

MathExp 2018: Chaînes de Markov

Simulation des chaînes de Markov

Ana Busic

Inria Paris - DI ENS

<http://www.di.ens.fr/~busic/>

Saint Flour, Mai 2018

Plan

Garder la suite aléatoire en mémoire ?

SED non-monotones ?

Echantillonneur de Gibbs

Remarques

Nous avons vu :

- ▶ Représentation des chaînes de Markov par des SED.
- ▶ Simulation MCMC.
- ▶ Simulation parfaite par couplage depuis le passé.
- ▶ Cas monotone (et anti-monotone).

Remarques

Nous avons vu :

- ▶ Représentation des chaînes de Markov par des SED.
- ▶ Simulation MCMC.
- ▶ Simulation parfaite par couplage depuis le passé.
- ▶ Cas monotone (et anti-monotone).

Quelques remarques/problèmes :

- ▶ Temps de couplage ?
- ▶ Garder la suite aléatoire en mémoire ?
- ▶ SED non-monotones ?

Read-once randomness (Wilson 1999)

Algorithme Propp & Wilson : on recommence les trajectoires en partant des temps $-N_1, -N_2, \dots$ (avec $N_1 < N_2 < \dots$) jusqu'à j tel que $|S_{N_j}| = 1$, où $S_{N_j} \triangleq \mathcal{X} \cdot a_{N_j \rightarrow 1}$.

Problème : garder la suite aléatoire en mémoire peut être très cher (dépend du temps de couplage!)

Read-once randomness (Wilson 1999)

Algorithme Propp & Wilson : on recommence les trajectoires en partant des temps $-N_1, -N_2, \dots$ (avec $N_1 < N_2 < \dots$) jusqu'à j tel que $|S_{N_j}| = 1$, où $S_{N_j} \triangleq \mathcal{X} \cdot a_{N_j \rightarrow 1}$.

Problème : garder la suite aléatoire en mémoire peut être très cher (dépend du temps de couplage!)

En pratique : garder les "seed" utilisées pour générer les sous-séquences - peut mener à toute sorte des comportements inattendus (à éviter!)

Wilson's modification

[Wilson 1999] Idée : une sorte de couplage en avant mais où on n'arrete pas la simulation au moment même du couplage - on continue encore pendant **un certain temps aléatoire**.

Wilson's modification

[Wilson 1999] Idée : une sorte de couplage en avant mais où on n'arrete pas la simulation au moment même du couplage - on continue encore pendant **un certain temps aléatoire**.

Comment définir ce temps ?

Wilson's modification

[Wilson 1999] Idée : une sorte de couplage en avant mais où on n'arrête pas la simulation au moment même du couplage - on continue encore pendant un certain temps aléatoire.

Comment définir ce temps ?

On commence par faire quelques observations sur le couplage depuis le passé...

Observation I

Pour le cas monotone, on a utilisé les temps
 $(N_1, N_2, N_3, N_4 \dots) = (1, 2, 4, 8, \dots)$ mais toute suite strictement croissante
marche aussi.

Observation I

Pour le cas monotone, on a utilisé les temps
 $(N_1, N_2, N_3, N_4 \dots) = (1, 2, 4, 8, \dots)$ mais toute suite strictement croissante marche aussi.

*On peut même prendre une suite **aléatoire strictement croissante** !*

Observation I

Pour le cas monotone, on a utilisé les temps $(N_1, N_2, N_3, N_4, \dots) = (1, 2, 4, 8, \dots)$ mais toute suite strictement croissante marche aussi.

*On peut même prendre une suite **aléatoire strictement croissante** !*

Preuve : soit $N_1 < N_2 < \dots$ une suite aléatoire d'entiers, indépendante de la suite aléatoire des événements tirés. Alors sachant (N_1, N_2, \dots) l'algorithme renvoie un échantillon non-biaisé de π . Cela est vrai pour toute réalisation de (N_1, N_2, \dots) , ainsi l'algorithme (avec les temps (N_1, N_2, \dots) aléatoires) donne un échantillon non-biaisé de π .

Observation II

Continuer l'algorithme depuis encore plus loin dans le passé n'est pas gênant !

(à part pour le temps supplémentaire d'exécution de l'algorithme)

Observation II

Continuer l'algorithme depuis encore plus loin dans le passé n'est pas gênant !

(à part pour le temps supplémentaire d'exécution de l'algorithme)

Preuve : si $|S_n| = 1$ pour un $n \in \mathbb{N}$, alors pour tout $m > n$, on a $S_m = S_n$.

Choix de la suite aléatoire

Comment choisir la suite aléatoire strictement croissante $N_1 < N_2 < \dots$?

Soit (T_1, T_2, T_3, \dots) une suite aléatoire d'entiers i.i.d. avec la distribution égale à celle du couplage en avant.

Choix de la suite aléatoire

Comment choisir la suite aléatoire strictement croissante $N_1 < N_2 < \dots$?

Soit (T_1, T_2, T_3, \dots) une suite aléatoire d'entiers i.i.d. avec la distribution égale à celle du couplage en avant.

Question

Comment générer (T_1, T_2, T_3, \dots) ?

Choix de la suite aléatoire

Comment choisir la suite aléatoire strictement croissante $N_1 < N_2 < \dots$?

Soit (T_1, T_2, T_3, \dots) une suite aléatoire d'entiers i.i.d. avec la distribution égale à celle du couplage en avant.

Question

Comment générer (T_1, T_2, T_3, \dots) ?

On fixe :

$$N_1 = T_1$$

$$N_2 = T_1 + T_2$$

$$N_3 = T_1 + T_2 + T_3$$

$$\vdots$$

$$N_n = N_{n-1} + T_n$$

Observation III

La probabilité que l'algorithme Propp & Wilson en partant du temps initial $-N_1 = -T_1$ se termine (couplage avant le temps 0) est au moins $1/2$.

Observation III

La probabilité que l'algorithme Propp & Wilson en partant du temps initial $-N_1 = -T_1$ se termine (couplage avant le temps 0) est au moins $1/2$.

Preuve : Soit M_1 le temps nécessaire pour coupler en partant du temps $-N_1$ (et dépassant le temps 0 si besoin !)

Alors M_1 et N_1 ont la même distribution et ils sont indépendants, donc :

$$\mathbb{P}(M_1 \leq N_1) = \mathbb{P}(M_1 \geq N_1).$$

Observation III

La probabilité que l'algorithme Propp & Wilson en partant du temps initial $-N_1 = -T_1$ se termine (couplage avant le temps 0) est au moins $1/2$.

Preuve : Soit M_1 le temps nécessaire pour coupler en partant du temps $-N_1$ (et dépassant le temps 0 si besoin !)

