# Exemple de chaîne de Markov à espace d'état discret – II

1) Selon vous, la chaîne est-elle irréductible ? Récurrente ? Apériodique ?

. . .

2) Supposons que Bébé dorme. Que fait-il 2 min après ? et 10 min après ?

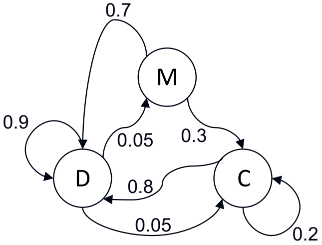
. . .

3) Supposons maintenant qu'on lui change la couche. Que fait-il 10 min après ?

. . .

# Exemple de chaîne de Markov à espace d'état discret – II

1) Selon vous, la chaîne est-elle irréductible ? Récurrente ? Apériodique ?




2) Supposons que Bébé dorme. Que fait-il 2 min après ? et 10 min après ?

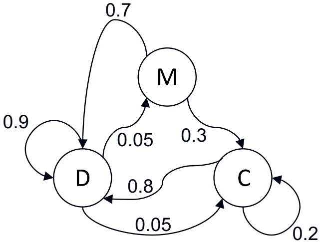
...

3) Supposons maintenant qu'on lui change la couche. Que fait-il 10 min après ?

...

# Exemple de chaîne de Markov à espace d'état discret – II

1) Selon vous, la chaîne est-elle irréductible ? Récurrente ? Apériodique ?




2) Supposons que Bébé dorme. Que fait-il 2 min après ? et 10 min après ?

$$x_0 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}^T \quad x_2 = x_0 P^2 = \begin{pmatrix} 0.885 \\ 0.045 \\ 0.070 \end{pmatrix}^T \quad x_{10} = x_2 P^8 = x_0 P^{10} = \begin{pmatrix} 0.884 \\ 0.044 \\ 0.072 \end{pmatrix}^T$$

# Exemple de chaîne de Markov à espace d'état discret – II

3) Supposons maintenant qu'on lui change la couche. Que fait-il 10 min après ?

...

# Exemple de chaîne de Markov à espace d'état discret – II

3) Supposons maintenant qu'on lui change la couche. Que fait-il 10 min après ?

$$x_0 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}^T \qquad x_{10} = x_0 P^{10} = \begin{pmatrix} 0.884 \\ 0.044 \\ 0.072 \end{pmatrix}^T$$

Ici la chaîne est apériodique, récurrente et irréductible, il y a donc une loi stationnaire : $\tilde{p} = \tilde{p}P$.

# Algorithmes MCMC : principe général

Approximer une intégrale (ou d'une autre fonctionnelle)
d'une distribution d'intérêt

# Algorithmes MCMC : principe général

Approximer une intégrale (ou d'une autre fonctionnelle) d'une distribution d'intérêt

$\Rightarrow$ générer une chaîne de Markov dont la loi stationnaire est la loi *a posteriori*, puis d'y appliquer la méthode de Monte-Carlo

# Algorithmes MCMC : principe général

Approximer une intégrale (ou d'une autre fonctionnelle) d'une distribution d'intérêt

$\Rightarrow$ générer une chaîne de Markov dont la loi stationnaire est la loi *a posteriori*, puis d'y appliquer la méthode de Monte-Carlo

Nécessite une **double convergence** :

1. la convergence de la chaîne de Markov vers sa distribution stationnaire : $\forall\, X_0\,,\, X_n \xrightarrow[n\to+\infty]{\mathscr{L}} \tilde{p}$

# Algorithmes MCMC : principe général

Approximer une intégrale (ou d'une autre fonctionnelle) d'une distribution d'intérêt

$\Rightarrow$ générer une chaîne de Markov dont la loi stationnaire est la loi *a posteriori*, puis d'y appliquer la méthode de Monte-Carlo

Nécessite une **double convergence** :

1. la convergence de la chaîne de Markov vers sa distribution stationnaire : $\forall\, X_0\,,\, X_n \xrightarrow[n\to+\infty]{\mathcal{L}} \tilde{p}$
2. la convergence de Monte-Carlo, une fois la distribution stationnaire atteinte : $\frac{1}{N}\sum_{i=1}^{N} f(X_{n+i}) \xrightarrow[N\to+\infty]{} \mathbb{E}[f(X)]$

# Algorithmes MCMC : principe général

Approximer une intégrale (ou d'une autre fonctionnelle) d'une distribution d'intérêt

$\Rightarrow$ générer une chaîne de Markov dont la loi stationnaire est la loi *a posteriori*, puis d'y appliquer la méthode de Monte-Carlo

Nécessite une **double convergence** :

