particulier des naissances, morts et déplacements des deux espèces.

Chapitre 7

Simulation de procesus : convergence

On s'intéresse au problème général suivant :

On sait simuler une chaîne de Markov, que l'on suppose indécomposable et apériodique, et l'on veut estimer la moyenne d'une certaine fonction f sous la mesure invariante : $\theta = E_\pi[g(X_n)]$.

Rappelons les deux points importants déjà mentionnés :

- ▶ on ne simule plus des variables indépendantes
- ▶ les variables n'ont pas la loi π dès le début ; il y a seulement convergence.

Le premier point introduit de la variance supplémentaire dans les estimateurs tandis que le second introduit du biais. Le théorème 13 garantit que ce biais décroît avec vitesse exponentielle ; cependant, lorsque l'espace d'états est très grand, il arrive souvent que dans la formule (5.1) $|\lambda_2|$ soit très proche de 1 et C très grand. La méthode simple consistant à simuler une trajectoire X_n et à poser

$$\hat{\theta} = \frac{1}{N} \sum_{n=1}^N g(X_n)$$

devra être améliorée. Le problème posé dans toute sa généralité est donc difficile, en particulier celui de l'estimation de la variance d'estimation. L'espoir est généralement que $|\lambda_2|$ ne soit pas trop proche de 1 ou que C soit raisonnablement petit, mais encore faut-il être capable de mettre cette propriété à profit.

7.1 Algorithme des répétitions

Une méthode pour avoir une estimation de la variance d'estimation est de tirer plusieurs trajectoires indépendamment dans le même esprit que l'algorithme du § 4.7

ALGORITHME

1. Se donner une mesure initiale π_0 , le nombre de répétitions R et la longueur des trajectoires N
2. Simuler R trajectoires de longueur N , $X_{r,n}$ avec loi initiale π_0
3. $\hat{\theta}_r = N^{-1} \sum_n g(X_{rn})$, $\hat{\theta} = R^{-1} \sum_r \hat{\theta}_r$
4. $\hat{\sigma}(\hat{\theta})^2 = R^{-2} \sum_r (\hat{\theta}_r - \hat{\theta})^2$.

La validité de la méthode ne pose pas de problème. Le véritable ennui est que *le biais reste inconnu* et donc l'erreur sur θ est inconnue. En particulier N doit être choisi assez grand de sorte à oublier les conditions initiales et réduire le biais.

Choix de π_0 . Il sera bon de prendre pour π_0 une mesure assez proche de π si possible pour diminuer le biais et de support assez grand pour être sûr de ne pas faire une mesure trop locale si la chaîne communique mal ; dans ce dernier cas, même si la valeur de $\hat{\sigma}(\hat{\theta})$ n'est pas très précise pour des R modestes, elle donnera un ordre de grandeur correct pourvu que π_0 soit bien choisi.

Choix de R . Il semble a priori inutile de prendre R trop grand ; à moins de partir de $\pi_0 = \pi$, on peut montrer qu'on a toujours intérêt à investir dans des trajectoires longues plutôt que dans des répétitions, voir [14] § 6.6.

7.2 Élimination du transitoire

On se propose maintenant de ne pas introduire les variables dès le début de simulation en raison de la lenteur de convergence vers la mesure initiale. Ce retard de convergence arrivera typiquement si l'on part par malchance d'une zone transiente, mais sur laquelle on reste avec très grande probabilité. On élimine donc une période transitoire dite "période d'échauffement", et le nouvel algorithme sera

ALGORITHME

1. Se donner une mesure initiale π_0 , le nombre de répétitions R , la longueur des trajectoires N et le temps d'échauffement d .
2. Simuler R trajectoires de longueur N , X_n^r avec loi initiale π_0
3. $\hat{\theta} = R^{-1} \sum_r \hat{\theta}_r$, $\hat{\theta}_r = (N - d)^{-1} \sum_{n=d+1}^N g(X_n^r)$
4. $\hat{\sigma}(\hat{\theta})^2 = R^{-2} \sum_r (\hat{\theta}_r - \hat{\theta})^2$.

Le problème est maintenant de déterminer le meilleur d et l'erreur sur θ . Une approche graphique est d'essayer toutes les valeurs de d (ou un échantillonnage) et de tracer l'évolution de $\hat{\theta}$ en fonction de d , ainsi que les barres d'erreur à 2σ . On verra plus ou moins nettement selon l'expérience se dégager une valeur raisonnable de d . Dans la figure ci-dessous, on voit un biais se dessiner avant la valeur 600. On voit également la difficulté d'utilisation de cette méthode.

Toutefois l'inspection de ces courbes à elle seule ne peut rien décider : on ne peut estimer le temps d'échauffement qu'à condition d'avoir fait une expérience au moins aussi longue que celui-ci ! Tout ceci reste donc très expérimental.

7.3 Régénération

7.3.1 Renouvellement

Fixons $i_0 \in E$ et considérons les temps d'atteinte successifs de l'état i_0 , supposé récurrent, pour la chaîne partant de $X_0 = i_0$:

$$T_0 = 0$$

$$T_q = \inf(n > T_{q-1} : X_n = i_0).$$

Les instants T_n seront appelés **instants de renouvellement** car les variables à valeurs dans $\cup_d \mathbb{R}^d$

$$Z_q = (X_{T_{q-1}}, \dots, X_{T_q-1})$$

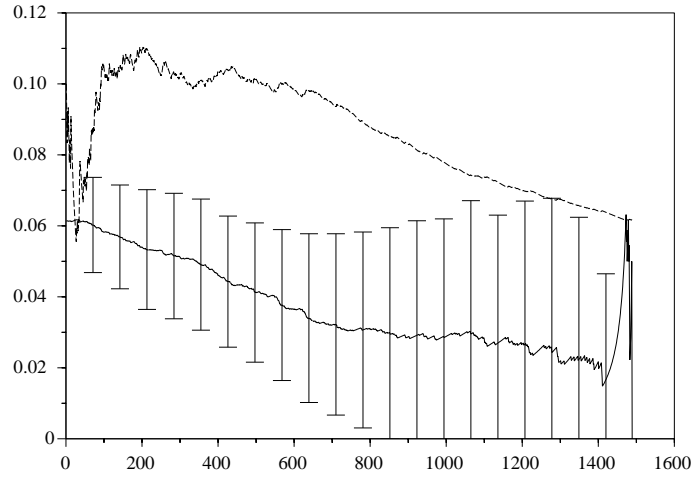


FIG. 7.1 – Courbe de $\hat{\theta}$ avec la barre d'erreur à $2\hat{\sigma}(\hat{\theta})$ en fonction du décalage (nombre de points initiaux éliminés). Au dessus se trouve la courbe de l'évolution de l'estimateur sans décalage en fonction du nombre de points utilisés (son dernier point a donc même ordonnée que le premier de la courbe continue). Il s'agit de l'estimation de π_1 dans une chaîne de Markov à 4 états qui communiquent mal, d'où la lenteur de convergence. La vraie valeur est 0,044. On a pris $R = 10$ et $\pi_0 = \delta_1$. La période d'échauffement est visiblement de l'ordre de 600 au moins. Noter la lenteur de convergence de l'estimateur sans décalage.

sont i.i.d [7] : leur loi commune est celle de la chaîne partant en i_0 et arrêtée en i_0 . Les variables (S_q, τ_q) définies par

$$S_q = \sum_{n=T_{q-1}}^{T_q-1} g(X_n), \quad \tau_q = T_q - T_{q-1}.$$

forment donc une suite i.i.d. Par ailleurs on peut réécrire l'estimateur basé sur les T_Q premières variables :

$$\hat{\theta} = \frac{Q^{-1} \sum_{n=1}^Q S_q}{Q^{-1} \sum_{n=1}^Q \tau_q} = \frac{\bar{S}}{\bar{\tau}}.$$

Il s'ensuit que

$$\theta = \frac{E[S_1]}{E[\tau_1]}.$$

En utilisant la normalité asymptotique $(\bar{S}, \bar{\tau}) \simeq (E[S_1], E[\tau_1]) + Q^{-1/2}(N_S, N_\tau)$, et en développant au premier ordre en $Q^{-1/2}$ dans l'expression de $\hat{\theta}$, on obtient facilement

$$\sqrt{Q} \bar{\tau} \frac{\hat{\theta} - \theta}{s} \rightarrow \mathcal{N}(0, 1), \quad s^2 = \text{Var}(N_S - \theta N_\tau) = \text{Var}(S_1 - \theta \tau_1).$$

Le problème du biais a disparu (un biais existe toujours, mais il est simplement dû au fait que $\hat{\theta}$ est un rapport de variables aléatoires, ce qui n'a que des conséquences secondaires) et l'on a l'estimée

de la variance :

$$\hat{\sigma}(\hat{\theta})^2 = \frac{1}{\bar{\tau}^2 Q^2} \sum_q (S_q - \tau_q \hat{\theta})^2 = \frac{1}{T_Q^2} \sum_q (S_q - \tau_q \hat{\theta})^2. \quad (7.1)$$

Cette méthode, malheureusement rarement applicable, est donc infiniment supérieure aux précédentes. Si l'on compte que s est au plus de l'ordre de $\bar{\tau}$, on obtient un écart-type au pire de l'ordre de $Q^{-1/2}$; il peut être bien plus petit car la variance de $S_1 - \theta \tau_1$ dépend aussi de g et de la nature de la chaîne. En vertu du théorème qui suit, le nombre de variables simulées sera quant à lui de l'ordre de Q/π_{i_0} , et l'on a donc intérêt à partir d'un point de forte mesure invariante. Noter que si l'on arrive à avoir Q de l'ordre de quelques unités, les estimées de variance sont déjà raisonnables.

Théorème 20 *Soit X_n une chaîne de Markov à valeurs dans un espace dénombrable, de mesure invariante unique π , et T_i le premier instant d'atteinte de l'état i , alors*

$$E_i[T_i] = \pi_i^{-1}.$$

Pour la démonstration, voir [7] § 6.9.

7.3.2 Théorie générale

La propriété d'indépendance des trajectoires partielles, et donc l'applicabilité de la formule (7.1), s'étend à d'autres situations.

Théorème 21 *Soit X_n une chaîne de Markov homogène, T_1 un temps d'arrêt fini avec probabilité 1 et f la fonction telle que $T_1 = f(X_1, X_2, \dots)$. Soit $(T_q)_{q \geq 1}$ la suite croissante de temps d'arrêt, définie par*

$$T_{q+1} = f(X_{T_q}, X_{T_q+1}, \dots).$$

Si l'on a la propriété de régénération :

$$P(X_n = x | T_1 = n, X_{n-1}, X_{n-2}, \dots) = \varphi(x)$$

où φ est une certaine fonction, alors les variables

$$Z_q = (X_{T_{q-1}}, \dots, X_{T_q-1}) \quad (7.2)$$

sont i.i.d.

DÉMONSTRATION: Notons $X_n^+ = (X_i)_{i \geq n}$ et $X_n^- = (X_i)_{0 \leq i \leq n}$. En vertu de la formule de Bayes, puis de la propriété de Markov, on a

$$\begin{aligned} & P(X_n = x, X_{n+1}^+ \in B | T_q = n, X_{n-1}^-) \\ &= P(X_{n+1}^+ \in B | X_n = x, T_q = n, X_{n-1}^-) P(X_n = x | T_q = n, X_{n-1}^-) \\ &= P(X_{n+1}^+ \in B | X_n = x) \varphi(x) \end{aligned}$$

qui ne dépend ni de n ni de X_{n-1}^- ; par conséquent pour tout C

$$P(X_n^+ \in C | T_q = n, X_{n-1}^-) = \psi(C)$$

pour une certaine mesure ψ indépendante du reste. Comme sur $\{T_q = n\}$, Z_{q+1} est une fonction du futur après X_n , il s'ensuit que

$$P(Z_{q+1} \in A | T_q = n, X_{T_q-1}^-) = \psi_1(A)$$

pour une certaine mesure ψ_1 et donc, comme Z_q^- est mesurable par rapport à $X_{T_q-1}^-$:

$$P(Z_{q+1} \in A | Z_q^-) = \psi_1(A)$$

Ce qui prouve la stationnarité et l'indépendance. ■

Nous allons présenter au § 7.3.3 une méthode qui permet de créer davantage d'instants de régénération [44].

7.3.3 Méthode atomique

Le but de ce paragraphe est de présenter une méthode classique pour créer des instants de régénération [44].

Supposons qu'il existe une mesure de probabilité μ sur E et une suite $q_i \geq 0$ non identiquement nulle, telles que pour tous $i, j \in E$

$$q_i \mu_j \leq p_{ij}. \tag{7.3}$$

On dit que (q, μ) est un atome pour p . C'est-à-dire que lorsqu'on est à l'état i , la loi du suivant est minorée par μ avec un facteur q_i . Notons que s'il y avait égalité pour tout i , alors $q_i = 1$ et la chaîne de Markov serait une suite de variables indépendantes de loi μ . Plus généralement, on peut remarquer que

$$\begin{aligned} P(X_n = j | X_{n-1} = i) &= q_i \mu_j + (1 - q_i) \frac{p_{ij} - q_i \mu_j}{1 - q_i} \\ &= q_i \mu_j + (1 - q_i) \check{p}_{ij}. \end{aligned} \tag{7.4}$$

Le membre de droite peut être vu comme un mélange de deux lois avec poids q_i et $1 - q_i$. Ceci signifie que pour simuler la chaîne sachant $X_{n-1} = i$, on peut simuler une variable U_n valant 1 ou 0 avec probabilité q_i et $1 - q_i$ et selon cette valeur tirer X_n sous la loi μ ou bien avec la transition \check{p}_{ij} . La paire (X_n, U_n) est markovienne et chaque instant où $U_n = 1$ est un instant de régénération (la démonstration est laissée au lecteur).