Alors M_1 et N_1 ont la même distribution et ils sont indépendants, donc :

$$\mathbb{P}(M_1 \leq N_1) = \mathbb{P}(M_1 \geq N_1).$$

Aussi :

$$\mathbb{P}(M_1 \leq N_1) + \mathbb{P}(M_1 \geq N_1) = 1 + \mathbb{P}(M_1 = N_1) \geq 1.$$

Donc $\mathbb{P}(M_1 \leq N_1) \geq 1/2$.

Observation III - suite

On peut continuer le raisonnement...

Si l'algorithme ne termine pas en partant de $-N_1$, alors avec la probabilité au moins $1/2$ on observe le couplage avant le temps $-N_1$ en partant du temps $-N_2 = -(N_1 + T_2)$.

Plus généralement, avec la probabilité au moins $1/2$ le couplage avant le temps $-N_{j-1}$ en partant du temps $-N_j = -(N_{j-1} + T_j)$.

Observation III - suite

On peut continuer le raisonnement...

Si l'algorithme ne termine pas en partant de $-N_1$, alors avec la probabilité au moins $1/2$ on observe le couplage avant le temps $-N_1$ en partant du temps $-N_2 = -(N_1 + T_2)$.

Plus généralement, avec la probabilité au moins $1/2$ le couplage avant le temps $-N_{j-1}$ en partant du temps $-N_j = -(N_{j-1} + T_j)$.

On va dire qu'on a un succès à l'étape j si, en partant du temps $-N_j$, on observe le couplage avant le temps $-N_{j-1}$.

Observation III - suite

On peut continuer le raisonnement...

Si l'algorithme ne termine pas en partant de $-N_1$, alors avec la probabilité au moins $1/2$ on observe le couplage avant le temps $-N_1$ en partant du temps $-N_2 = -(N_1 + T_2)$.

Plus généralement, avec la probabilité au moins $1/2$ le couplage avant le temps $-N_{j-1}$ en partant du temps $-N_j = -(N_{j-1} + T_j)$.

On va dire qu'on a un succès à l'étape j si, en partant du temps $-N_j$, on observe le couplage avant le temps $-N_{j-1}$.

Et pour avoir la probabilité de succès égale à $1/2$, au cas où on observe le couplage exactement au temps $-N_{j-1}$, on tire une pièce non-biaisée et avec proba $1/2$ on déclare le succès, et sinon l'échec.

Modification de Wilson

L'algorithme (juste, mais un peu artificiel !) qui consiste à recommencer depuis les temps $-N_j$ jusqu'au succès donne bien un échantillon non-biaisé de π .

Modification de Wilson

L'algorithme (juste, mais un peu artificiel !) qui consiste à recommencer depuis les temps $-N_j$ jusqu'au succès donne bien un échantillon non-biaisé de π .

Notons le nombre (aléatoire) d'échec avant le succès par Y . Alors Y est une variable aléatoire géométrique de paramètre $p = 1/2$:

$$\mathbb{P}(Y = n) = p(1 - p)^n.$$

L'itération finale (et succès) commence en $-N_{Y+1}$.

Modification de Wilson

L'algorithme (juste, mais un peu artificiel !) qui consiste à recommencer depuis les temps $-N_j$ jusqu'au succès donne bien un échantillon non-biaisé de π .

Notons le nombre (aléatoire) d'échec avant le succès par Y . Alors Y est une variable aléatoire géométrique de paramètre $p = 1/2$:

$$\mathbb{P}(Y = n) = p(1 - p)^n.$$

L'itération finale (et succès) commence en $-N_{Y+1}$.

Idée de modification de Wilson : trouver un moyen de simuler d'abord les trajectoires depuis $-N_{Y+1}$ jusqu'à $-N_Y$, puis de $-N_Y$ jusqu'à $-N_{Y-1}$... et cela jusqu'au temps 0.

Twin run

Executer deux copies indépendantes de couplage en avant : on s'arrête quand les deux couplent !

On déclare la copie qui a couplé d'abord le **gagnant** et l'autre le **perdant** (avec une pièce non-biaisée pour départager si égalité).

Twin run

Executer deux copies indépendantes de couplage en avant : on s'arrête quand les deux couplent !

On déclare la copie qui a couplé d'abord le **gagnant** et l'autre le **perdant** (avec une pièce non-biaisée pour départager si égalité).

Observation : l'évolution de nos trajectoires depuis le passé en partant de $-N_{Y+1}$ jusqu'à $-N_Y$ a justement la même distribution que celle du gagnant.

Algorithme

- ▶ On simule un twin run jusqu'au couplage du perdant. On garde la valeur du gagnant au moment du couplage du perdant.
- ▶ Choisir Y selon la loi géométrique avec $p = 1/2$.
- ▶ Si $Y = 0$ on a fini.
- ▶ Sinon on doit continuer la simulation depuis $-N_Y$ jusqu'à 0. Pour simuler les trajectoires de $-N_Y$ jusqu'à $-N_{Y-1}$, on fait un autre twin run et on prend l'évolution du perdant, ou le perdant évolue de 0 jusqu'au couplage du gagnant !

Puis on continue avec un autre twin run indépendant pour $-N_{Y-1}$ jusqu'à $-N_{Y-2}$...

On renvoie la valeur finale du perdant du dernier twin run (au moment du couplage du gagnant du même twin run).

Remarques

Quelques remarques/problèmes :

- ▶ Temps de couplage ?
- ▶ Garder la suite aléatoire en mémoire ?
- ▶ SED non-monotones ?

Le cas non-monotone

- ▶ [Kendall 1998] et [Hägglström and Nelander 1998] une idée similaire pour le cas anti-monotone :

$$x \preceq y \Rightarrow x \cdot a \succeq y \cdot a, \forall a \in A.$$

- ▶ [Kendall and Møller 2000] ont introduit une première idée des processus bornants qui enveloppent toutes les trajectoires.
- ▶ [Huber 2004] une idée similaire des processus pour détecter le couplage, mais sans introduire une notion d'ordre sur les états.