1. la convergence de la chaîne de Markov vers sa distribution stationnaire : $\forall\, X_0\,,\; X_n \xrightarrow[n\to+\infty]{\mathscr{L}} \tilde{p}$
2. la convergence de Monte-Carlo, une fois la distribution stationnaire atteinte : $\frac{1}{N}\sum_{i=1}^{N} f(X_{n+i}) \xrightarrow[N\to+\infty]{} \mathbb{E}[f(X)]$

$$\overbrace{X_0 \to X_1 \to X_2 \to \cdots \to X_n}^{\text{convergence de la chaîne de Markov}} \;\to \overbrace{X_{n+1} \to X_{n+2} \to \cdots \to X_{n+N}}^{\text{échantillon de Monte-Carlo}}$$

# Schéma général des algorithmes MCMC

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

1. Initialiser $x^{(0)}$
2. Pour $t = 1, \dots, n+N$ :
   a. Proposer un nouveau candidat $y^{(t)} \sim q(y^{(t)} | x^{(t-1)})$
   b. Accepter $y^{(t)}$ avec la probabilité $\alpha(x^{(t-1)}, y^{(t)})$ :
      $x^{(t)} := y^{(t)}$

   $\Rightarrow$ si $t > n$, « sauver » $x^{(t)}$ (pour ensuite calculer la fonctionnelle d'intérêt)

avec $q$ la loi instrumentale de proposition
et $\alpha$ la probabilité d'acceptation

# Schéma général des algorithmes MCMC

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

1. Initialiser $x^{(0)}$
2. Pour $t = 1, \ldots, n+N$ :
   a. Proposer un nouveau candidat $y^{(t)} \sim q(y^{(t)}|x^{(t-1)})$
   b. Accepter $y^{(t)}$ avec la probabilité $\alpha(x^{(t-1)}, y^{(t)})$ :
      $x^{(t)} := y^{(t)}$

   $\Rightarrow$ si $t > n$, « sauver » $x^{(t)}$ (pour ensuite calculer la fonctionnelle d'intérêt)

avec $q$ la loi instrumentale de proposition
et $\alpha$ la probabilité d'acceptation

# Choix de la loi instrumentale

**Pas de choix absolument optimal** pour la loi instrumentale de proposition $q$

$\Rightarrow$ infinité de lois possibles : certaines meilleures que d'autres

# Choix de la loi instrumentale

**Pas de choix absolument optimal** pour la loi instrumentale de proposition $q$

$\Rightarrow$ infinité de lois possibles : certaines meilleures que d'autres

Afin de garantir la convergence vers la loi cible $\tilde{p}$ :

- le support de $q$ doit contenir le support $\tilde{p}$
- $q$ ne doit pas générer de valeurs périodiques

# Choix de la loi instrumentale

**Pas de choix absolument optimal** pour la loi instrumentale de proposition $q$

$\Rightarrow$ infinité de lois possibles : certaines meilleures que d'autres

Afin de garantir la convergence vers la loi cible $\tilde{p}$ :

- le support de $q$ doit contenir le support $\tilde{p}$
- $q$ ne doit pas générer de valeurs périodiques

**NB :** *Idéalement* on choisit $q$ de manière à ce que son **calcul** soit **simple** (et **rapide**)

# Algorithme de Metropolis-Hastings

1. Initialiser $x^{(0)}$
2. Pour $t = 1, \ldots, n+N$ :
   a. Proposer $y^{(t)} \sim q(y^{(t)}|x^{(t-1)})$
   b. Calculer la probabilité d'acceptation
   $$\alpha^{(t)} = \min\left\{1, \frac{\tilde{p}(y^{(t)})}{q(y^{(t)}|x^{(t-1)})} \Big/ \frac{\tilde{p}(x^{(t-1)})}{q(x^{(t-1)}|y^{(t)})}\right\}$$
   c. Étape d'acceptation-rejet : générer $u^{(t)} \sim \mathcal{U}_{[0;1]}$
   $$x^{(t)} = \begin{cases} y \text{ si } u^{(t)} \leq \alpha^{(t)} \\ x^{(t-1)} \text{ sinon} \end{cases}$$

$$\alpha^{(t)} = \min\left\{1, \frac{\tilde{p}(y^{(t)})}{\tilde{p}(x^{(t-1)})} \frac{q(x^{(t-1)}|y^{(t)})}{q(y^{(t)}|x^{(t-1)})}\right\}$$

$\Rightarrow$ calculable en ne connaissant $\tilde{p}$ qu'à une constante près !