En pratique on préfère procéder sans toucher au mode de simulation de X_n . On simule d'abord X_n , puis U_n sous sa loi conditionnelle à $X_n, X_{n-1}, \dots, X_0, U_{n-1}, \dots, U_0$; on trouve avec la formule de Bayes :

$$P(U_n = 1 | X_n = j, X_{n-1} = i, \dots, X_0) = q_i \mu_j / p_{ij} \tag{7.5}$$

ce qui permet de produire U_n . D'où la procédure complète

ALGORITHME

1. Choisir q et μ satisfaisant (7.3)
2. $U_0 = 1$; simuler X_0 sous μ
3. Simuler les X_n , et les U_n selon (7.5)
4. Calculer les instants de renouvellement : $T_q = \inf(n > T_{q-1} : U_n = 1)$, $T_0 = 0$
5. $\hat{\theta} = T_Q^{-1} \sum_{n=1}^{T_Q} g(X_n)$, $\hat{\sigma}(\hat{\theta})^2 = T_Q^{-2} \sum_q (S_q - \tau_q \hat{\theta})^2$.

Remarques : Si le nombre d'états est fini, on trouve des conditions simples sur q pour avoir, avec probabilité 1, un nombre infini d'instants de régénération.

Si l'on pose $I = \{i : q_i > 0\}$ et $J = \{j : \mu_j > 0\}$, on voit que tout point de I peut faire une transition vers tout point de J avec probabilité non-nulle ; ceci montre tout de suite les limitations de la méthode et donne une première idée sur la construction de q et μ .

La situation idéale pour appliquer ce résultat est de connaître un gros ensemble I et une mesure μ tels que pour $i \in I$ la transition se fasse avec une loi toujours proche de $\mu : p_{ij} \simeq \mu_j$. Dans ce cas on prend $q_i = \min_j p_{ij} / \mu_j$ qui sera souvent proche de 1 pour $i \in I$ et les passages dans I réaliseront souvent un renouvellement. Si μ est une masse de Dirac, on est dans la situation du §7.3.1.

Il peut être plus facile d'appliquer cette méthode à la chaîne sous-échantillonnée de matrice de transition P^n (à la limite $n = +\infty$, on a $q = 1$ et $\mu = \pi$).

Cas continu. La relation (7.3) devient $P(x, dy) \geq q(x)\mu(dy)$ pour une fonction $q \geq 0$ non identiquement nulle et μ une probabilité. Le membre de droite de (7.5) devient $q(x)\mu(dy)/P(x, dy)$.

Atome uniforme. Dans le cas où $\varepsilon = \min_i q_i > 0$, il existe des méthodes de simulation d'échantillons indépendants sous π qui pour chaque échantillon demandent de simuler la chaîne sur une longueur d'ordre $1/\varepsilon$ (exercices 7.6.1 et 7.6.2).

Algorithme de Metropolis. On reprend les notations du §5.2. Soit (q, μ) un atome pour P , alors (\check{q}, μ) , avec

$$\check{q}_i = \frac{c\mu_i}{f_i}, \quad c = \min_{\mu_i \neq 0} \frac{f_i q_i}{\mu_i}$$

est un atome pour \check{P} , matrice de transition de (X_n) . La démonstration est laissée en exercice.

Un cas intéressant est celui où $p_{ij} = p_j$ ne dépend que de j ; c'est-à-dire que la chaîne de Markov P produit une suite de v.a.i.i.d. Dans ce cas $q_i = 1$ et $\mu_j = p_j$ et $c = \min f_j/p_j$.

7.4 Échantillonnage parfait : couplage sur le passé

Les algorithmes précédents permettent de calculer des moments de π mais ne permettent pas de simuler *exactement* des variables sous π .

L'échantillonnage parfait est une méthode plus sophistiquée permettant de réaliser cela. Elle est due à J.Propp et D.Wilson, sur une idée remontant à G.Letac, voir [50]. La référence [41] donne un exposé clair de la méthode et de variantes. Elle permet d'échantillonner selon la loi invariante exacte d'une chaîne de la forme

$$X_{n+1} = f(X_n, U_n) \tag{7.6}$$

où les U_n sont i.i.d (comme l'espace d'états est supposé fini, c'est toujours le cas avec U_n uniforme sur $[0, 1]$). Elle est particulièrement adaptée au cas où la chaîne est stochastiquement monotone, c'est-à-dire que f est une fonction croissante de la première variable (pour une certaine relation d'ordre sur l'espace d'états).

7.4.1 Cas général

Supposons la chaîne X_n réalisée pour $n \in \mathbb{Z}$. Si l'on note $U_p^q = (U_p, U_{p+1}, \dots, U_q)$, l'équation (7.6) permet d'exprimer X_0 en fonction de X_{-n} et de U_{-n}^{-1} :

$$\begin{aligned} X_0 &= f(X_{-n}, U_{-n}^{-1}) = \varphi_n(X_{-n}, \omega) \\ \varphi_n(x, \omega) &= \varphi_{n-1}(f(x, U_{-n}), \omega). \end{aligned}$$

La fonction f a été étendue à un nombre d'arguments arbitraire pour opérer plusieurs transitions successives. La fonction φ_n est la même avec une notation plus adéquate pour la suite, où n est le nombre de transitions à effectuer et ω est la suite des U_i . On a donc le schéma suivant pour φ_n

$$X_{-n} = x \xrightarrow{U_{-n}} X_{-n+1} \xrightarrow{U_{-n+1}} X_{-n+2} \dots \xrightarrow{U_{-1}} X_0 = \varphi_n(x, \omega)$$

Très souvent, dans le cas d'une chaîne indécomposable apériodique les applications $x \rightarrow \varphi_n(x, \omega)$ convergent avec probabilité 1 vers une limite aléatoire Z indépendante de x (noter que l'image de cette application ne fait que diminuer d'un n au suivant)¹. Remarquons que ce n'est pas systématiquement le cas, et que cette propriété peut dépendre de la fonction f choisie (même si la loi de la chaîne reste la même), voir l'exercice 7.6.4. On a alors pour tout A et tout x

$$P(Z \in A) = P(\lim_{n \rightarrow \infty} \varphi_n(x, \omega) \in A) = \lim_{n \rightarrow \infty} P(\varphi_n(x, \omega) \in A) = \lim_{n \rightarrow \infty} P_x(X_n \in A) = \pi(A)$$

L'idée est donc de réaliser φ_n pour des n de plus en plus grands jusqu'à ce que son image soit réduite à un point. Ce point suit la loi π . Ceci va donc imposer de stocker à chaque itération les valeurs $\varphi_n(x)$, $x \in E$:

ALGORITHME (Version théorique)

1. Initialisation : Partir de $n = 1$, $\varphi(x) = x$, $x \in E$.
2. Simuler U indépendante du reste.
3. Recalculer φ : $\forall x \in E$, $\varphi(x) \leftarrow \varphi(f(x, U))$.
4. Si $\varphi(E)$ est un singleton, cette valeur suit π et c'est terminé, sinon, $n \leftarrow n + 1$ et retourner au 2.

On pourrait être tenté de propager vers le futur au lieu de reculer vers le passé : Simuler les trajectoires partant de toutes les valeurs possibles de X_0 avec la formule (7.6) et attendre qu'elles se confondent en un seul point, c-à-d le premier n tel que $x \rightarrow f(x, U_1^n)$ soit constant. Malheureusement la distribution de ce point ne sera pas π . L'avantage de la propagation directe est que le nombre de points à simuler diminue au cours du temps. Une méthode essayant de mettre cela à profit est la suivante qui consiste à reculer dans le temps par paquets de longueur croissante, p.ex. 2^k , chaque paquet correspondant à une simulation directe, ceci se faisant au prix de reculer plus que nécessaire (deux fois au pire) :

¹On peut se faire une image de ce qui se passe en travaillant sur le cas d'un AR(1) : $X_{n+1} = aX_n + U_n$, $|a| < 1$; dans ce cas $\varphi_n(x, \omega) = a^n x + a^{n-1}U_{-n} + \dots U_{-1}$. On n'est plus dans un espace d'états fini, mais on voit bien la convergence $\varphi_n(x, \omega)$ vers une limite.

68 CHAPITRE 7. SIMULATION DE PROCÉDÉS : CONVERGENCE

ALGORITHME On pose $\bar{U} = (U_{-n}, \dots, U_{-1})$

1. Initialisation : Partir de $\bar{U} = U_{-1}$
2. Doubler la longueur de \bar{U} en prolongeant cette suite par la gauche avec des variables i.i.d
3. Calculer l'image de E par ces transitions, $f(E, \bar{U})$, en utilisant (7.6). Si c'est un singleton cette valeur suit π et c'est terminé, sinon retourner au 2.

Pour le calcul d'image, on peut réutiliser à chaque étape l'image obtenue à l'étape précédente, puisqu'on compose les applications.

7.4.2 Cas monotone

On suppose que l'on s'est donné une relation d'ordre partiel \preceq sur E avec un plus petit élément x_- et un plus grand x_+ (c-à-d $\forall x \in E, x_- \preceq x \preceq x_+$). Si pour tout $u, x \rightarrow f(x, u)$ est croissante, alors $\varphi_n(x, \omega)$ aussi, et $|\varphi_n(E)| = 1$ si et seulement si $\varphi_n(x_-) = \varphi_n(x_+)$. L'algorithme consiste alors simplement à calculer ces deux états pour des n de plus en plus grands :

ALGORITHME On pose $\bar{U} = (U_{-n}, \dots, U_{-1})$

1. Initialisation : Partir de $\bar{U} = U_{-1}$
2. Doubler la longueur de \bar{U} en prolongeant cette suite par la gauche avec des variables i.i.d
3. Calculer les deux états terminaux $f(x_-, \bar{U})$ et $f(x_+, \bar{U})$ avec condition initiale x_- et x_+ , en utilisant (7.6). S'il coïncident, cette valeur suit π et c'est terminé, sinon retourner au 2.

Ceci s'applique avec succès en traitement d'image, où x_- est l'image noire et x_+ et l'image blanche.

Comme l'a remarqué R. Tweedie, on peut également l'appliquer pour l'algorithme de Metropolis avec tirages indépendants (exercice 7.6.3).

Cette version a également l'avantage de s'étendre sans difficulté au cas continu.

7.5 Réduction de variance

Les méthodes de réduction de variance sont également applicables dans le contexte markovien. L'échantillonnage préférentiel donnera l'algorithme suivant (on note $p_{ij} = p(i, j)$) :

ALGORITHME

1. Simuler les X_n selon la loi markovienne de matrice transition Q
2. Calcul d'une estimée $\hat{\theta}$:

$$\hat{\theta} = d_N N^{-1} \sum_{n=1}^N g(X_n), \quad d_N = \prod_{n=1}^N \frac{p(X_{n-1}, X_n)}{q(X_{n-1}, X_n)}$$

3. Pour estimer la variance, répéter (1) et (2) R fois (cf § 7.1) puis

$$\hat{\theta} = R^{-1} \sum_{r=1}^R \hat{\theta}_r, \quad \hat{\sigma}(\hat{\theta})^2 = R^{-2} \sum_{r=1}^R (\hat{\theta}_r - \hat{\theta})^2.$$

Pour ce qui est de la méthode des variables antithétiques du § 4.4, remarquons que la simulation de X_n peut se faire à partir d'une suite i.i.d de variables uniformes U_n , avec une formule $X_n = f(X_{n-1}, U_n)$, si bien que $\hat{\theta}(X_1, \dots, X_N)$ est de la forme $\hat{\theta} = \varphi(U_1, \dots, U_N)$; en simulant $Y_n = f(Y_{n-1}, 1 - U_n)$ on peut donc fabriquer la variable antithétique $\hat{\theta}(Y_1, \dots, Y_N)$, et moyennner les deux estimateurs obtenus (la valeur initiale est issue d'un échauffement qui est le même pour les deux suites : $X_1 = Y_1$).

7.6 Exercices et compléments

Exercice 7.6.1 Soit une P matrice de transition possédant un atome (ε, μ) :

$$\varepsilon \mu_j \leq p_{ij}$$

où ε est un réel > 0 . Dans la suite, on note $\mathbf{1}$ le vecteur colonne de 1.

1. Montrer que la décomposition (7.4) peut se réécrire : $P = \varepsilon \mathbf{1} \mu + (1 - \varepsilon) \check{P}$.
2. En déduire que la mesure invariante de P s'écrit : $\pi = \sum_{i=0}^{\infty} \varepsilon (1 - \varepsilon)^i \mu \check{P}^i$
3. En déduire la validité de l'algorithme (Murdoch et Green) :
 - (a) Tirer N selon la loi géométrique de paramètre ε et X_0 selon μ
 - (b) Faire tourner la chaîne \check{P} partant de X_0 jusqu'au temps N . X_N suit la loi π .
4. Étendre cette méthode au cas continu.

Exercice 7.6.2 Soit P , ε et μ comme à l'exercice précédent. On propose ici une méthode qui évite de simuler sous \check{P} .

Expliquer comment on peut appliquer la méthode de couplage par le passé en ne simulant la chaîne que pour un seul état initial (s'inspirer de la méthode atomique).

Expliquer pourquoi le nombre de transitions à simuler sera d'ordre ε^{-1} .

Montrer comment ceci s'étend au cas continu.

Exercice 7.6.3 On considère l'algorithme de Metropolis avec $p_{ij} = p_j$ (tirages indépendants). Montrer qu'on a monotonie comme au 7.4.2 avec $i \leq j$ ssi $p_i f_j \leq p_j f_i$ (On écrira la chaîne sous la forme $X_{n+1} = f(X_n, (U_n, V_n))$, avec U_n et V_n uniformes sur $[0, 1]$).