Envelope Perfect Sampling Algorithm

[B., Gaujal, Vincent 2008]

- ▶ Hypothèse : (\mathcal{X}, \preceq) est un treillis.
- ▶ Pour $m, M \in \mathcal{X}$, $[m, M] \triangleq \{x \in \mathcal{X} : m \preceq x \preceq M\}$ un intervalle entre m et M .
- ▶ $\mathcal{I} = \{[m, M] : m, M \in \mathcal{X}, m \preceq M\}$ l'ensemble des intervalles non-vides.
- ▶ Nouvelle fonction de transition $\square : \mathcal{I} \times A \rightarrow \mathcal{I}$ appelée **transition enveloppe** : pour tout $[m, M] \in \mathcal{I}$ et $a \in A$

$$[m, M] \square a \triangleq \left[\inf_{m \preceq x \preceq M} \{x \cdot a\}, \sup_{m \preceq x \preceq M} \{x \cdot a\} \right].$$

Extention aux mots finis :

$$[m, M] \square u_{1 \rightarrow n} \triangleq [m, M] \square u_1 \square u_2 \square \cdots \square u_n.$$

Envelope Perfect Sampling Algorithm

Soient $\perp \triangleq \inf \mathcal{X}$ et $\top \triangleq \sup \mathcal{X}$. Le processus $([m_n, M_n])_{n \in \mathbb{N}}$

$$[m_n, M_n] \triangleq [\perp, \top] \square u_{1 \rightarrow n}$$

est une chaîne de Markov sur l'espace d'états $\mathcal{X} \times \mathcal{X}$, appelée **la chaîne enveloppe**.

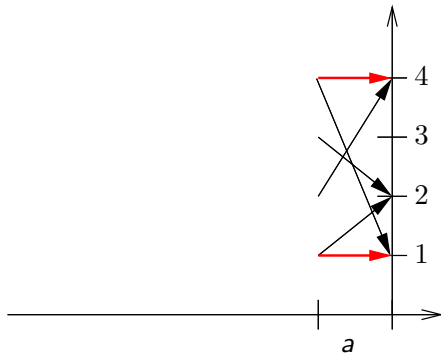
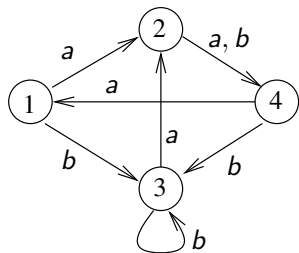
Théorème

Si $[m_n, M_n]$ appartient à $\mathcal{S} = \{[x, x] : x \in \mathcal{X}\}$ pour un n fini :

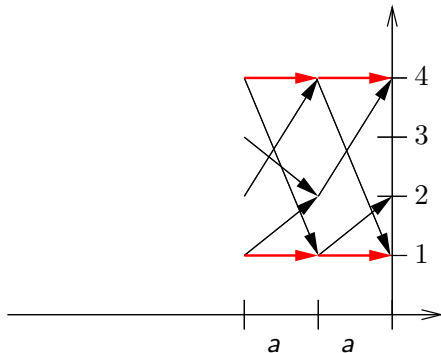
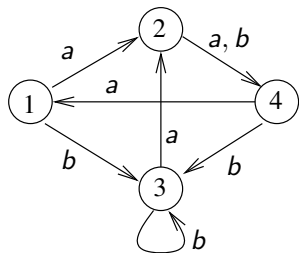
$$\tau_e \triangleq \min \left\{ n : [\perp, \top] \square u_{-n+1 \rightarrow 0} \in \mathcal{S} \right\},$$

alors τ_e est le temps de couplage depuis le passé de la chaîne enveloppe.
L'état défini par $[\perp, \top] \square u_{-\tau_e+1 \rightarrow 0}$ est distribué selon π .

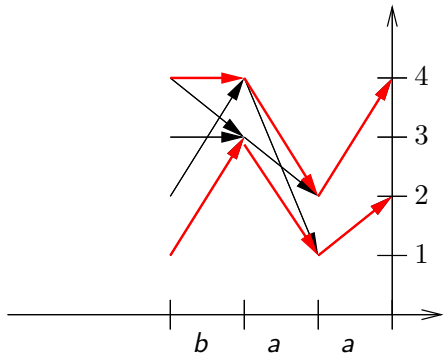
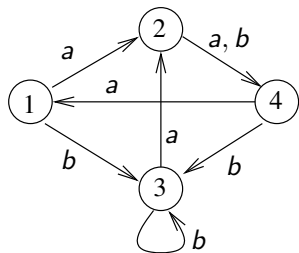
Exemple



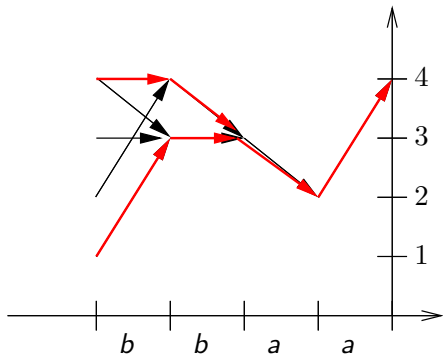
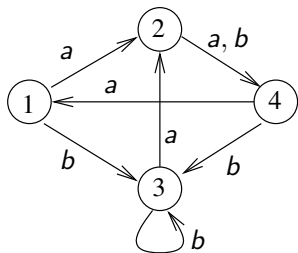
Exemple



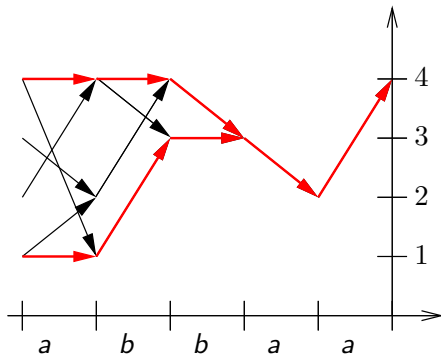
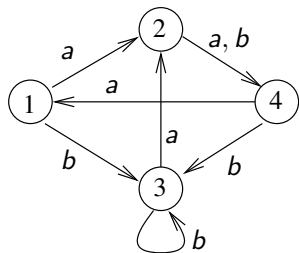
Exemple



Exemple



Exemple



Envelope Perfect Sampling Algorithm (EPSA)

Data : suite i.i.d. $(u_{-n})_{n \in \mathbb{N}} \in A^{\mathbb{N}}$; l'opération enveloppe \boxplus

Result : Un état de \mathcal{X} distribué selon π .

begin

$n = 1$; **repeat**

$m := \perp$; $M := \top$;

for $i = n - 1$ **downto** 0 **do**

$[m, M] := [m, M] \boxplus u_{-i}$;

end

$n := 2n$;

until $m = M$;

$x^* := m$;

return x^* ;

end

Complexité : $O(\mathcal{C}_e \times \tau_e)$ (à comparer avec $O(\mathcal{C} \times |\mathcal{X}| \times \tau^b)$).

Commentaires

1. L'opération \boxtimes peut être remplacée par une sur-approximation \odot telle que pour tout intervalle $[m, M]$, $[m, M] \boxtimes a \subseteq [m, M] \odot a$, sans modifier la validité de l'algorithme.

Commentaires

1. L'opération \boxdot peut être remplacée par une sur-approximation \odot telle que pour tout intervalle $[m, M]$, $[m, M] \boxdot a \subseteq [m, M] \odot a$, sans modifier la validité de l'algorithme.
2. La définition de l'opération enveloppe dépend de la description SED de la chaîne de Markov.

Commentaires

1. L'opération \boxtimes peut être remplacée par une sur-approximation \odot telle que pour tout intervalle $[m, M]$, $[m, M] \boxtimes a \subseteq [m, M] \odot a$, sans modifier la validité de l'algorithme.
2. La définition de l'opération enveloppe dépend de la description SED de la chaîne de Markov.
3. Si un événement $a \in A$ est monotone, alors pour tout $m \preceq M$, $[m, M] \boxtimes a = [m \cdot a, M \cdot a]$.
Si tous les événements sont monotones (ou anti-monotones), alors EPSA coïncide avec la simulation parfaite classique - cas monotone.

Problèmes

- ▶ Il est possible que les enveloppes ne couplent pas même si les vraies trajectoires couplent.

Problèmes

- ▶ Il est possible que les enveloppes ne couplent pas même si les vraies trajectoires couplent.
- ▶ Quand les enveloppes couplent, leur temps de couplage peut être plus long que celui des trajectoires.

Problèmes

- ▶ Il est possible que les enveloppes ne couplent pas même si les vraies trajectoires couplent.
- ▶ Quand les enveloppes couplent, leur temps de couplage peut être plus long que celui des trajectoires.
- ▶ La complexité du calcul des enveloppes peut être trop grande.
Complexité de EPSA : $O(\mathcal{C}_e \times \tau_e)$.
 \mathcal{C}_e ne peut pas être dans $\Omega(N)$ - sinon on ne gagne rien !

Simulation de files d'attentes non monotones

Dans le cas des files d'attentes, les événements sont “homogènes en espace” par partie (i.e. $x \cdot a = x + v_R$ pour x dans la région R) et on a souvent $\mathcal{C} \sim \mathcal{C}_e$. [B., Gaujal, Pin, 2012].

La différence entre la méthode classique et celle des enveloppes se joue donc sur la comparaison entre $|\mathcal{X}| \times \tau$ et τ_e .

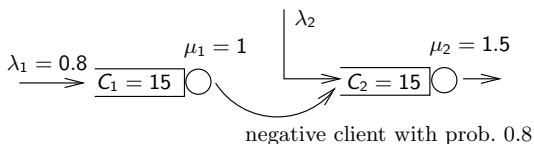


FIGURE – Un réseaux de files avec clients négatifs.

Exemple

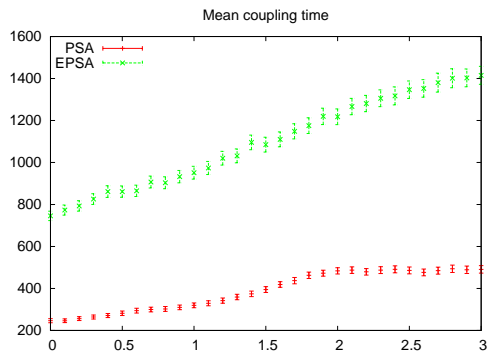


FIGURE – Temps moyen de couplage en fonction de λ_2 .

Files d'attente avec arrivée par lot (batch)

Les clients arrivent et partent par lot. un nouveau lot est accepté si il y a suffisamment de place dans la file. Sinon l'intégralité du lot est rejeté.

Files d'attente avec arrivée par lot (batch)

Les clients arrivent et partent par lot. un nouveau lot est accepté si il y a suffisamment de place dans la file. Sinon l'intégralité du lot est rejeté.

- ▶ Les événements 'lot' sont non-monotones.
(Exemple : l'arrivée par lot de taille 2 dans une file de capacité C :
 $C - 2 < C - 1$ mais $C > C - 1$.)

Files d'attente avec arrivée par lot (batch)

Les clients arrivent et partent par lot. un nouveau lot est accepté si il y a suffisamment de place dans la file. Sinon l'intégralité du lot est rejeté.

- ▶ Les événements 'lot' sont non-monotones.
(Exemple : l'arrivée par lot de taille 2 dans une file de capacité C : $C - 2 < C - 1$ mais $C > C - 1$.)
- ▶ Prop. EPSA couple si et seulement si les lots de taille 1 peuvent survenir avec une probabilité positive dans chaque file.

Files d'attente avec arrivée par lot (batch)

Les clients arrivent et partent par lot. un nouveau lot est accepté si il y a suffisamment de place dans la file. Sinon l'intégralité du lot est rejeté.

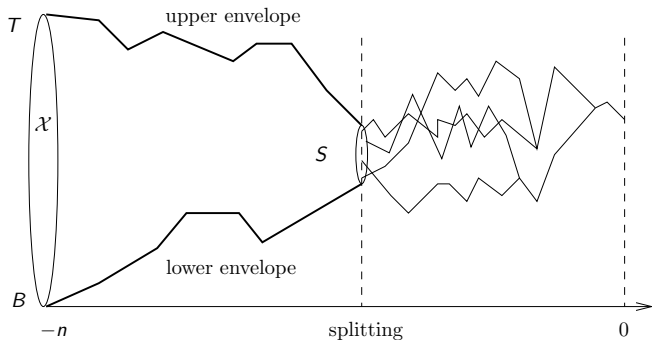
- ▶ Les événements 'lot' sont non-monotones.
(Exemple : l'arrivée par lot de taille 2 dans une file de capacité C :
 $C - 2 < C - 1$ mais $C > C - 1$.)
- ▶ Prop. EPSA couple si et seulement si les lots de taille 1 peuvent survenir avec une probabilité positive dans chaque file.
- ▶ Les enveloppes peuvent être calculées en temps constant :

$$m' = \begin{cases} m + k, & M \leq C - k \\ (m + k) \wedge ((C + k - 1) \vee m), & M > C - k. \end{cases}$$
$$M' = \begin{cases} (M + k) \wedge C, & m \leq C - k \\ M, & m > C - k \end{cases}$$

Au delà des enveloppes

Dans le cas où les enveloppes couplent beaucoup plus lentement que les trajectoires elles-mêmes (ou si elles ne couplent pas) :

- ▶ bornes
- ▶ splitting - on peut combiner les avantages des deux méthodes en faisant un éclatement.



Exemple

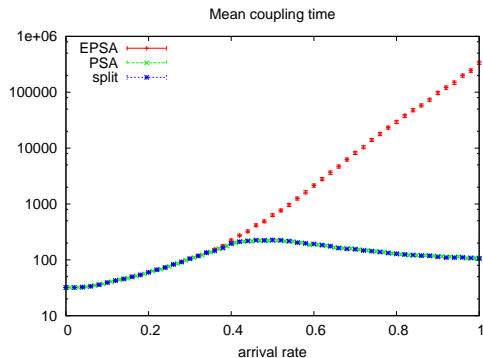


FIGURE – Temps moyen de couplage de PSA, EPSA et EPSA avec éclatement pour une file $(+2, +3, -1)$.