Exercice 7.6.4 Il s'agit de montrer l'importance du choix de f pour la méthode de couplage sur le passé. Considérons la marche aléatoire sur les sommets du triangle de l'exercice 5.5.1.3. On identifiera les trois sommets aux nombres 0, 1 et 2.

1. On décide de paramétrer la chaîne par une addition modulo 3

$$X_{n+1} = X_n + U_n \pmod{3}, \quad P(U_n = -1) = P(U_n = 1) = \frac{1}{2}.$$

Montrer que la convergence de $\varphi_n(x, \omega)$ vers une limite indépendante de x est impossible.

2. On propose une paramétrisation différente avec les transitions suivantes selon la valeur de U

$$\begin{array}{ll}
 & 0 \rightarrow 1 & & 0 \rightarrow 2 \\
 U = -1 : & 1 \rightarrow 0 & & U = 1 : \quad 1 \rightarrow 2 \\
 & 2 \rightarrow 0 & & & 2 \rightarrow 1
 \end{array}$$

Vérifier que la loi de la chaîne est la même. Observer l'effet de deux valeurs différentes de U successives sur la chaîne et en déduire que $\varphi_n(x, \omega)$ converge avec probabilité 1.

Chapitre 8

Optimisation par Monte-Carlo

On discutera dans ce chapitre de deux méthodes : l'approximation stochastique et le recuit simulé. L'approximation stochastique étudie la résolution d'un problème d'optimisation sur des données bruitées. En revanche, dans le recuit simulé le problème est déterministe et bien posé, minimiser une fonction V sur un ensemble E , mais l'algorithme de recherche est randomisé afin de bien parcourir l'ensemble E .

Les deux méthodes sont donc fondamentalement différentes : dans un cas l'aléatoire vient des données tandis que dans l'autre il est introduit dans l'algorithme.

8.1 Approximation stochastique

L'approximation stochastique est un domaine difficile où les résultats généraux nécessitent des jeux d'hypothèses compliqués et où les preuves sont généralement assez ardues ; voir cependant [18] pour une situation raisonnablement générale traitée complètement. Nous renvoyons pour les études théoriques aux traités généraux [1, 42] et à [1, 52] pour ce qui est des applications. Nous ne nous livrons ici qu'à une discussion assez informelle.

8.1.1 Deux exemples simples

Gradient bruité. La méthode de gradient pour minimiser une fonction $V(\theta)$ consiste à réaliser l'algorithme suivant

$$\theta_n = \theta_{n-1} - \gamma \nabla V(\theta_{n-1})$$

qui pour γ assez petit converge assez vite vers un minimum de V (sous certaines hypothèses....).

Supposons maintenant que l'on ne puisse mesurer le gradient qu'avec un certain bruit. C'est-à-dire que l'on peut faire des mesures où l'on observe $\nabla V(\theta) + \xi$ où θ est au choix mais ξ est une v.a de moyenne nulle. Ceci arrive si par exemple le calcul de V passe par des espérances que l'on estime par simulation.

L'algorithme stochastique consiste à s'inspirer de la méthode du gradient de la façon suivante :

$$\theta_n = \theta_{n-1} - \gamma_n (\nabla V(\theta_{n-1}) + \xi_n).$$

Des calculs théoriques conduisent à deux observations importantes : La première est que le bon choix de γ_n est de prendre ce gain d'ordre $1/n$ (p.ex. $\gamma_n = \frac{a}{n+b}$). La seconde est que l'idée d'essayer de faire des expériences successives avec le même θ pour estimer $\nabla V(\theta)$ en moyennant, et ensuite

utiliser l'algorithme déterministe est mauvaise, c'est-à-dire conduira à une variance d'estimation supérieure pour un nombre fixé d'expériences.

Un algorithme stochastique peut donc être vu comme un algorithme déterministe où l'on a oublié le signe d'espérance et modifié le gain en conséquence.

Moyenne empirique. Un cas particulier de gradient stochastique est $V(\theta) = E[(Y - \theta)^2]/2$, où Y est une v.a. dont on peut observer des réalisations ; ici V atteint son minimum en $\theta^* = E[Y]$. On ne peut pas observer $\nabla V(\theta) = \theta - E[Y]$ mais des réalisations de Y , et donc de $\theta - Y$ dont l'espérance est bien $\nabla V(\theta)$; d'où l'algorithme

$$\theta_n = \theta_{n-1} - \gamma_n(\theta_{n-1} - Y_n).$$

On observe facilement que si $\gamma_n = 1/n$, et $\theta_0 = 0$ alors $\theta_n = (Y_1 + \dots + Y_n)/n$.

Dosage : un algorithme de Robbins-Monro. Soit un paramètre réel θ représentant un dosage entre différents produits. Pour chaque θ , on peut faire une expérience dont le résultat est un nombre réel Y (quantité de chaleur produite,...) et l'on cherche à résoudre l'équation

$$E_\theta[Y] = \alpha. \tag{8.1}$$

Là encore, on observe Y alors qu'il faudrait $E_\theta[Y]$. On suppose que $E_\theta[Y]$ est une fonction continue strictement croissante de θ . L'algorithme de Robbins-Monro consiste à faire une série d'expériences avec un choix de paramètre dépendant à chaque fois des résultats précédents :

$$\theta_n = \theta_{n-1} + \gamma_n(\alpha - Y_n).$$

Ici Y_n est le résultat de l'expérience faite avec θ_{n-1} . L'algorithme augmente donc θ si Y est trop faible et le fait diminuer sinon.

On montre alors, sous des hypothèses raisonnables, la convergence de θ_n vers la solution θ^* de (8.1). Cet algorithme est le pendant stochastique de l'algorithme déterministe

$$\theta_n = \theta_{n-1} + \gamma E[\alpha - Y_n].$$

Dans ce dernier algorithme, on sait qu'on a intérêt à prendre γ constant.

8.1.2 Forme générale et la trajectoire moyenne

Dans toute la suite, le signe ∇ désignera le gradient par rapport à la variable θ .

Le problème général est le suivant :

Soit un processus stationnaire Y_n et $H(y, \theta)$ une fonction paramétrée par le vecteur θ . On cherche à résoudre en θ l'équation $E[H(Y_n, \theta)] = 0$. À chaque instant n , on a à sa disposition une observation de $H(Y_n, \theta)$ en un point θ de son choix.

L'algorithme stochastique associé est :

$$\theta_n = \theta_{n-1} + \gamma_n H(Y_n, \theta_{n-1}).$$

Une situation typique est le cas du **gradient stochastique** : On cherche à minimiser en θ un coût moyen $V(\theta) = E[Q(Y_n, \theta)]$, et dans ce cas $H(Y_n, \theta) = -\nabla Q(Y_n, \theta)$.

On montre que pour $\gamma_n = c/n^\alpha$, $1/2 < \alpha \leq 1$, et c pas trop grand si $\alpha = 1$, θ_n convergera vers la solution θ^* du problème, et il y a normalité asymptotique de $(\theta_n - \theta^*)/\sqrt{\gamma_n}$. Ceci pousse à prendre α proche de 1, mais le choix de c devient critique.

Accélération. Un meilleur choix (en terme de variance d'estimation) est de prendre un gain matriciel

$$\theta_n = \theta_{n-1} + \Gamma_n^{-1} H(Y_n, \theta_{n-1}) \quad (8.2)$$

$$\Gamma_n = - \sum_{k=1}^n \nabla H(Y_k, \theta_{k-1}) \quad (8.3)$$

ou mieux $\Gamma_n = -nE[\nabla H(Y_n, \theta)]$, si cette quantité est connue ; en effet *si l'algorithme ci-dessus est stable*, il conduit à la meilleure variance asymptotique possible (borne de Cramér-Rao). Noter qu'on retrouve, dans le cas du gradient stochastique, un algorithme du type Newton. Si cette méthode ne peut être employée (comme dans le cas du dosage), on peut avoir intérêt à utiliser l'algorithme avec moyennisation, dit "de Ruppert-Polyak" [49] :

$$\begin{aligned} \theta_n &= \theta_{n-1} + \gamma_n H(Y_n, \theta_{n-1}) \\ \bar{\theta}_n &= \frac{1}{n} \sum_{k=1}^n \theta_k \end{aligned}$$

où un bon choix de γ_n est d'ordre $n^{-2/3}$; $\sqrt{n}(\bar{\theta}_n - \theta^*)$ converge en loi vers une gaussienne de variance optimale. En pratique, on laissera un temps d'échauffement avant de commencer la moyennisation (deuxième équation) car les premiers θ_n sont souvent assez mauvais.

Convergence et trajectoire moyenne. De manière générale, la convergence n'a lieu que si les trajectoires de

$$\frac{d\bar{\theta}(t)}{dt} = h(\bar{\theta}(t)), \quad h(\theta) = E[H(Y, \theta)], \quad (8.4)$$

convergent pour tout $\bar{\theta}(0)$. La convergence de ces trajectoires ne pose généralement pas trop de problème dans le cas d'un algorithme de gradient ($h(\theta) = -\nabla V(\theta)$), cf exercice 8.3.1. Le champ h est appelé champ moyen et ses trajectoires sont appelées trajectoires moyennes car elles représentent le comportement moyen de la suite θ_n , sous des hypothèses adéquates ; on a l'approximation par des morceaux de trajectoire moyenne :

$$\theta_{n+p} \simeq \bar{\theta}(\gamma_n + \dots + \gamma_{n+p}), \quad \bar{\theta}(0) = \theta_n.$$

Sans entrer dans les détails, disons que cette approximation est surtout valide pour n grand. L'heuristique cachée derrière cette idée est :

$$\theta_{n+p} = \theta_n + \sum_{k=1}^p \gamma_{n+k} H(Y_{n+k}, \theta_{n+k-1}) \simeq \theta_n + \sum_{k=1}^p \gamma_{n+k} h(\theta_{n+k-1})$$

à un terme de moyenne nulle près, petit si n est assez grand (sa variance est d'ordre $\sum_{i=n}^{n+p} \gamma_i^2 < \infty$). L'étude rigoureuse est bien entendu difficile ; les conditions sur γ_n sont $\sum \gamma_n = +\infty$ et $\sum \gamma_i^2 < \infty$.

8.1.3 La régression linéaire

On observe des paires $(y_n, x_n) \in \mathbb{R} \times \mathbb{R}^d$ et l'on cherche le vecteur θ^* minimisant l'erreur de prédiction linéaire de y_n avec x_n :

$$\theta^* = \arg \min E[(y_n - x_n \theta^*)^2].$$

On supposera que la suite (y_n, x_n) est i.i.d. Un premier algorithme est l'algorithme de gradient :

$$\theta_n = \theta_{n-1} + \gamma_n x_n^T (y_n - x_n \theta_{n-1}) \quad (8.5)$$

qui minimise $E[(y_n - x_n^T \theta)^2]$, cf l'exercice 8.3.3. Si l'on essaye de l'améliorer selon l'équation (8.2), on obtient l'algorithme RLS (Recursive Least Squares) :

$$\theta_n = \theta_{n-1} + \frac{1}{n} R_n^{-1} x_n^T (y_n - x_n \theta_{n-1}), \quad R_n = \frac{1}{n} \sum_{i=1}^n x_i^T x_i, \quad (8.6)$$

qui doit être modifié pour les n petits, en raison de la non-inversibilité de R_n ; noter que, si l'on fait abstraction de cette modification, et si $\theta_0 = 0$, θ_n coïncide avec

$$\theta_n = (n R_n)^{-1} \sum_{i=1}^n x_i y_i$$

(multiplier (8.6) par $n R_n$) qui est exactement l'estimée aux moindres carrés (le θ qui minimise $\sum_i (y_i - x_i \theta)^2$). On préfère souvent exprimer le calcul de R_n de façon récursive avec un calcul direct de $\Gamma_n = (\sum_{i=1}^n x_i^T x_i)^{-1}$

$$\theta_n = \theta_{n-1} + \Gamma_n x_n^T (y_n - x_n \theta_{n-1}), \quad \Gamma_n = \Gamma_{n-1} - \frac{\Gamma_{n-1} x_n^T x_n \Gamma_{n-1}}{1 + x_n \Gamma_{n-1} x_n^T}.$$

Exemple : Annulation de bruit. Afin de permettre aux conducteurs de téléphoner au volant de leur voiture on se propose de fixer un micro au volant. Ce micro enregistre la somme du signal de parole et du bruit ambiant

$$y_n = p_n + b_n.$$

Un second micro placé au plafond, derrière le conducteur, enregistre le bruit seul. Il reçoit un signal r_n qu'on espère proche de b_n . Le signal transmis au correspondant sera de la forme

$$t_n = y_n - \theta r_n.$$

Le meilleur θ est celui qui rendra t_n aussi proche que possible de p_n et donc qui minimisera $E[(t_n - p_n)^2]$. Comme p_n est indépendant de b_n et r_n , on a en vertu du théorème de Pythagore

$$E[t_n^2] = E[(t_n - p_n + p_n)^2] = E[(t_n - p_n)^2] + E[p_n^2].$$

Le meilleur θ minimise donc l'énergie de t_n , c'est la corrélation entre r et y .

De manière plus générale on pose $x_n = (r_n, \dots, r_{n-d+1})$, $\theta \in \mathbb{R}^d$ et l'on minimise $E[(y_n - x_n \theta)^2]$. C'est un problème de régression linéaire pour lequel on peut utiliser le RLS.

8.1.4 Exemple : Optimisation d'une chaîne de Markov contrôlée

On se donne une chaîne de Markov X_n dont la dynamique s'écrit

$$X_n = f(X_{n-1}, \theta, \xi_n) \quad (8.7)$$

où θ est un paramètre et ξ_n une suite i.i.d de distribution connue. On cherche la valeur de θ qui minimise un coût moyen $E[Q(X_n)]$.

Si l'on considère X_n comme une fonction de θ , X_0 et des ξ_n , on peut dire que l'on cherche à minimiser $E[Q(X_n(\theta))]$. Pour construire l'algorithme stochastique correspondant, il va falloir dériver X_n par rapport à θ . Si l'on note X'_n cette dérivée, on a

$$X'_n = f_\theta(X_{n-1}, \theta, \xi_n) + f_X(X_{n-1}, \theta, \xi_n)X'_{n-1} \quad (8.8)$$

d'où l'algorithme complet

$$\begin{aligned} \theta_n &= \theta_{n-1} - \gamma_n Q_\theta(X_{n-1})X'_{n-1} \\ X_n &= f(X_{n-1}, \theta_{n-1}, \xi_n) \\ X'_n &= f_\theta(X_{n-1}, \theta_{n-1}, \xi_n) + f_X(X_{n-1}, \theta_{n-1}, \xi_n)X'_{n-1} \end{aligned}$$

où l'indice indique la variable par rapport à laquelle on dérive. Noter que dans la dernière équation X'_n n'est qu'une approximation de la dérivée, puisque θ évolue ; cette approximation est rendue valide par le fait que θ change de plus en plus lentement : l'étude du champ moyen sort du cadre établi jusqu'à présent et relève du § 8.1.7.

Tout ceci n'est bien entendu valable que si l'équation (8.8) donne un système stable, c'est-à-dire que X_n est effectivement une fonction régulière de θ . C'est par exemple le cas si $f(X, \theta, \xi) = X\theta + \xi$, pour $|\theta| < 1$ car alors $X_n = \xi_n + \theta\xi_{n-1} + \theta^2\xi_{n-2} + \dots$.

8.1.5 Exemple : Données manquantes

Reprenons la situation du § 5.4.2. On dispose d'une famille paramétrique de lois candidates p_θ pour un ensemble de données X partiellement observées. On note $X = (Y, Z)$, où Y est la partie observée et Z la partie manquante. L'algorithme de maximum de vraisemblance consiste à maximiser la probabilité des données observées :

$$\hat{\theta} = \arg \max_{\theta} p_\theta(Y), \quad p_\theta(Y) = \int p_\theta(Y, z) dz. \quad (8.9)$$

Proposons l'algorithme stochastique suivant :

$$\theta_{n+1} = \theta_n + \gamma_{n+1} \frac{d \log p_{\theta_n}(Y, Z_n)}{d\theta}$$

où la variable Z_n est simulée à chaque étape selon la loi $p_{\theta_n}(z|Y)$, et γ_n est le gain. Le champ moyen de l'algorithme est

$$h(\theta) = \int \frac{d \log p_\theta(Y, z)}{d\theta} p_\theta(z|Y) dz = \int \frac{\nabla p_\theta(Y, z)}{p_\theta(Y, z)} \frac{p_\theta(Y, z)}{p_\theta(Y)} dz = \frac{d}{d\theta} \log(p_\theta(Y)).$$

On a donc un algorithme de gradient dont la limite sera un maximum de $p_\theta(Y)$, si toutes les hypothèses techniques non mentionnées ici sont vérifiées.

8.1.6 Exemple : Algorithme de Kiefer-Wolfowitz

Supposons que l'on cherche à régler un paramètre θ d'un robot de sorte à ce qu'il atteigne au mieux sa cible dans des configurations différentes (de bruit, ou de positionnement de la cible...). On est alors (en général) dans la situation du § 8.1.2 sauf que l'on ne sait pas calculer le gradient de la fonction que l'on veut optimiser. En particulier, contrairement à la situation Robbins-Monro, on ne sait pas du tout dans quelle direction modifier le paramètre.

Une expérience avec un certain θ et certaines conditions permettra d'évaluer une performance (distance à la cible...) $Q(Y, \theta)$ où la v.a. Y , a priori non observée, représente ici l'aléa (la position de la cible, les bruits,...). On cherche à minimiser $E[Q(Y, \theta)]$. Une solution est de faire les expériences par paire de sorte à estimer le gradient ; l'algorithme devient

$$\theta_n = \theta_{n-1} + \gamma_n \frac{Q(Y_n, \theta_{n-1} + c_n) - Q(Y'_n, \theta_{n-1} - c_n)}{2c_n}$$

où l'on a en plus les conditions $\sum (\gamma_n/c_n)^2 < \infty$ et $c_n \rightarrow 0$ qui vont garantir la convergence [52] ; un bon choix est, $\gamma_n = \gamma_1/n$, $c_n \simeq \log(n)$, avec normalité asymptotique de $c_n \sqrt{n}(\hat{\theta}_n - \theta^*)$ [18].

Si θ est vectoriel, on tire à chaque fois un vecteur aléatoire Δ_n de ± 1 , et le quotient ci-dessus est remplacé par $(2c_n)^{-1} \Delta_n (Q(Y_n, \theta_{n-1} + c_n \Delta_n) - Q(Y'_n, \theta_{n-1} - c_n \Delta_n))$.

8.1.7 Cadre général à dynamique markovienne

Le cadre du §8.1.2 est trop restrictif car il ne contient pas en particulier les algorithmes de Robbins-Monro. On considère donc plus généralement des algorithmes de la forme :

$$\theta_n = \theta_{n-1} + \gamma_n H(Y_n, \theta_{n-1}).$$

où la loi de Y_n sachant le passé est donnée par la probabilité de transition $P_{\theta_{n-1}}(Y_{n-1}, dy)$ (une famille de probabilités de transition $P_{\theta}(x, dy)$ est donc donnée à l'avance).

C'est typiquement le cas de l'algorithme du §8.1.4, avec $Y_n = (X_n, X'_n)$.

On peut montrer que sous certaines hypothèses (en particulier la stabilité de la chaîne (θ_n, Y_n)) cet algorithme convergera vers une solution de $E_{\theta}[H(Y, \theta)] = 0$ où l'espérance est prise sous la mesure invariante de P_{θ} . L'hypothèse essentielle est toujours la convergence de la trajectoire moyenne

$$\frac{d\theta_t}{dt} = h(\theta_t), \quad h(\theta) = E_{\theta}[H(Y, \theta)].$$

S'il y a forte dépendance de la loi de la chaîne par rapport à θ , l'algorithme (8.2) perd de son sens et il est difficile de trouver une matrice de gain adéquate. Toutefois la méthode de Ruppert-Polyak s'applique toujours.

8.1.8 Algorithmes de poursuite

Reprenons l'exemple de la régression, mais on va supposer maintenant que θ^* varie lentement au cours du temps car les processus ne sont pas stationnaires. Supposons que l'on ait l'équation

$$y_n = x_n \theta_n^* + u_n$$

avec (x_n, u_n) i.i.d. Les changements de θ^* font que (y_n, x_n) n'est plus stationnaire. L'idée des algorithmes adaptatifs est de ne plus prendre γ_n tendant vers 0, car on ne cherche plus à avoir convergence vers une valeur fixe, mais de prendre γ constant :

$$\theta_n = \theta_{n-1} + \gamma x_n^T (y_n - x_n \theta_{n-1})$$

ou encore en RLS :

$$\begin{aligned} \theta_n &= \theta_{n-1} + \gamma R_n^{-1} x_n^T (y_n - x_n \theta_{n-1}) \\ R_n &= \frac{1}{n} \sum_{i=1}^n x_i^T x_i. \end{aligned}$$

Dans cette dernière équation, si x_n a une dynamique lentement variable, il vaut mieux prendre

$$R_n = R_{n-1} - \gamma (R_{n-1} - x_n^T x_n),$$

cf [34]. Le choix de la meilleure valeur de γ est difficile ; le mieux est de la déterminer par simulation.

8.2 Recuit simulé

Soit E un ensemble fini et V une fonction réelle définie sur E . On cherche le minimum de V . L'exemple typique d'application du recuit simulé est le problème du voyageur de commerce : trouver un chemin fermé le plus court possible qui passe par un ensemble de villes données ; E est donc l'ensemble des tels chemins. L'idée est de simuler les distributions de probabilité suivante sur E

$$\pi_\beta(i) = Z_\beta^{-1} e^{-\beta V(i)}$$

où β est un paramètre et Z_β est la constante de normalisation. On voit en effet facilement que lorsque β tend vers $+\infty$, π_β tend vers la mesure uniforme sur les minima de V .

Si β est fixe, on est dans la situation d'utilisation de l'algorithme de Metropolis avec $f_i = e^{-\beta V(i)}$. Il suffit de se donner une matrice de transition P sur E ; dans le cas du voyageur de commerce, on choisit typiquement pour p_{ij} la mesure uniforme sur tous les chemins j qui se déduisent de i par l'interversion de deux villes (d'autres versions sont possibles...). En simulant cette loi avec β grand, on visitera sans doute assez souvent de bonnes solutions ; malheureusement, plus β est grand, plus les transitoires sont longs (la chaîne de Markov a une dynamique très faible).

L'idée du recuit simulé est de faire augmenter β au cours de l'algorithme de Metropolis ; il faut donc donner une suite croissante β_n qui sera utilisée à la n -ième étape, avec l'idée que la loi de la variable simulée X_n sera toujours approximativement π_{β_n} . D'où l'algorithme, où X^* désigne la meilleure solution trouvée jusqu'au temps n :

ALGORITHME Tirage de X_n

1. Tirer Y_n parmi les voisins de X_{n-1} selon une loi prédéterminée (généralement uniforme)
2. Faire aléatoirement $X_n = Y_n$ ou $X_n = X_{n-1}$ avec probabilité q et $1 - q$:
$$q = e^{-\beta_n(V(Y_n) - V(X_{n-1}))_+}$$
3. Enregistrement du meilleur : si $V(X_n) < V(X_{n-1}^*)$ faire $X_n^* = X_n$.

On voit donc que la transition est acceptée sûrement si $V(Y_n) \leq V(X_{n-1})$, et éventuellement sinon. On accepte donc parfois des transitions qui font augmenter V , ce qui permet d'éviter les minima locaux.

La difficulté est que si β augmente trop vite, on "gèle" l'algorithme et l'on ne poursuit plus π_β : l'algorithme se calera dans une région où V fait une cuvette. La règle théorique est de prendre à chaque instant $\beta_n = C / \log(n)$, pour un C assez grand¹ [17] ; cependant l'application brute de cette formule, avec le C recommandé, donne en général des algorithmes extrêmement longs. Cela vient en partie du fait que les études de convergence du recuit simulé se soucient de la convergence en loi de X_n et non pas de celle de X_n^* . La convergence de ce dernier ne requiert pas que β tende vers 0.

La convergence est donc a priori très lente et l'efficacité (observée) de la méthode dans des situations concrètes est une affaire de praticien.

Il est souvent assez difficile de trouver une vitesse de croissance de β adéquate. Il faut remarquer qu'un β un peu trop grand aura pour effet de transformer l'algorithme en algorithme de descente

¹Pour tout x , on définit le coût d'un chemin partant de x et menant au minimum de V comme le maximum de $(V(y) - V(x))_+$ sur tous les points y de ce chemin ; ce coût est nul si le chemin descend. On définit ensuite $c(x)$ comme le coût minimum de tout les chemins menant au minimum de V . $c(x)$ est nul si l'on peut de x au minimum en descendant. C est alors le maximum de $c(x)$ pour tous les x .

pure et un β trop petit fera accepter quasiment toutes les transitions. Il y a donc un juste milieu à adopter, qui ne laisse en réalité pas une très grande marge ; on trouve en effet par calcul, que si les $V(j) - V(i)$ sont du même ordre, disons δ , pour j voisin de i , et si q est le nombre de voisins de chaque point :

$$\beta_{min} \simeq \delta^{-1}, \quad \beta_{max} \simeq \delta^{-1} \log(q)$$

au sens où si $\beta < \beta_{min}$, on accepte en gros toute transition une fois sur deux, et si $\beta > \beta_{max}$ on attend d'avoir visité tous les voisins avant d'accepter une transition montante. Il est donc raisonnable d'essayer de viser entre ces deux valeurs.

On peut par exemple estimer un β_n raisonnable de manière adaptative par :

$$\begin{aligned} \delta_n &= \delta_{n-1} + \frac{1}{n} |V(Y_n) - V(X_{n-1})| \quad \text{ou bien} \quad \delta_{n-1} + \frac{1}{n} (V(Y_n) - V(X_{n-1}))_+ \\ \beta_n &= c \delta_n^{-1} \log(q) \end{aligned}$$

où q est le nombre de voisins d'un point et c une constante inférieure à 1.

Une solution alors efficace est d'utiliser une règle du type de la précédente et de faire varier la taille des voisinages : au début on prend des voisinages de grande taille pour bien visiter l'espace et l'on réduit peu à peu ; dans le cas du voyageur de commerce, on pourra ainsi se restreindre à des interversions de villes de plus en plus proches au fur et à mesure de l'avancée de l'algorithme.

8.3 Exercices

Exercice 8.3.1 On suit les notations du § 8.1.2. Montrer que si $V(\theta)$ est strictement convexe et tend vers l'infini, alors l'équation différentielle $\dot{\theta} = -\nabla V(\theta)$ converge vers le minimum de V (on montrera que $V(\theta_t)$ tend vers une limite, et que $\int |\nabla V(\theta_t)|^2 dt < \infty$; on en déduira une contradiction si θ_t ne converge pas vers le minimum de V).