Partition de l'espace d'états

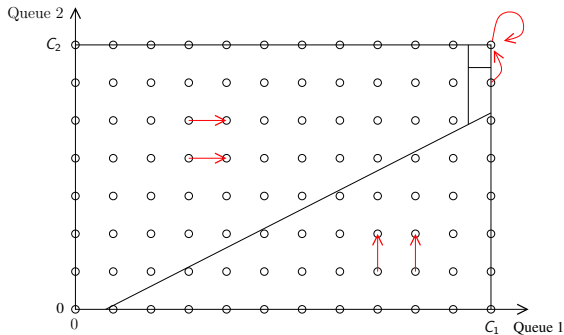
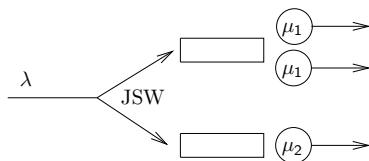


FIGURE – Les zones pour l'événement JSW quand $2\mu_1 > \mu_2$.

Partition de l'espace d'états

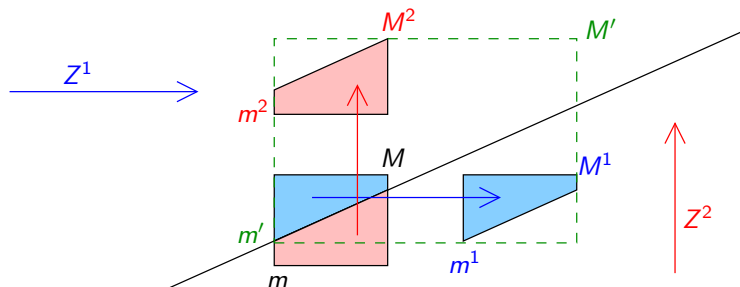


FIGURE – Construction des enveloppes. L'intervalle $[m, M]$ intersecte 2 zones Z^1 and Z^2 . Calcul de $[m', M'] \triangleq [m, M] \boxplus a : [m^1, M^1] = \llbracket [m, M] \cap Z^1 \cdot a \rrbracket$, $[m^2, M^2] = \llbracket [m, M] \cap Z^2 \cdot a \rrbracket$. Enfin, $[m', M'] = \llbracket [m^1, M^1] \cup [m^2, M^2] \rrbracket$.

[B., Gaujal, Pin, 2012]

Partition de l'espace d'états

$$\begin{aligned} [m, M] \boxtimes a &= \left[[m, M] \cdot a \right] \\ &= \left[\left(\bigcup_{k=1}^K [m, M] \cap Z^k \right) \cdot a \right] \\ &= \left[\bigcup_{k=1}^K ([m, M] \cap Z^k \cdot a^k) \right] \\ &= \left[\bigcup_{k=1}^K \left[[m, M] \cap Z^k \cdot a^k \right] \right] \end{aligned} \tag{1}$$

$$\subset \left[\bigcup_{k=1}^K \left[[m, M] \cap Z^k \right] \boxtimes a^k \right]. \tag{2}$$

Chaînes de Markov réversibles

Soit $\{X_t\}_{t \geq 0}$ une chaîne de Markov avec l'espace d'états $S = \{s_1, \dots, s_n\}$ et la matrice de transition P . Une distribution de probabilités π sur S est dite réversible si pour tout $i, j \in \{1, \dots, n\}$,

$$\pi_i P_{i,j} = \pi_j P_{j,i}.$$

Theorem

Si π est une distribution réversible, alors elle est aussi une distribution stationnaire.

Démonstration.

Pour tout $j \in \{1, \dots, n\}$, on a

$$\pi_j = \pi_j \sum_{i=1}^n P_{j,i} = \sum_{i=1}^n \pi_j P_{j,i} = \sum_{i=1}^n \pi_i P_{i,j}.$$



Exemple : processus de naissance et de mort

Matrice de transition P telle que :

- ▶ $P_{i,j} > 0$ si $|i - j| = 1$,
- ▶ $P_{i,j} = 0$ si $|i - j| \geq 2$.

Notons par

$$\pi_i^* = \frac{\prod_{k=1}^{i-1} P_{k,k+1}}{\prod_{k=1}^{i-1} P_{k+1,k}}.$$

Alors,

$$\pi = (\pi_1, \dots, \pi_n) = (\pi_1^*, \dots, \pi_n^*) / \left(\sum_{i=1}^n \pi_i^* \right)$$

est une distribution de probabilités réversible.

Exemple : marche aléatoire sur un graphe

Un graphe $G = (V, E)$ avec les sommets $V = \{v_1, \dots, v_n\}$ et les arêtes $E = \{e_1, \dots, e_m\}$. Soit d_i le degré du sommet v_i .

Un marcheur qui est au sommet v_i à l'instant t change sa position pour un sommet voisin de v_i avec la même probabilité pour tous les voisins.

Matrice de transition P :

$$P_{i,j} = 1/d_i, \text{ si } (v_i, v_j) \in E$$

et sinon $P_{i,j} = 0$.

Alors,

$$\pi = (d_1, \dots, d_n) / \left(\sum_{i=1}^n d_i \right)$$

est une distribution de probabilités réversible.

Echantillonneur de Gibbs

$G = (V, E)$ un graphe avec les sommets $V = \{v_1, \dots, v_n\}$ et les arêtes $E = \{e_1, \dots, e_m\}$. $N(v)$ les voisins du sommet v .
 C un ensemble fini - l'ensemble des valeurs pour les sommets.

Echantillonneur de Gibbs

$G = (V, E)$ un graphe avec les sommets $V = \{v_1, \dots, v_n\}$ et les arêtes $E = \{e_1, \dots, e_m\}$. $N(v)$ les voisins du sommet v .

C un ensemble fini - l'ensemble des valeurs pour les sommets.

L'espace d'états : $\mathcal{X} = C^V$.

But : échantillonner une distribution π sur \mathcal{X} .

Echantillonneur de Gibbs

$G = (V, E)$ un graphe avec les sommets $V = \{v_1, \dots, v_n\}$ et les arêtes $E = \{e_1, \dots, e_m\}$. $N(v)$ les voisins du sommet v .

C un ensemble fini - l'ensemble des valeurs pour les sommets.

L'espace d'états : $\mathcal{X} = C^V$.

But : échantillonner une distribution π sur \mathcal{X} .

Idée : créer une chaîne de Markov qui a la distribution stationnaire π .