Exercice 8.3.2 Soit X_n une suite de v.a.i.i.d. et $0 < \alpha < 1$. Quel est le champ moyen de l'algorithme

$$\theta_n = \theta_{n-1} - \gamma_n (1_{X_n < \theta_{n-1}} - \alpha).$$

Vérifier que l'algorithme est un algorithme de gradient. Vers quelle valeur devrait converger l'algorithme ?

Exercice 8.3.3 Vérifier que (8.5) est l'algorithme de gradient pour minimiser $E[(y_n - x^T \theta)^2]$ et construire l'algorithme correspondant à la minimisation de $E[|y_n - x_n^T \theta|]$.

Exercice 8.3.4 Une machine à sous a deux bras. La probabilité de gain avec le bras A (resp. B) est inconnue et vaut p_A (resp. p_B). Un joueur veut estimer quel bras a la plus grande probabilité de gain sans perdre trop d'argent. Sa stratégie est la suivante : au temps n , il choisit la machine A (resp. B) avec probabilité θ_{n-1} (resp. $1 - \theta_{n-1}$) et

$$\theta_n = \begin{cases} \theta_{n-1} & \text{s'il a perdu} \\ \theta_{n-1} + \gamma_n (1 - \theta_{n-1}) & \text{s'il a gagné sur } A \\ \theta_{n-1} - \gamma_n \theta_{n-1} & \text{s'il a gagné sur } B \end{cases}$$

Il espère que θ_n convergera vers 1 (resp. 0) si A (resp. B) est la meilleure machine.

Calculer le champ moyen. Vérifier que si $\theta^* = 0$ (resp. 1) alors $E[\theta_n]$ est décroissant (croissant).

Chapitre 9

Simulation d'équations différentielles stochastiques

9.1 Introduction

Il s'agit d'approcher numériquement la solution de l'équation différentielle stochastique

$$dx_t = b(x_t)dt + \sigma(x_t)dw_t \quad (9.1)$$

partant d'un point donné (pour un exposé introductif au EDS, voir [45]). On oublie la dépendance éventuelle de b et σ par rapport à t , qui ne jouerait qu'un rôle mineur dans la suite. Pour simplifier les écritures, on fera également comme si x était scalaire, ceci n'intervenant pas réellement dans les arguments présentés.

Les schémas de simulation sont des extensions de ceux existant pour le cas déterministe $\sigma = 0$, et les principes de démonstration de convergence sont très proches, bien qu'il mènent ici à des calculs bien plus compliqués. C'est pourquoi nous commencerons en première approche par un aperçu du cas déterministe.

9.2 Le cas déterministe : $\sigma = 0$

L'équation est :

$$dx_t = b(x_t)dt, \quad x_0 = x_0, \quad 0 \leq t \leq T < +\infty.$$

On va considérer différents schémas de simulation en N instants espacés de $h = T/N$. On notera y_n l'approximation de x_{nh} .

Schéma d'Euler d'ordre 1. Il s'agit de la discrétisation la plus naturelle de l'équation :

$$y_{n+1} = y_n + b(y_n)h.$$

Schéma d'Euler d'ordre 2. Il correspond à une approximation plus fine basée sur la formule de Taylor appliquée à $f(t) = x_t$:

$$x_{(n+1)h} = x_{nh} + hb(x_{nh}) + \frac{h^2}{2}b'(x_{nh})b(x_{nh}) + O(h^3)$$

d'où

$$y_{n+1} = y_n + b(y_n)h + \frac{h^2}{2}b'(y_n)b(y_n).$$

Schéma de Runge et Kutta. Il s'agit de l'équation suivante qui clairement suit la précédente jusqu'à l'ordre 2, sans nécessiter l'intervention de la dérivée de b :

$$y_{n+1} = y_n + b(y_n + hb(y_n)/2).$$

On voit qu'en argument de $b(\cdot)$ est le point médian calculé avec l'approximation précédente. On peut donc voir cette méthode comme une méthode des trapèzes approchée basée sur

$$y_{n+1} - y_n = \int_{nh}^{(n+1)h} b(x_t)dt \simeq hb(x_{t+h/2}) \simeq hb(y_n + hb(y_n)/2).$$

Schéma implicite. Cette équation suit également les précédentes jusqu'à l'ordre 2, elle force la résolution d'un système :

$$y_{n+1} = y_n + \frac{h}{2}(b(y_{n+1}) + b(y_n))$$

Ce schéma est préféré quand se posent des problèmes de stabilité.

Convergence. On mesure la vitesse de convergence à l'aune du plus grand exposant α tel que

$$\sup_{0 \leq t \leq T} |y_{[t/h]} - x_t| = O(h^\alpha).$$

On peut trouver des schémas d'ordre arbitraire. On a les vitesses d'approximation suivantes :

Schéma	α	Δy	Condition
Euler d'ordre 1	1	$h b$	$b \in C^1$
Euler d'ordre 2	2	$hb + h^2bb'/2$	$b \in C^2$
R-K d'ordre 2	2	$h b(y + hb/2)$	$b \in C^2$
Implicite	2	$h(b + b_+)/2$	$b \in C^2$

TAB. 9.1 – Vitesse de convergence des schéma et conditions de régularité sur b . On note b pour $b(y_n)$ et b_+ pour $b(y_{n+1})$.

La démonstration peut se faire à base de formules de Taylor. Par exemple le schéma d'ordre 1 :

$$x_{(n+1)h} = x_{nh} + hb(x_{nh}) + O(h^2) \quad \text{car } b \in C^1$$

et donc $z_n = y_n - x_{nh}$ satisfait

$$\begin{aligned} |z_{n+1}| &\leq |z_n|(1 + h\beta) + Ch^2, \quad \beta = \|b'\|_\infty \\ &\leq |z_n|e^{h\beta} + OCh^2 \\ |z_{n+1}|e^{-(n+1)h\beta} &\leq |z_n|e^{-nh\beta} + Ch^2e^{-nh\beta} \\ &\leq Ch^2 \sum_{k=0}^n e^{-kh\beta} \\ &\leq Ch^2(1 - e^{-h\beta})^{-1}. \end{aligned}$$

Par conséquent $|z_n| \leq C'h$ pour un C' assez grand. Il s'ensuit que $|y_n - x_{nh}| \leq C'h$ et donc que $|y_{t/h} - x_t| \leq C'h$.

9.3 Les deux mesures de vitesse de convergence

Si $y_n \simeq x_{nh}$ est une approximation de la trajectoire de (9.1), La vitesse de convergence forte est donnée par le plus grand γ tel que :

$$E[|x_T - y_N|] = O(h^\gamma)$$

et la vitesse de convergence faible concerne la convergence des moments et est donnée par le plus grand β tel que :

$$|E[f(x_T)] - E[f(y_N)]| = O(h^\beta) \quad (9.2)$$

pour tout polynôme f (ou pour f assez régulière à support compact). Cette équation restera généralement valide pour les fonction assez régulières.

Pour chaque méthode, des conditions de régularité spécifiques devront être imposées sur les fonctions b et σ pour que la vitesse propre à la méthode soit effectivement atteinte.

9.4 Schémas d'intégration pour les eds

9.4.1 Construction du schéma

Le schéma d'Euler consiste à considérer les intégrands constants. On approxime donc

$$x_{t+h} = x_t + \int_t^{t+h} b(x_s) ds + \int_t^{t+h} \sigma(x_s) dw_s$$

par

$$x_{t+h} \simeq x_t + hb(x_t) + \sigma(x_t)(w_{nh+h} - w_{nh}).$$

Comme les variables $w_{nh+h} - w_{nh}$ sont normales indépendantes de variance h , on arrive au schéma

$$y_{n+1} = y_n + b(y_n)h + \sigma(y_n)\sqrt{h} \delta_n$$

où δ_n sont des variables i.i.d $\mathcal{N}(0, 1)$. Si b et σ dépendent de t , il faut bien entendu mettre $b(nh, y_n)$ et $\sigma(nh, y_n)$.

9.4.2 Schéma de Milstein

En raison du terme brownien, d'ordre \sqrt{h} , le schéma d'Euler n'a un ordre fort que de 0,5. Une façon de corriger cela est d'avancer plus loin dans le développement de sorte à arriver jusqu'à l'ordre h . On est naturellement conduit à une approximation plus fine de x_s dans la deuxième intégrale :

$$\begin{aligned} x_{t+h} &\simeq x_t + hb(x_t) + \int_t^{t+h} \sigma(x_t + \sigma(x_t)(w_s - w_t)) dw_s \\ &\simeq x_t + hb(x_t) + \int_t^{t+h} \sigma(x_t) + \sigma'(x_t)\sigma(x_t)(w_s - w_t) dw_s \\ &\simeq x_t + hb(x_t) + \sigma(x_t)(w_{t+h} - w_t) + \sigma'(x_t)\sigma(x_t) \int_t^{t+h} (w_s - w_t) dw_s \\ &= x_t + hb(x_t) + \sigma(x_t)(w_{t+h} - w_t) + \sigma'(x_t)\sigma(x_t)((w_{t+h} - w_t)^2 - h)/2 \end{aligned}$$

(car $\int_0^t w_s dw_s = \frac{w_t^2}{2} - \frac{t^2}{2}$) d'où

$$y_{n+1} = y_n + b(y_n)h + \sigma(y_n)\sqrt{h}\delta_n + \sigma(y_n)\sigma'(y_n)(\delta_n^2 - 1)h/2$$

Noter que ce schéma se réécrit $y_+ = y + b_s(y)h + (\sigma(y) + \sigma(y + \sigma(y)\sqrt{h}\delta))\sqrt{h}\delta/2$, où $b_s = b - \sigma\sigma'/2$ est le drift intervenant dans l'écriture en Stratonovitch de l'équation.

9.4.3 Convergence forte du schéma d'Euler

On suppose dans la suite que b' , b'' et σ' sont bornées. Rappelons que ceci implique que

$$E[|x_t - x_s|^n] \leq C_n(t-s)^{n/2}, \quad 0 < s < t < T$$

pour un certain C_n dépendant de n . On a

$$x_{t+h} = x_t + hb(x_t) + \sigma(x_t)(w_{t+h} - w_t) + v_{t+h},$$

$$v_{t+h} = \int_t^{t+h} (b(x_s) - b(x_t))ds + \int_t^{t+h} (\sigma(x_s) - \sigma(x_t))dw_s.$$

On montre en utilisant le calcul stochastique habituel que pour une constante $C > 0$

$$\begin{aligned} E[|v_{t+h}|^2] &\leq Ch^2, \\ |E[v_{t+h}|\mathcal{F}_t]| &\leq Ch^2 \quad \text{avec probabilité 1} \end{aligned}$$

(pour le deuxième point, calculer $b(x_s) - b(x_t)$ avec la formule d'Itô).

L'erreur d'approximation $z_n = y_n - x_{nh}$ satisfait,

$$z_{n+1} = z_n + hz_n b' + \sqrt{h}\delta_n z_n \sigma' + v_{nh+h}$$

où b' et σ' sont les dérivées de b et σ en des points adéquats. Si l'on prend l'espérance du carré de cette expression, en exploitant l'indépendance de δ_n et z_n , on obtient facilement :

$$E[z_{n+1}^2] \leq E[z_n^2] + C'hE[z_n^2] + C'h^2$$

pour un C' assez grand. Ceci implique que $E[z_n^2] \leq C''nh^2$. Donc $\gamma \geq 1/2$. On montre que γ n'est pas supérieur.

9.4.4 Convergence faible du schéma d'Euler

Le calcul de l'ordre faible est plus compliqué. Introduisons le générateur de la diffusion A et l'opérateur de transition P_t :

$$\begin{aligned} Af(x) &= b(x)f'(x) + \frac{1}{2}\sigma(x)f''(x) \\ P_t f(x) &= E_x[f(x_t)]. \end{aligned}$$

L'opérateur A est défini pour les fonctions C_2 . Si b ou σ dépend de t , A également, auquel cas on le note A_t . Rappelons que A_t intervient dans la formule d'Itô

$$df(x_t) = A_t f(x_t)dt + \sigma(x_t)f'(x_t)dw_t.$$

Il est bien connu que P_t satisfait $dP_t = P_t A_t dt$ (c'est une conséquence de la formule d'Itô et de la propriété de Markov). En particulier, si b et σ ne dépendent pas de t (la chaîne de Markov est homogène), on montre que $P_t = e^{tA}$ (au sens de $\lim_n (1 - tA/n)^{-n}$).

On peut vérifier que si $b, b', b'', \sigma, \sigma'$ et σ'' sont bornés

$$P_h f(x) = E_x[f(x_h)] = f(x) + hAf(x) + \|f\|O(h^2), \quad \|f\| = \sum_{i=0}^4 \|f^{(i)}\|_\infty.$$

Ceci vient de l'équation $P_h - P_0 = \int_0^h A_s ds + \int_0^h \int_0^s P_u A_u A_s du ds$ (utiliser que $\int_0^h P_s A_s ds = P_h - I$). Par ailleurs si b et σ sont bornées, on obtient à l'aide de la formule de Taylor la propriété :

$$E[f(y_{n+1})] = E[f(y_n)] + hE[Af(y_n)] + \|f\|O(h^2).$$

Finalement

$$E[f(y_{n+1})] = E[P_h f(y_n)] + \|f\|O(h^2).$$

D'où, en appliquant cette formule par récurrence à partir de N :

$$\begin{aligned} E[f(y_N)] &= P_T f(y_0) + O(h^2) \sum_{0 \leq n \leq N} \|P_{nh} f\| \\ &= E[f(x_T)] + O(h) \sup_{0 \leq t \leq T} \|P_t f\|. \end{aligned}$$

Il ne s'agit plus que de borner les dérivées de $P_t f$. Si l'on note $x_t(z)$ la solution de l'EDS partant de z , vue comme fonction de t et z , on a

$$P_t f(z) = E[f(x_t(z))].$$

Il est classique que, si b et σ sont C^1 à dérivée bornée, $x_t(z)$ est une fonction mesurable des deux variables et que sa dérivée en z , que l'on notera $x'_t(z)$ s'obtient par dérivation de l'équation de départ

$$dx'_t = b'(x_t)x'_t dt + \sigma'(x_t)x'_t dw_t, \quad x'_0 = 1.$$

On a par conséquent

$$\begin{aligned} \frac{d}{dz} P_t f(z) &= E[f'(x_t(z))x'_t(z)] \\ \left| \frac{d}{dz} P_t f(z) \right| &= \|f'\|_\infty \sup_{0 < s < T} E[|x'_s(z)|]. \end{aligned}$$

Les dérivées suivantes se traitent de la même manière. On a donc le résultat si b et σ ont leurs quatre premières dérivées bornées.

Pour des résultats plus fins, nous renvoyons à [55], théorème 2.2.

9.5 Résumé

Schéma	ordre fort	ordre faible	Δy
Euler	0,5	1	$bh + \sigma\delta\sqrt{h}$
Milstein	1	1	$bh + \sigma\delta\sqrt{h} + \sigma\sigma'(\delta^2 - 1)h/2$

TAB. 9.2 – Vitesse de convergence des schéma et conditions. On note b et σ pour $b(nh, y_n)$ et $\sigma(nh, y_n)$, et δ v.a. normale centrée réduite.

On peut, sans changer l'ordre faible, remplacer les δ_n gaussiennes standard par d'autres variables centrées réduites. Par exemple un Bernoulli, ou encore la variable δ , telle que $P(\delta = \sqrt{3}) = P(\delta = -\sqrt{3}) = 1/6$ et $P(\delta = 0) = 2/3$ (dont les moments sont ceux de la gaussienne jusqu'à l'ordre 4). Remarquer que dans le cas du Bernoulli, les schémas de Milstein et Euler coïncident, ceci explique pourquoi ils ont même ordre faible.

L'équation (9.2) restera souvent valide même pour des fonctions f non-régulières pourvu que la simulation soit faite avec des variables normales.

Il existe des schémas d'ordre supérieur, R-K, et implicites ; ils ont très vite des expressions assez compliquées, pas toujours généralisables au cas multidimensionnel ; nous renvoyons à [28] et [55].

9.6 Exercices de simulation. Exemples

9.6.1 Martingale exponentielle.

Le processus $x_t = \exp((b - \sigma^2/2)t + \sigma w_t)$ satisfait

$$dx_t = bx_t dt + \sigma x_t dw_t.$$

Si $b = 0$, c'est une martingale qui tend p.s vers 0. Noter qu'ici on a pour Euler et Milstein respectivement :

$$\begin{aligned} y_{n+1} &= y_n(1 + bh + \sigma\sqrt{h}\delta_n) \\ y_{n+1} &= y_n(1 + bh + \sigma\sqrt{h}\delta_n + \sigma^2 h(\delta_n^2 - 1)/2). \end{aligned}$$

Vérifier expérimentalement la vitesse de convergence forte pour $b = 1/2, \sigma = 1, T = 1$: On calculera par Monte-Carlo $E[|x_1 - y_N|^2]$ pour différentes valeurs de h (par exemple $h = 0, 1^p, p = 2 + k/2, k = 0, \dots, 4$), points que l'on portera sur un graphique log-log ; pour le Monte Carlo, on fera environ une centaine de tirages pour chaque valeur de h . On comparera ainsi Euler et Milstein.

9.6.2 Pont brownien.

Il s'agit de $x_t = w_t - tw_1$. On vérifie que $x_t - \int_0^t +\frac{x_s}{1-s} ds$ est un mouvement brownien et donc sa loi coïncide avec celle de

$$dx_t = -\frac{x_t}{1-t} dt + dw_t.$$

Simuler et étudier la convergence comme précédemment.

9.6.3 Oscillateur de Van der Pol.

On part de l'équation déterministe

$$\ddot{x} + \dot{x} - (\alpha - x^2)x = 0.$$

Si α est considéré comme aléatoire, on l'écrit alors (formellement) comme $\alpha + \sigma x \xi_t$ où ξ_t est un bruit blanc ("dérivée" du mouvement brownien $\xi_t dt = dw_t$) et l'équation devient

$$\ddot{x}_t + \dot{x}_t - (\alpha - x_t^2)x_t = \sigma x_t \xi_t$$

En notant $y = \dot{x}$, le processus vectoriel (x, y) satisfait

$$\begin{aligned} dx_t &= y_t dt \\ dy_t &= -y_t dt + (\alpha - x_t^2)x_t dt + \sigma x_t dw_t. \end{aligned}$$

On simulera les situations

$$\alpha = 1, \quad \sigma \in \{0; 0, 2; 0, 5\}, \quad T = 8, \quad x_0 \in [-4; -2], \quad y_0 = 0$$

et l'on regardera, en particulier quand σ grandit, le comportement au voisinage des attracteurs -1 et 1 en traçant x seulement, avec T de l'ordre de 100 (ou 1000. Pour aller vite, ne pas prendre h trop petit).

9.6.4 Dynamique des populations.

L'équation déterministe est $\dot{x} = (\lambda - x)x$ qui exprime que pour une population x petite, le nombre de naissances moins le nombre de morts est proportionnel à x , mais quand x approche de λ , le manque de ressources se fait sentir, impliquant une décroissance quand x dépasse cette valeur. Noter les deux points stationnaires et remarquer que 0 est instable si $\lambda > 0$ et stable sinon. Les aléas sur les ressources disponibles font que λ est aléatoire, et est remplacé par $\lambda + \sigma\xi_t$ où ξ_t est un bruit blanc. On obtient

$$dx_t = (\lambda - x_t)x_t dt + \sigma x_t dw_t.$$

Un modèle vectoriel pour plusieurs espèces est

$$dx_i = (\lambda_i + \sum \mu_{ij}x_j)x_i dt + \sigma_i x_i dw_t.$$

où $\lambda_i - \mu_{ii}x_i$ est la vitesse de croissance en absence d'autre espèce, et μ_{ij} est positif si i est prédateur de j , négatif sinon, et nul si les espèces s'ignorent (matrice antisymétrique en signe). Exemple à trois populations :

$$\begin{aligned} dx_1 &= (\lambda_1 - \mu_{11}x_1 - \mu_{12}x_2)x_1 dt + \sigma_1 x_1 dw_t \\ dx_2 &= (-\lambda_2 + \mu_{21}x_1 - \mu_{22}x_2 - \mu_{23}x_3)x_2 dt + \sigma_2 x_2 dw_t \\ dx_3 &= (-\lambda_3 + \mu_{32}x_2 - \mu_{33}x_3)x_3 dt + \sigma_3 x_3 dw_t. \end{aligned}$$

Interpréter les signes.

Chapitre 10

Bootstrap

10.1 Introduction

Soit $T(y_1, \dots, y_n)$ une statistique basée sur n observations, par exemple un estimateur d'un certain paramètre, ou encore, un estimateur d'un moment des y_i (supposés indépendantes et de même loi). On cherche à avoir de l'information sur la distribution de cette statistique, par exemple

- variance
- intervalles de confiance
- fonction de répartition

Une méthode pourrait consister à faire une estimation de la densité de y pour en déduire ensuite les informations nécessaires sur la distribution de T . Cette façon de faire est toutefois difficile à réaliser et coûteuse.

Les méthodes considérées ici consistent à estimer par simulation cette distribution. Comme la loi de y n'est pas connue, il faut l'approcher.

Un bonne courte introduction à la théorie du bootstrap se trouve dans [58]. La référence [12] contient de nombreux exemples et détails pratiques et [37] est une excellente étude plus approfondie.

10.2 Estimation par répétitions

On est ici nécessairement dans un cadre paramétrique. On se donne donc une famille de lois P_θ , et une suite d'observations $Y = (y_1, \dots, y_n)$, indépendantes, de loi P_{θ^*} chacune ; θ^* est le paramètre inconnu. Soit $\hat{\theta}(Y)$ un estimateur de θ^* .

Si θ^* était connu, on pourrait simuler d'autres suites d'échantillons sous la loi P_{θ^*} , disons $Y^r = (y_1^r, \dots, y_n^r)$, $r = 1, \dots, R$, et observer la distribution empirique des $\hat{\theta}(Y^r)$. On aurait par exemple un estimateur de la variance de $\hat{\theta}(Y)$ avec la formule

$$\hat{\sigma}^2 = \frac{1}{R} \sum_{r=1}^R (\hat{\theta}(Y^r) - \theta^*)^2.$$

Comme θ^* est inconnu, on peut procéder de la façon suivante : simuler d'autres suites $Y^r = (y_1^r, \dots, y_n^r)$ sous la loi $P_{\hat{\theta}(Y)}$ et observer la distribution des $\hat{\theta}(Y^r)$. On a par exemple un estimateur de la variance σ^2 de $\hat{\theta}(Y)$ avec la formule

$$\hat{\sigma}^2 = \frac{1}{R} \sum_{r=1}^R (\hat{\theta}(Y^r) - \hat{\theta}(Y))^2.$$

Noter qu'on a mis ici $\hat{\theta}(Y)$ car c'est la valeur du vrai paramètre dans le cas où la distribution est $P_{\hat{\theta}(Y)}$. Cette méthode est clairement basée sur les hypothèses

1. Exactitude du modèle : les y_i suivent effectivement la loi P_{θ^*} .
2. Régularité et proximité : la statistique recherchée (ici σ^2) est une fonction suffisamment régulière de θ^* , et $\hat{\theta}(Y)$ est suffisamment proche de θ^* .

10.3 Principes de base du bootstrap

Cette méthode a l'avantage de fonctionner même dans un cadre **non-paramétrique**. Elle est simplement basée sur le principe de base de l'estimation : remplacer une fonction de la vraie distribution (inconnue) par la même fonction appliquée à la distribution empirique. Pour fixer les idées, donnons-nous l'exemple suivant où l'on estime le moment d'ordre de 4 des y_i supposés de même loi

$$T(Y) = \frac{1}{n} \sum_{i=1}^n y_i^4, \quad T^* = E[y_1^4] = \lim_{n \rightarrow \infty} T(Y).$$

Pour bien comprendre le bootstrap, il faut interpréter $T(y_1, \dots, y_n)$ comme une fonction de la répartition empirique F_n des données :

$$F_n(x) = \frac{1}{n} \#\{i : y_i \leq x\}.$$

En effet, la donnée de F_n est équivalente à celle de l'échantillon Y . On utilisera alors l'abus de notations : $T(Y) = T(F_n)$; dans l'exemple plus haut on a $T(F_n) = \int y^4 dF_n(y)$. $T(F_n)$ est une estimée de $T^* = T(F)$, limite quand n tend vers l'infini, car F_n tend vers F en norme uniforme (théorème de Glivenko-Cantelli) et T sera supposée continue pour cette métrique (condition généralement satisfaite).

Le bootstrap consiste à produire des "échantillons indépendants" de T en faisant **comme si la distribution empirique des y_i était la vraie distribution** des y_i :

- Générer un échantillon $Y' = (y'_1, \dots, y'_n)$ en effectuant n tirages uniformes avec remise dans $\{y_1, \dots, y_n\}$
- $T(Y')$ est un échantillon de la statistique.

Ayant produit un grand nombre d'échantillons Y^1, \dots, Y^B et de répétitions, $T(Y^1), \dots, T(Y^B)$, on peut par exemple estimer l'écart quadratique moyen entre $T(Y)$ et T^* par

$$\hat{\sigma}^2 = \frac{1}{B} \sum_{b=1}^B (T(Y^b) - T(Y))^2.$$

Pour une région de confiance, soit ε tel que 95% des $T(Y_b)$ satisfassent

$$T(Y) \in [T(Y^b) - \varepsilon, T(Y^b) + \varepsilon]$$

alors on considèrera que l'intervalle

$$[T(Y) - \varepsilon, T(Y) + \varepsilon]$$

est une région de confiance pour T^* de niveau approximatif 5%. Noter que le passage se fait de la façon suivante :

	échantillons		paramètres	
monde de départ	Y	$T(Y)$	T^*	F
monde bootstrappé	Y^b	$T(Y^b)$	$T(Y)$	F_n

et l'on calcule sous F_n (monde bootstrappé) ce que l'on veut connaître sous F .

Avantages et inconvénient du bootstrap. Il sont essentiellement :

1. La simplicité du principe.
2. Des résultats expérimentaux assez bons en divers domaines, par exemple en régression non-paramétrique [22], ou en calibration d'intervalles de confiance [20], même sur des échantillons de taille modeste.
3. Des théorèmes démontrant la validité des approximations jusqu'au deuxième ordre en la taille de l'échantillon. Ceci explique le point précédent.
4. Un inconvénient : Le bootstrap n'est sûr que dans un cadre où l'approximation gaussienne est valide ; de plus les valeurs extrêmes dans le monde réel et le monde bootstrappé ont a priori des distributions très différentes [59]. Ceci signifie que dans l'exemple ci-dessus, il faut que $\sqrt{n}(T(Y) - T^*)$ tende en loi vers une gaussienne et que l'on ne peut prendre ε d'un ordre inférieur à $1/\sqrt{n}$. Voir cependant le § 10.6.4.

Un mot de la théorie. Un résultat théorique général typique est que sous certaines hypothèses de régularité de T (considérée comme fonction de F^1) alors les v.a. $E = \sqrt{n}(T(F_n) - T(F))$ tendent en loi vers $\mathcal{N}(0, \sigma^2)$, $\sigma > 0$ et pour (presque) tout Y , les $E^b = \sqrt{n}(T(F_n^b) - T(F_n))$ ont la même limite en loi. Cependant cette propriété n'est pas très passionnante, car on peut estimer en général σ sans trop de problème.