Echantillonneur de Gibbs

$G = (V, E)$ un graphe avec les sommets $V = \{v_1, \dots, v_n\}$ et les arêtes $E = \{e_1, \dots, e_m\}$. $N(v)$ les voisins du sommet v .

C un ensemble fini - l'ensemble des valeurs pour les sommets.

L'espace d'états : $\mathcal{X} = C^V$.

But : échantillonner une distribution π sur \mathcal{X} .

Idée : créer une chaîne de Markov qui a la distribution stationnaire π .

Algorithme

1. Choisir un sommet $v \in V$ uniformément.

Echantillonneur de Gibbs

$G = (V, E)$ un graphe avec les sommets $V = \{v_1, \dots, v_n\}$ et les arêtes $E = \{e_1, \dots, e_m\}$. $N(v)$ les voisins du sommet v .

C un ensemble fini - l'ensemble des valeurs pour les sommets.

L'espace d'états : $\mathcal{X} = C^V$.

But : échantillonner une distribution π sur \mathcal{X} .

Idée : créer une chaîne de Markov qui a la distribution stationnaire π .

Algorithme

1. Choisir un sommet $v \in V$ uniformément.
2. Choisir $X_{t+1}(v)$ selon la loi π sachant les valeurs de tous les sommets voisins :

$$p(X_{t+1}(v)) = \pi(X_{t+1}(v) \mid X_{t+1}(w) = X_t(w), \forall w \in N(v))$$

Echantillonneur de Gibbs

$G = (V, E)$ un graphe avec les sommets $V = \{v_1, \dots, v_n\}$ et les arêtes $E = \{e_1, \dots, e_m\}$. $N(v)$ les voisins du sommet v .

C un ensemble fini - l'ensemble des valeurs pour les sommets.

L'espace d'états : $\mathcal{X} = C^V$.

But : échantillonner une distribution π sur \mathcal{X} .

Idée : créer une chaîne de Markov qui a la distribution stationnaire π .

Algorithme

1. Choisir un sommet $v \in V$ uniformément.
2. Choisir $X_{t+1}(v)$ selon la loi π sachant les valeurs de tous les sommets voisins :

$$p(X_{t+1}(v)) = \pi(X_{t+1}(v) \mid X_{t+1}(w) = X_t(w), \forall w \in N(v))$$

3. $X_{t+1}(w) = X_t(w), \forall w \neq v$.

Exemple : coloriage propres

$G = (V, E)$ un graphe avec les sommets $V = \{v_1, \dots, v_n\}$ et les arêtes $E = \{e_1, \dots, e_m\}$. $N(v)$ les voisins du sommet v .

Exemple : coloriage propres

$G = (V, E)$ un graphe avec les sommets $V = \{v_1, \dots, v_n\}$ et les arêtes $E = \{e_1, \dots, e_m\}$. $N(v)$ les voisins du sommet v .

$C = \{1, \dots, q\}$. L'espace d'états : $\mathcal{X} \subset C^V$.

Un coloriage propre : pas de sommets voisins avec la même couleur.

Exemple : coloriage propres

$G = (V, E)$ un graphe avec les sommets $V = \{v_1, \dots, v_n\}$ et les arêtes $E = \{e_1, \dots, e_m\}$. $N(v)$ les voisins du sommet v .

$C = \{1, \dots, q\}$. L'espace d'états : $\mathcal{X} \subset C^V$.

Un coloriage propre : pas de sommets voisins avec la même couleur.

Algorithme

1. Choisir un sommet $v \in V$ uniformément.

Exemple : coloriage propres

$G = (V, E)$ un graphe avec les sommets $V = \{v_1, \dots, v_n\}$ et les arêtes $E = \{e_1, \dots, e_m\}$. $N(v)$ les voisins du sommet v .

$C = \{1, \dots, q\}$. L'espace d'états : $\mathcal{X} \subset C^V$.

Un coloriage propre : pas de sommets voisins avec la même couleur.

Algorithme

1. Choisir un sommet $v \in V$ uniformément.
2. Choisir $X_{t+1}(v)$ uniformément parmi toutes les couleurs autorisées (pas de sommet voisin de cette couleur).

Exemple : coloriage propres

$G = (V, E)$ un graphe avec les sommets $V = \{v_1, \dots, v_n\}$ et les arêtes $E = \{e_1, \dots, e_m\}$. $N(v)$ les voisins du sommet v .

$C = \{1, \dots, q\}$. L'espace d'états : $\mathcal{X} \subset C^V$.

Un coloriage propre : pas de sommets voisins avec la même couleur.

Algorithme

1. Choisir un sommet $v \in V$ uniformément.
2. Choisir $X_{t+1}(v)$ uniformément parmi toutes les couleurs autorisées (pas de sommet voisin de cette couleur).
3. $X_{t+1}(w) = X_t(w), \forall w \neq v$.

Question

Quelle est la loi π et montrer qu'elle est réversible.

Exemple : ensembles indépendants d'un graphe

$G = (V, E)$ un graphe avec les sommets $V = \{v_1, \dots, v_n\}$ et les arêtes $E = \{e_1, \dots, e_m\}$. $N(v)$ les voisins du sommet v .

Un ensemble indépendant $I \subset V$ est un ensemble tel qu'il n'y a pas d'arêtes dans E reliant deux sommets dans I .

Exemple : ensembles indépendants d'un graphe

$G = (V, E)$ un graphe avec les sommets $V = \{v_1, \dots, v_n\}$ et les arêtes $E = \{e_1, \dots, e_m\}$. $N(v)$ les voisins du sommet v .

Un ensemble indépendant $I \subset V$ est un ensemble tel qu'il n'y a pas d'arêtes dans E reliant deux sommets dans I .

$C = \{0, 1\}$. L'espace d'états : $\mathcal{X} \subset C^V$.

L'état $x \in C^V$, $x(v) = 1$ ssi $v \in I$.

Exemple : ensembles indépendants d'un graphe

$G = (V, E)$ un graphe avec les sommets $V = \{v_1, \dots, v_n\}$ et les arêtes $E = \{e_1, \dots, e_m\}$. $N(v)$ les voisins du sommet v .

Un ensemble indépendant $I \subset V$ est un ensemble tel qu'il n'y a pas d'arêtes dans E reliant deux sommets dans I .

$C = \{0, 1\}$. L'espace d'états : $\mathcal{X} \subset C^V$.

L'état $x \in C^V$, $x(v) = 1$ ssi $v \in I$.

Algorithme

1. Choisir un sommet $v \in V$ uniformément.

Exemple : ensembles indépendants d'un graphe

$G = (V, E)$ un graphe avec les sommets $V = \{v_1, \dots, v_n\}$ et les arêtes $E = \{e_1, \dots, e_m\}$. $N(v)$ les voisins du sommet v .