D'autres résultats, [37] p.19, assurent que la loi E^b est proche de celle E à un *ordre meilleur* que l'approximation gaussienne ; malheureusement, cette proximité n'est garantie en général que pour des versions plus sophistiquées du bootstrap, comme le bootstrap avec "normalisation pivotale" ou le bootstrap régularisé en régression non-paramétrique [22], ou encore le double bootstrap (cf [4] pour le traitement complet d'un exemple simple).

Voir aussi le § 10.6.4 pour le cas non asymptotiquement gaussien.

10.4 Exemples

Calibration d'une région de confiance. Les régions de confiance pour les estimateurs sont de la forme :

$$\mathcal{R}_\alpha = \{\theta : \varphi(\theta, \hat{\theta}) < \lambda\}$$

¹ On demandera l'existence d'une certaine fonction $\varphi(x)$ à variation totale finie, appelée fonction d'influence, telle que pour toute fonction de répartition G ,

$$T(G) = T(F) + E_G[\varphi(y)] + o(\|G - F\|_\infty).$$

On a alors $\sigma^2 = \text{Var}(\varphi(Y))$. En particulier, T doit varier peu si l'on perturbe la distribution de y par une masse de Dirac ($G = (1 - \varepsilon)F + \varepsilon 1_{\geq x_0}$).

où typiquement $\varphi(\theta, \hat{\theta}) = (\theta - \hat{\theta})^T S(\theta - \hat{\theta})$ pour une certaine matrice S et $\lambda = \chi_p^2(\alpha)$. Malheureusement, cette région n'est bien de niveau α que si le nombre d'échantillon est suffisamment élevé. Le bootstrap permet d'évaluer autrement la valeur de λ pour laquelle cette région est de niveau α . En effet, le niveau de confiance de la région estimé par bootstrap sera simplement donné par la proportion de $\hat{\theta}^b$ tels que $\varphi(\hat{\theta}, \hat{\theta}^b) < \lambda$. On peut donc régler λ de sorte à avoir un niveau donné.

Distribution de la médiane (ou autre...). On veut savoir la loi de $m_Y - m^*$ où m_Y est la médiane empirique des y_i et m^* la médiane de leur distribution. Pour cela, on observera simplement la loi de $m_{Y^b} - m_Y$ quand Y^b est un échantillon bootstrappé (et que Y est fixe, puisque c'est l'observation). Dans le deuxième tracé de la figure 10.1 on a superposé la fonction de répartition de $m_Y - m^*$ (calculée théoriquement) et celle $m_{Y^b} - m_Y$ (calculée empiriquement).

Régression non-paramétrique adaptative. C'est une situation où il faut choisir un estimateur parmi une infinité indexée par un paramètre. On observe :

$$y_i = f(x_i) + \xi_i$$

où les (x_i, ξ_i) sont i.i.d. et f est inconnue. On a par exemple l'estimateur à noyau

$$\hat{f}_h(x) = \frac{\sum_i p_i y_i}{\sum_i p_i}, \quad p_i = K\left(\frac{x - x_i}{h}\right).$$

On cherche h de sorte à minimiser l'erreur de prédiction $E[(y - \hat{f}_h(x))^2]$ où l'espérance porte sur la paire (x, y) de distribution F ainsi que sur les (x_i, y_i) implicites dans l'estimateur. Le choix de h n'est pas toujours très facile [23]. L'estimateur par bootstrap de cette erreur est

$$c(h) = \frac{1}{nB} \sum_{i=1}^n \sum_{b=1}^B (y_i - \hat{f}_h^b(x_i))^2$$

où \hat{f}_h^b est l'estimateur de f basé sur un échantillon bootstrappé de (x_i, y_i) :

$$\hat{f}_h^b(x) = \frac{\sum_i p_i^b y_i^b}{\sum_i p_i^b}, \quad p_i^b = K\left(\frac{x - x_i^b}{h}\right).$$

On choisira donc la valeur de h qui minimise $c(h)$. Cette méthode est la plus rustique ; on préfère généralement utiliser un bootstrap régularisé (exercice 10.7.3).

Estimation de densité. On aura cette fois $\hat{p}_h(x) = \frac{1}{nh} \sum_i K\left(\frac{x - x_i}{h}\right)$ et l'on minimisera

$$c(h) = \frac{-1}{nB} \sum_{i=1}^n \sum_{b=1}^B \log \hat{p}_h^b(x_i) \quad \text{où} \quad \hat{p}_h^b(x) = \frac{1}{nh} \sum_i K\left(\frac{x - x_i^b}{h}\right).$$

Le log est simple et naturel pour illustrer le propos mais n'est pas très bon, et il est meilleur en pratique d'utiliser les estimées habituelles du risque quadratique ; on pourra consulter [26] pour un exposé concis et récent sur les estimateurs généraux de h .

10.5 Échec du bootstrap dans un cas non-régulier

Considérons l'estimation des quantiles : $T(F) = \max\{x : F(x) < \alpha\}$. On a ici $\sqrt{n}(T(F_n) - T(F)) \sim \mathcal{N}(0, \sigma^2)$, avec $\sigma^2 = \alpha(1 - \alpha)$.

Dans le cas où $0 < \alpha < 1$, tout va bien (deuxième figure). Sinon, l'expérience n'est pas régulière et la bonne normalisation pour $T(F_n) - T(F)$ est n et non pas \sqrt{n} ; dans ce cas on n'obtient pas la distribution asymptotique de $n(T(F_n) - T(F))$ en calculant celle de $n(T(F_n^b) - T(F_n))$.

Faisons le lien avec la note page 89. On montre sans grande difficulté que si F est dérivable et de dérivée strictement positive en $T(F)$, alors il y a dérivabilité et la fonction d'influence vaut

$$\varphi(x) = F'(T(F))^{-1}(\alpha - 1(x < T(F))).$$

En particulier on voit que si $\alpha = 1$ (estimation du maximum du support de la distribution et $T(Y) = T(F_n) = \sup_i y_i$), les ennuis commencent, surtout si F admet une dérivée nulle en $F^{-1}(1)$, auquel cas φ est infini ; on peut également vérifier que T n'est pas une fonction continue au voisinage de F , en effet

$$\text{si } T(F) = \theta \quad \text{alors } \forall \varepsilon > 0, T((1 - \varepsilon)F + \varepsilon\delta_{\theta+1}) = \theta + 1.$$

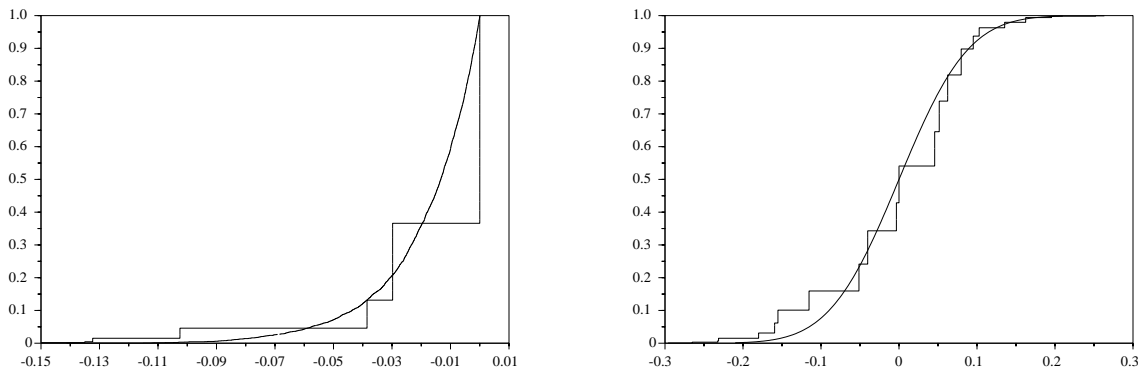


FIG. 10.1 – L'échantillon $Y = (y_1, \dots, y_n)$ consiste en 50 tirages de la loi uniforme sur $[0, 1]$. $T(Y) = \sup(y_i)$. La première figure représente la fonction de répartition de la distribution de $T(Y) - T(F_Y) = T(Y) - 1$ obtenue théoriquement et son estimation par bootstrap. On retrouve la valeur $e^{-1} \simeq 0,37$ qui est la probabilité asymptotique qu'un sous-échantillon de taille n de tiré uniformément dans $\{y_1, \dots, y_n\}$ ne contienne pas le plus grand élément de cet échantillon.

Sur la seconde figure l'expérience est la même sauf que cette fois-ci T est la médiane.

Un autre exemple d'échec du bootstrap est celui des U -statistiques : $T = \sum_{ij} H(y_i, y_j)$, où H est une certaine fonction, cf [37] p. 37.

10.6 Variantes et détails pratiques

10.6.1 Bootstrap régularisé.

Ce n'est rien d'autre que l'estimation de la distribution des y_i pour générer de nouveaux échantillons ; c'est donc une simple méthode de répétitions couplée avec une estimation préalable. On peut estimer cette loi dans un cadre paramétrique ou non-paramétrique .

Pour un exemple en estimation de densité, voir l'exercice 10.7.3.

10.6.2 Bootstrap semi-paramétrique.

Dans un modèle paramétrique de la forme

$$y_i = f(\theta^*, x_i) + u_i$$

où les x_i sont déterministes, plutôt que de faire le bootstrap direct des données (ce qui peut paraître assez injustifié puisque les régresseurs sont généralement considéré comme déterministes), on préférera estimer θ^* , puis les u_i puis générer les données bootstrappées :

$$y_i^b = f(\hat{\theta}, x_i) + u_i^b$$

où les u_i^b sont tirés uniformément dans l'ensemble des $\hat{u}_i = y_i - f(\hat{\theta}, x_i)$.

Il pourra être plus fin de générer les u_i par leave-one-out : $\hat{u}_i = y_i - f(\hat{\theta}^{(i)}, x_i)$, où $\hat{\theta}^{(i)}$ est l'estimée obtenue en excluant la paire (x_i, y_i) des données.

10.6.3 Normalisation pivotale.

Cette méthode améliore souvent considérablement les performances du bootstrap. Elle est tout-à-fait recommandée dans les cas réguliers, [37] chap.5.

Soit $\hat{\sigma}(Y)$ un estimateur (non-nécessairement précis) de la variance de $T(Y)$. Alors la loi de $(T(Y) - T(F_y))/\hat{\sigma}(Y)$ peut être approchée par la loi empirique des $(T(Y^b) - T(Y))/\hat{\sigma}(Y^b)$, $b = 1, \dots, B$. Ceci conduit à l'approximation en loi

$$T(Y) - T(F_y) \sim \frac{\hat{\sigma}(Y)}{\hat{\sigma}(Y^b)}(T(Y^b) - T(Y))$$

dans le membre de gauche c'est Y qui varie, et dans celui de droite, c'est bien entendu Y^b et Y est fixé. Cette méthode conduit à des **intervalles de confiance** pour $T(F_y)$, qui sont de la forme $[T(Y) - \delta_1 \hat{\sigma}(Y), T(Y) + \delta_2 \hat{\sigma}(Y)]$.

10.6.4 Bootstrap par sous-échantillonnage

Dans cette méthode, Y^b est un sous-échantillon de Y de longueur $b_n < n$, obtenu par tirage sans remise. On montre que si b_n n'est pas trop grand, on pourra estimer la distribution $T(Y) - T^*$ dans un cadre très général, comprenant même le contre-exemple du paragraphe précédent. En effet, si

$$\begin{aligned} b_n &\longrightarrow \infty \\ \frac{b_n}{n} &\longrightarrow 0 \\ \tau(n)(T(Y) - T^*) &\longrightarrow \mathcal{D} \quad \text{en distribution} \end{aligned}$$

où \mathcal{D} est une certaine loi et $\tau(n)$ est la vitesse de convergence (dans les cas réguliers \mathcal{D} est gaussienne et $\tau(n) = \sqrt{n}$), alors, pour presque tout ω ,

$$\tau(b_n)(T(Y^b) - T(Y)) \longrightarrow \mathcal{D} \quad \text{en distribution.}$$

Ceci signifie que pour tout point x de continuité de \mathcal{D}

$$\Phi_n(x/\tau(b_n)) = \frac{1}{B} \sum_{b=1}^B 1_{(T(Y^b) - T(Y)) \leq x/\tau(b_n)} \longrightarrow P(\mathcal{D} \leq x) \quad (10.1)$$

où la somme est étendue aux $B = C_n^{b_n}$ sous-échantillons de Y . Pour un énoncé plus précis, voir [48]².

Si cette méthode est plus robuste que le simple bootstrap, on peut vérifier qu'en revanche la convergence vers \mathcal{D} est moins rapide dans les cas réguliers [2].

La valeur $b_n = n^{2/3}$ est souvent recommandée. Cette méthode est donc très bonne pour obtenir au moins la forme de la distribution ; en effet, le facteur d'échelle $\tau(n)$ est a priori inconnu. La référence [3] propose, dans le cas où $\tau(n) = n^\alpha$, une méthodologie pour estimer α qui donne, après report dans (10.1), une estimée consistante de la fonction de répartition de \mathcal{D} . L'idée est simplement de calculer $\Phi_n(x)$ pour différents choix de b_n et de voir pour quel α les courbes $\Phi_n(b_n^{-\alpha}x)$ se superposent.

10.7 Exercices et compléments

Exercice 10.7.1 (Estimation de biais) Proposer une méthode de bootstrap pour estimer le biais $E[T(Y) - T(F_y)]$ de l'estimateur T .

Exercice 10.7.2 (Rapport de vraisemblance empirique) Soit G une fonction de répartition associée à une mesure portant les y_i , ce que l'on notera $G \ll F_n$. Si les poids correspondants sont p_i , on définira le rapport de vraisemblance

$$R(G) = 2n \sum p_i^0 \log(p_i^0/p_i)$$

où les p_i^0 sont les poids donnés par F_n (généralement $1/n$). Il a été montré [46] que l'on a asymptotiquement sous certaines hypothèses de régularité

$$\inf \{R(G); T(G) = T(F_y), G \ll F_n\} \sim \chi^2$$

(si $T(G) \in \mathbb{R}^p$, on trouve un χ_p^2). Ceci conduit aux intervalles de confiance asymptotiques

$$T(F_y) \in \{T(G); R(G) \leq c, G \ll F_n\}$$

avec le niveau $1 - F_{\chi^2}(c)$.

1. Proposer une méthode de bootstrap pour réestimer le niveau.
2. On suppose que l'on dispose d'une méthode raisonnable pour estimer F_y . Proposer une méthode de simulation pour réestimer le niveau.

Exercice 10.7.3 (Bootstrap régularisé) On s'intéresse à l'estimation de densité. Supposons que l'on connaisse une valeur h_0 qui est inférieure à la valeur optimale h^* . Utiliser \hat{p}_{h_0} pour construire une méthode de bootstrap régularisé. Donner une justification heuristique à ce choix (on considérera la même méthode en remplaçant h_0 par 0 ou h^*).

Proposer une méthode analogue en régression non-paramétrique $y_i = f(x_i) + \xi_i$ (on supposera les x_i *déterministes*) : il s'agit de partir d'une estimée pilote \hat{f}_0 , d'en déduire des $\hat{\xi}_i$, puis fabriquer des échantillons bootstrappés en tirant des $\hat{\xi}_i^b$.

En pratique, on pourra prendre h_0 inférieur à la valeur de h obtenue par validation croisée, ou par un bootstrap non-régularisé [22]. Noter que ce h_0 est effectivement généralement plus petit que h^* .

Voir aussi [19] pour une comparaison avec la validation croisée.

²En particulier, dans l'énoncé ci-dessus, la convergence n'a en réalité pas lieu presque sûrement mais en probabilité, à moins d'affiner l'hypothèse de décroissance de b_n/n .

Bibliographie

- [1] A.BENVENISTE, M.MÉTIVIER & P.PRIOURET, *Adaptive Algorithms and Stochastic Approximations*, Springer-Verlag, 1990.
- [2] P. BERTAIL, “Second order properties of an extrapolated bootstrap without replacement under weak assumptionse”, *Bernoulli* 3(2), 1997, 149-179.
- [3] P. BERTAIL, D.N.POLITIS & J.P.ROMANO, “On subsampling estimators with unknown rate of convergence”, *J. Amer. Statist. Assoc.* 94 (1999), No 446, 569-579.
- [4] J.G. BOOTH & P. HALL, “Monte Carlo approximation and the iterated bootstrap”, *Biometrika*, vol 81, No 2, 331-340, 1994.
- [5] I. BOROSH & H. NIEDERREITER, “Optimal multipliers for pseudo-random number generation by the linear congruential method”, *BIT*, 23 :65-74, 1983.
- [6] M. BRAMSON & C. NEUHAUSER, Coexistence for a catalytic surface reaction model. *Ann. Appl. Prob.*, Vol 7, No 3,565-614, 1997.
- [7] L. BREIMAN, *Probability*, Addison-Wesley,1968.
- [8] J.N. CORCORAN & R.L. TWEEDIE, “Perfect Sampling From Independent Metropolis-Hastings Chains”. Disponible sur le site web de R.Tweedie.
- [9] R. CRANLEY, T.N.L. PATTERSON, “Randomization of number theoretic methods for multiple integration”, *SIAM J. Num. Anal.*, 13(6), 904-914, 1976.
- [10] N. CRESSIE, *Statistics for spatial data*, Wiley, 1991.
- [11] L. DEVROYE, *Non-Uniform Random Variate Generation*, Springer, 1986. Voir aussi d'autres références sur cgm.cs.mcgill.ca/~luc/rng.html.
- [12] B. EFFRON & R.J. TIBSHIRAMI, *An intrduction to the bootstrap*, Chapman and Hall, 1993.
- [13] K. ENTACHER, *A collection of selected pseudorandom number generators with linear structures*, <http://random.mat.sbg.ac.at/charly/server>.
- [14] G.S. FISHMAN, *Monte Carlo*, Springer 1997.
- [15] X. GUYON, *Méthodes numériques par chaînes de Markov*, prépublication No 113 du SAMOS, Université Paris I, 1999.
- [16] X. GUYON, *Random Fields on a network. Modelling, Statistics, and Applications*, Springer, 1996.
- [17] B. HAJEK, “Cooling schedules for optimal annealing”, *Math. Oper. Res.*, 13 (1988), no. 2, 311–329.
- [18] P. HALL & C.C. HEYDE, *Martingale Limit Theory and Its Applications*, Academic Press, 1980.
- [19] P. HALL, J.S. MARRON & B.U. PARK, “Smoothed cross-validation”, *Probab. Th. Relat. Fields*, 92,1-20, 1992.

- [20] P. HALL, L. PENG, "On prediction intervals based on predictive likelihood or bootstrap methods", *Biometrika*, 86, 4, 871-880, 1999.
- [21] J.M. HAMMERSLEY & D.C. HANDSCOMB, *Monte Carlo Methods*, Chapman & Hall, 1964.
- [22] W. HÄRDLE & A.W. BOWMAN, "Bootstrapping in nonparametric regression : local adaptive smoothing and confidence bands", *J. Amer. Statist. Assoc.* 83 (1988), no. 401, 102–110.
- [23] W. HÄRDLE, P. HALL & J.S. MARRON , "How Far Are Automatically Chosen Regression Smoothing Parameters From Their Optimum", *J. Amer. Statist. Assoc.* 83 (1988), no. 401, 86–101.
- [24] J. INCERA, G. RUBINO, N STIER, "On The Application of Accelerating Simulation Methods in Network Analysis", 2000 International Conference on Parallel and Distributed Processing Techniques and Applications, Las Vegas, June 26-29, 2000. <http://web.mit.edu/nstier/www/>.
- [25] F. JAMES, "A review of pseudorandom number generators", *Comput. Phys. Comm.* 60 (1990), no. 3, 329–344.
- [26] M.C. JONES, J.S. MARRON & S.J. SHEATER "A Brief Survey of Bandwith Selection for Density Estimation", *J. Amer. Statist. Assoc.* 91 (1996), no. 433, 401–407.
- [27] R.M. KARP & M. LUBY, "Monte Carlo algorithms for enumeration and reliability problems," *24th Annual symposium on foundations of computer science*, New-York 1983.
- [28] E. KLOEDEN & E. PLATEN, *Numerical Solution of Stochastic Differential Equations*, Springer, 1992.
- [29] D.E. KNUTH, *The Art of computer programming. Vol. 2*, Addison-Wesley , 1998.
- [30] L'ECUYER, "Efficient and Portable Combined Random Number Generators", *Communications of the ACM*, 31 (1988), 742–749 & 774. Ces générateurs sont disponibles à l'adresse hpux.u-aizu.ac.jp sous le nom de "ranlib".
- [31] P. L'ECUYER, "Testing Random Number Generators", *Proceedings of the 1992 Winter Simulation Conference*, pp 305–313. www.iro.umontreal.ca/~lecuyer/papers.html.
- [32] P. L'ECUYER & P. HELLEKALEK, "Random Number Generators : Selection Criteria and Testing", in *Random and Quasi-Random Point Sets*, Lectures Notes In Statistics, no. 138, Springer, 1998, 223–266.
- [33] P. L'ECUYER, "Good Parameter Sets for Combined Multiple Recursive Random Number Generators", *Operations Research*, 47, 1 (1999), 159–164.
- [34] L. LJUNG & S. GUNNARSON, "Adaptation and tracking in system identification—a survey", *Automatica-J IFAC*, 26 (1990), 7–21.
- [35] M. LÜSCHER, "A portable high-quality random number generator for lattice field theory simulations", *Comput. Phys. Comm.* 79 (1994), no. 1, 100–110.
- [36] P. McCULLAGH, J.A. NEDLER, *Generalized Linear Models*, Chapman & Hall, 1983.
- [37] E. MAMMEN, *When Does Bootstrap Work ?*, Springer, 1992.
- [38] M. MATSUMOTO & T. NISHIMURA, "Mersenne Twister : A 623-dimensionally equidistributed uniform pseudorandom number generator", *ACM Transactions on Modeling and Computer Simulation*, 8(1), 1998.
- [39] C.E. McCULLOCH, "Maximum likelihood algorithm for generalized linear mixed models", *J. Amer. Stat. Ass.*, 95, 223-252, 1996.
- [40] S.P. MEYN & R.L. TWEEDIE, *Markov chains and stochastic stability*, Springer-Verlag, 1993.

- [41] J. MØLLER, *A review on perfect simulation in stochastic geometry*, 2000, Preprint disponible à dimacs.rutgers.edu/~dbwilson/exact.html.
- [42] B.M.NEVEL'SON & R.Z.KHAS'MINSKII, *Stochastic Approximation and Recursive Estimation*, American Mathematical Society Translation of Math. Monographs, vol 47, 1976.
- [43] H. NIEDERREITER, *Random Number Generation and Quasi-Monte Carlo Methods*, CBMS-NSF Reg. Conf. Series in Appl.Math., Vol. 63, SIAM, 1992.
- [44] E. NUMMELIN, *General irreducible Markov chains and non-negative operators*, Cambridge University Press, 1984.
- [45] B. OKSENDAL, *Stochastic differential equations : an introduction with applications*, Springer, 1992.
- [46] A.B. OWEN, "Empirical Likelihood ratio confidence intervals", *Biometrika*, vol.75, No.2, 237-249, 1988.
- [47] D. PETRITIS, *Simulations numériques Monte-Carlo*, polycopié de cours de l'université de Rennes I.
- [48] D.N. POLITIS & J.P. ROMANO, "Large Sample Confidence Regions based on Subsamples under Minimal Assumptions" *Ann. Stat.*, 1994, Vol 22, No 4, 2013-2050.
- [49] B.T.POLYAK, "New stochastic Approximation Type Procedures", *Autom. & Remote Contr.*, 51(7).
- [50] J PROPP, D WILSON, "Coupling from the Past : a User's Guide", *Microsurveys in discrete probability*, 181-192, DIMACS Ser. Discrete Math. Theoret. Comput. Sci., 41, Amer. Math. Soc., 1998.
- [51] S.R. ROSS, *Simulation*, Academic Press, 1997.
- [52] G. N. SARIDIS, "Stochastic Approximation Methods for Identification and Control – A Survey", *IEEE-AC*, vol 19, No 6, décembre 1974.
- [53] J. SHAO & D. TU, *The Jackknife and Bootstrap*, Springer 1996.
- [54] G.R. SHORACK & J.A. WELLNER, *Empirical processes with applications to statistics*, Wiley 1986.
- [55] D. TALAY, "Discrétisation d'une équation différentielle stochastique", *Math. Mod. & Num. Anal.*, Vol 20 No 1, 1986, p.141-179.
- [56] D. TILMAN, "Competition and bio-diversity in spatially structured habits", *Ecology*, 75, 2-16, 1994.
- [57] B. TUFFIN, *Simulation accélérée par les méthodes de Monte Carlo*, thèse de l'université de Rennes 1, 1997.
- [58] A.W. VAN DER VAART, *Asymptotic Statistics*, Cambridge University Press, 1998.
- [59] A.T.A. WOOD, "Bootstrap relative errors and subexponential distributions", *Bernoulli*, 6(5), 2000, 809-834.
- [60] Zhang, Y. (1993) A shape theorem for epidemics and forest fires with finite range interactions. *Ann. Prob.* **21** , 1755-1781

Index

- algorithme stochastique, 71
- annulation de bruit, 74
- antithétiques (variables), 32
- approximation stochastique, 71

- bayésien hiérarchique, 47
- Binomial conditionnel (modèle), 56
- birthday spacings, 19

- chaîne de Markov contrôlée, 74
- champ de Gibbs, 53
- champ de Markov, 53
- champ gaussien, 55
- clique, 53
- couplage sur le passé, 66

- discrepance, 35
- données manquantes, 44, 75
- décompte, 22

- échantillonnage corrélé, 32
- échantillonnage préférentiel, 29
- échantillonnage stratifié, 33
- échantillonnage parfait, 66
- EM (algorithme), 44
- estimation, 43
- événements rares, 30

- forme produit, 43

- Gibbs (échantillonneur), 43, 55
- graphes, 22, 42

- hiérarchique (bayésien), 47

- image, 54
- importance sampling, 29
- importance splitting, 59
- inverse gaussienne, 15
- inversion, 10
- inversion approchée, 14

- Kiefer-Wolfowitz (algorithme), 75
- Kolmogorov (test), 17

- lacunes, 20

- Metropolis (algorithme), 41, 44, 55, 66, 69, 77
- mélange avec poids négatifs, 14

- points proches, 19, 20
- Poisson, 14
- poursuite, 76
- processus spatiaux, 55

- recuit simulé, 77
- rejet, 11
- renouvellement, 62
- Robbins-Monro, 72
- régénération, 62

- saut (processus), 50, 57
- SEM (algorithme), 45

- test des corrélations, 19
- test des lacunes, 20
- test des points proches, 19, 20
- test des séries, 19

- variables antithétiques, 32
- variables de contrôle, 31
- voisinage, 53