Un ensemble indépendant $I \subset V$ est un ensemble tel qu'il n'y a pas d'arêtes dans E reliant deux sommets dans I .

$C = \{0, 1\}$. L'espace d'états : $\mathcal{X} \subset C^V$.

L'état $x \in C^V$, $x(v) = 1$ ssi $v \in I$.

Algorithme

1. Choisir un sommet $v \in V$ uniformément.
2. Jeter une pièce non-biaisée. Si "tête" et tous les sommets voisins de v ont la valeur 0, alors $X_{t+1}(v) = 1$, sinon $X_{t+1}(v) = 0$.

Exemple : ensembles indépendants d'un graphe

$G = (V, E)$ un graphe avec les sommets $V = \{v_1, \dots, v_n\}$ et les arêtes $E = \{e_1, \dots, e_m\}$. $N(v)$ les voisins du sommet v .

Un ensemble indépendant $I \subset V$ est un ensemble tel qu'il n'y a pas d'arêtes dans E reliant deux sommets dans I .

$C = \{0, 1\}$. L'espace d'états : $\mathcal{X} \subset C^V$.

L'état $x \in C^V$, $x(v) = 1$ ssi $v \in I$.

Algorithme

1. Choisir un sommet $v \in V$ uniformément.
2. Jeter une pièce non-biaisée. Si "tête" et tous les sommets voisins de v ont la valeur 0, alors $X_{t+1}(v) = 1$, sinon $X_{t+1}(v) = 0$.
3. $X_{t+1}(w) = X_t(w), \forall w \neq v$.

Exemple : ensembles indépendants d'un graphe

Question

Quelle est la loi π et montrer qu'elle est réversible.

Question

Modifier l'algorithme pour favoriser les ensembles indépendants de grande cardinalité :

$$\pi_\lambda(x) = \frac{\lambda^{|x|}}{Z_\lambda}$$

pour toutes les configurations admissibles $x \subset C^V$ (et $\pi_\lambda(x) = 0$ si x n'est pas un ensemble indépendant). $|x|$ est la cardinalité de x (nombre de sommets avec la valeur 1) et Z_λ est la constante de normalisation.

Exemple : ensembles indépendants d'un graphe

Une chaîne de Markov irréductible et apériodique :

Exemple : ensembles indépendants d'un graphe

Une chaîne de Markov irréductible et apériodique :

- ▶ Tout indépendant I est accessible à partir de l'indépendant vide : on choisit successivement tous les sommets de I , que l'on ajoute (c'est possible avec probabilité $1/2$).

Exemple : ensembles indépendants d'un graphe

Une chaîne de Markov irréductible et apériodique :

- ▶ Tout indépendant I est accessible à partir de l'indépendant vide : on choisit successivement tous les sommets de I , que l'on ajoute (c'est possible avec probabilité $1/2$).
- ▶ De la même manière (on retire les sommets), l'indépendant vide est accessible à partir de tous les indépendants.

Exemple : ensembles indépendants d'un graphe

Une chaîne de Markov irréductible et apériodique :

- ▶ Tout indépendant I est accessible à partir de l'indépendant vide : on choisit successivement tous les sommets de I , que l'on ajoute (c'est possible avec probabilité $1/2$).
- ▶ De la même manière (on retire les sommets), l'indépendant vide est accessible à partir de tous les indépendants.
- ▶ De plus, on a $\mathbf{P}(x, x) \geq 1/2$ pour tout x , donc la chaîne est bien apériodique.

Exemple : ensembles indépendants d'un graphe

Une chaîne de Markov irréductible et apériodique :

- ▶ Tout indépendant I est accessible à partir de l'indépendant vide : on choisit successivement tous les sommets de I , que l'on ajoute (c'est possible avec probabilité $1/2$).
- ▶ De la même manière (on retire les sommets), l'indépendant vide est accessible à partir de tous les indépendants.
- ▶ De plus, on a $\mathbf{P}(x, x) \geq 1/2$ pour tout x , donc la chaîne est bien apériodique.

La distribution stationnaire est bien la distribution uniforme :

Exemple : ensembles indépendants d'un graphe

Une chaîne de Markov irréductible et apériodique :

- ▶ Tout indépendant I est accessible à partir de l'indépendant vide : on choisit successivement tous les sommets de I , que l'on ajoute (c'est possible avec probabilité $1/2$).
- ▶ De la même manière (on retire les sommets), l'indépendant vide est accessible à partir de tous les indépendants.
- ▶ De plus, on a $\mathbf{P}(x, x) \geq 1/2$ pour tout x , donc la chaîne est bien apériodique.

La distribution stationnaire est bien la distribution uniforme :

Si I_1 et I_2 sont deux indépendants qui ne diffèrent que d'un sommets, par exemple $I_1 = I_2 \cup \{u\}$, on a

$$\mathbf{P}(x, y) = \frac{1}{n} \times \frac{1}{2} = \mathbf{P}(y, x),$$

ce qui permet de conclure.

Chaînes bornantes et la simulation parfaite

[Huber, 2004]

L'espace d'états : $\mathcal{X} \subset C^V$.

Chaîne $\{X_n\}$ évolue dans \mathcal{X} .

Chaîne $\{Y_n\}$ évolue dans $(2^C)^V$.

Chaîne $\{Y_n\}$ est une chaîne bornante pour $\{X_n\}$ s'il existe un couplage entre $\{X_n\}$ et $\{Y_n\}$ tel que :

$$X_n(v) \in Y_n(v), \quad \forall v \in V \Rightarrow X_{n+1}(v) \in Y_{n+1}(v), \quad \forall v \in V.$$

Construction :

$$Y_{n+1}(v) = \bigcup_{x \in V} Y_n(\phi(x))(v),$$

avec $x \in V$ y si $x(v) \in y(v), \forall v \in V$.

Exemple : coloriage propres

Coloriage propre d'un graphe (V, E) : $x \in C^V$ tel que pour tout $v, w \in E$, $x(v) \neq x(w)$.

Gibbs sampler :

1. Choisir v uniformément dans $V = \{1, \dots, n\}$
2. Choisir c uniformément dans $C = \{1, \dots, k\} \setminus x(N_v)$
3. $x(v) \leftarrow c$

Example : coloriage propres

Coloriage propre d'un graphe (V, E) : $x \in C^V$ tel que pour tout $v, w \in E$, $x(v) \neq x(w)$.

Gibbs sampler :

1. Choisir v uniformément dans $V = \{1, \dots, n\}$
2. Choisir c uniformément dans $C = \{1, \dots, k\} \setminus x(N_v)$
3. $x(v) \leftarrow c$

Borne :

1. Choisir v uniformément dans $V = \{1, \dots, n\}$, $y(v) \leftarrow \emptyset$
2. Répéter :

Choisir c uniformément dans $C = \{1, \dots, k\}$

Si pas de $w \in N_v$ tel que $y(w) = \{c\}$,

alors $y(v) \leftarrow y(v) \cup \{c\}$

tant que $c \notin \cup_{w \in N_v} y(w)$ ou $|y(v)| > \Delta$

Example : coloriage propres

Soit Δ le degré maximum du graphe.

Théorème. (Huber '04) Si $k \geq \Delta(\Delta + 2)$, alors la probabilité que la chaîne bornante détecte le couplage en $\log_{\beta} n + \theta$ est au moins $1 - \beta^{\theta}$, avec

$$\beta = 1 - \frac{1 - (\Delta + 1)\Delta / (k - \Delta + 1)}{n}.$$

Variante

L'espace d'états : $\mathcal{X} \subset C^V$.

Chaîne $\{X_n\}$ évolue dans \mathcal{X} .

Chaîne $\{Y_n\}$ évolue dans $C(2^V)$. Décrite par $\{(B_n, D_n)\}$, $B_n, D_n \in C(2^V)$.

Chaîne $\{Y_n\}$ est une chaîne bornante pour $\{X_n\}$ s'il existe un couplage entre $\{X_n\}$ et $\{Y_n\}$ tel que : $B_n(c) \subset A_n(c) \subset B_n(c) \cup D_n(c)$, $\forall c \in C$
 $\Rightarrow B_{n+1}(c) \subset A_{n+1}(c) \subset B_{n+1}(c) \cup D_{n+1}(c)$, $\forall c \in C$.

Notation : $a \in_C (b, d)$ si $b(c) \subset a(c) \subset b(c) \cup d(c)$, $\forall c \in C$.

Construction :

$$B_{n+1}(c) = \bigcap_{a \in_C (B_n, D_n)} (\phi(a))(c),$$

$$D_{n+1}(c) = \left(\bigcup_{a \in_C (B_n, D_n)} (\phi(a))(c) \right) \setminus B_{n+1}(c),$$

Example : ensembles indépendants

Distribution sur les ensembles indépendants :

$$\pi(A) = \frac{\lambda^{|A|}}{Z_\lambda}.$$

Dyer et Greenhil (2000) chaîne de Markov
(variante de Gibbs sampler) :

1. Choisir v uniformement dans $\{1, \dots, n\}$
2. Choisir U uniformement dans $[0, 1]$
3. Cas I : Si $U > \frac{\lambda}{(\lambda+1)}$, $A \leftarrow A \setminus \{v\}$
4. Cas II : Si $U < \frac{\lambda}{(\lambda+1)}$ et pas de voisins de v dans A , alors $A \leftarrow A \cup \{v\}$

Example : ensembles indépendants

Distribution sur les ensembles indépendants :

$$\pi(A) = \frac{\lambda^{|A|}}{Z_\lambda}.$$

Dyer et Greenhil (2000) chaîne de Markov
(variante de Gibbs sampler) :

1. Choisir v uniformement dans $\{1, \dots, n\}$
2. Choisir U uniformement dans $[0, 1]$
3. Cas I : Si $U > \frac{\lambda}{(\lambda+1)}$, $A \leftarrow A \setminus \{v\}$
4. Cas II : Si $U < \frac{\lambda}{(\lambda+1)}$ et pas de voisins de v dans A , alors $A \leftarrow A \cup \{v\}$
5. Cas III : Si $U < p_{\text{swap}} \frac{\lambda}{(\lambda+1)}$ et exactement un voisin w de v dans A , alors $A \leftarrow (A \setminus \{w\}) \cup \{v\}$

Exemple : ensembles indépendants

Soit $\Delta > 2$ le degré maximum du graphe.

Théorème. Si $\lambda < \frac{2}{\Delta-2}$, le temps de couplage de la chaîne bornante est inférieur à $\log_{\beta} n + \theta$ avec probabilité au moins $1 - \beta^{\theta}$, avec

$$\beta = \frac{\Delta\lambda}{2(\lambda + 1)}.$$

Algorithme Metropolis

Objectif : Échantillonner une distribution π sur $S = \{s_1, \dots, s_n\}$.

L'idée : Construire un graphe de voisinage entre les états.

Hypothèses :

- ▶ Le graphe G est connexe.
- ▶ Les degrés des sommets sont petits.

$$P_{i,j} = \begin{cases} \frac{1}{d_i} \min\left\{\frac{\pi_j d_i}{\pi_i d_j}, 1\right\}, & (v_i, v_j) \in E \\ 0, & i \neq j, (v_i, v_j) \in E \\ 1 - \sum_{\ell \mid (s_\ell, s_i) \in E} \min\left\{\frac{\pi_j d_i}{\pi_i d_j}, 1\right\}, & i = j \end{cases}$$

Question

Montrer que π est la distribution réversible (et donc stationnaire) de P .

Algorithme Metropolis

Algorithme :

Soit $X_t = s_j$. Alors,

1. Choisir un voisin de s_j uniformement parmi les voisins.
- 2.

$$X_{t+1} = \begin{cases} s_j, & \text{avec probabilité } \min\left\{\frac{\pi_j d_i}{\pi_i d_j}, 1\right\} \\ s_i, & \text{avec probabilité } 1 - \min\left\{\frac{\pi_j d_i}{\pi_i d_j}, 1\right\} \end{cases}$$

L'optimisation

Problème :

- ▶ $\mathcal{X} = \{s_1, \dots, s_n\}$.
- ▶ On cherche à minimiser une fonction $f : \mathcal{X} \rightarrow \mathbb{R}$.

$T > 0$ temperature.

Distribution de Boltzmann :

$$\pi_{f,T}(s) = \frac{1}{Z_{f,T}} \exp\left(\frac{-f(s)}{T}\right).$$

avec $Z_{f,T}$ constante de normalisation.

L'optimisation : recuit simulé

Théorème

Soit $\alpha(T)$ la probabilité qu'un élément choisi selon $\pi_{f,T}$ vérifie

$$f(Y) = \min_{s \in \mathcal{X}} f(s)$$

Alors

$$\lim_{T \rightarrow 0} \alpha(T) = 1.$$

Question

Montrer le théorème sous l'hypothèse que f a un unique minimum.
Et si le minimum n'est pas unique ?

Recuit simulé : Refroidir progressivement le système.

Garanties théoriques pour les schémas de refroidissement très lents : en $O(1/\log(n))$.

En pratique : $1/n$ (cvg. vers un minimum local).