# Metropolis-Hastings : cas particuliers

On obtient parfois un calcul simplifié pour $\alpha^{(t)}$ :

- **Metropolis-Hastings indépendant** : $q(y^{(t)}|x^{(t-1)}) = q(y^{(t)})$
- **Metropolis-Hastings à marche aléatoire** :
  $q(y^{(t)}|x^{(t-1)}) = g(y^{(t)} - x^{(t-1)})$
  Si $g$ est symétrique $\left(g(-x) = g(x)\right)$, alors :

$$\frac{\tilde{p}(y^{(t)})}{\tilde{p}(x^{(t-1)})} \frac{q(y^{(t)}|x^{(t-1)})}{q(x^{(t-1)}|y^{(t)})} = \frac{\tilde{p}(y^{(t)})}{\tilde{p}(x^{(t-1)})} \frac{\cancel{g(y^{(t)} - x^{(t-1)})}}{\cancel{g(x^{(t-1)} - y^{(t)})}} = \frac{\tilde{p}(y^{(t)})}{\tilde{p}(x^{(t-1)})}$$

# Pour et contre de l'algorithme de Metropolis-Hastings

😁 très simple & très général

😁 permet d'échantillonner selon des lois uni- ou multi-dimensionnelles

☹️ choix de la loi de proposition crucial, mais difficile
$\Rightarrow$ impact considérable sur les performances de l'algorithme

☹️ devient inefficace dans les problèmes de trop grande dimension

**NB :** un fort taux de rejet implique souvent des temps de calculs très importants

# Algorithme du recuit-simulé

Faire varier le calcul de $\alpha^{(t)}$ au cours de l'algorithme :

1. $\alpha^{(t)}$ doit d'abord être grande afin d'explorer l'ensemble de l'espace
2. puis $\alpha^{(t)}$ doit diminuer au fur et à mesure que l'algorithme converge

# Algorithme du recuit-simulé

Faire varier le calcul de $\alpha^{(t)}$ au cours de l'algorithme :

1. $\alpha^{(t)}$ doit d'abord être grande afin d'explorer l'ensemble de l'espace
2. puis $\alpha^{(t)}$ doit diminuer au fur et à mesure que l'algorithme converge

1. Initialiser $x^{(0)}$
2. Pour $t = 1, \dots, n+N$ :
   a. Proposer $y^{(t)} \sim q(y^{(t)}|x^{(t-1)})$
   b. Calculer la probabilité d'acceptation

$$\alpha^{(t)} = \min\left\{1, \left(\frac{\tilde{p}(y^{(t)})}{\tilde{p}(x^{(t-1)})}\frac{q(x^{(t-1)}|y^{(t)})}{q(y^{(t)}|x^{(t-1)})}\right)^{\frac{1}{T(t)}}\right\}$$

   c. Étape d'acceptation-rejet : générer une valeur $u^{(t)} \sim \mathcal{U}_{[0;1]}$

$$x^{(t)} := \begin{cases} y^{(t)} & \text{si } u^{(t)} \leq \alpha^{(t)} \\ x^{(t-1)} & \text{sinon} \end{cases}$$

# Algorithme du recuit-simulé

Faire varier le calcul de $\alpha^{(t)}$ au cours de l'algorithme :

1. $\alpha^{(t)}$ doit d'abord être grande afin d'explorer l'ensemble de l'espace
2. puis $\alpha^{(t)}$ doit diminuer au fur et à mesure que l'algorithme converge

1. Initialiser $x^{(0)}$
2. Pour $t = 1, \dots, n+N$ :
   a. Proposer $y^{(t)} \sim q(y^{(t)}|x^{(t-1)})$
   b. Calculer la probabilité d'acceptation
   $$\alpha^{(t)} = \min\left\{1, \left(\frac{\tilde{p}(y^{(t)})}{\tilde{p}(x^{(t-1)})}\frac{q(x^{(t-1)}|y^{(t)})}{q(y^{(t)}|x^{(t-1)})}\right)^{\frac{1}{T(t)}}\right\}$$
   c. Étape d'acceptation-rejet : générer une valeur $u^{(t)} \sim \mathscr{U}_{[0;1]}$
   $$x^{(t)} := \begin{cases} y^{(t)} \text{ si } u^{(t)} \leq \alpha^{(t)} \\ x^{(t-1)} \text{ sinon} \end{cases}$$

*Ex* : $T(t) = T_0\left(\frac{T_f}{T_0}\right)^{\frac{t}{n}}$ $\Rightarrow$ particulièrement utile en présence d'optimums locaux

# Échantillonneur de Gibbs

Dimension $\nearrow$ $\Rightarrow$ très difficile de proposer des valeurs probables

**Échantillonneurs de Gibbs** : réactualisation coordonnée par coordonnée, en conditionnant sur les dernières valeurs obtenues (pas d'acceptation-rejet)

1. Initialiser $x^{(0)} = (x_1^{(0)}, \dots, x_d^{(0)})$
2. Pour $t = 1, \dots, n+N$ :
   a. Simuler $x_1^{(t)} \sim p(x_1 | x_2^{(t-1)}, \dots, x_d^{(t-1)})$
   b. Simuler $x_2^{(t)} \sim p(x_2 | x_1^{(t)}, x_3^{(t-1)}, \dots, x_d^{(t-1)})$
   c. ...
   d. Simuler $x_i^{(t)} \sim p(x_i | x_1^{(t)}, \dots, x_{i-1}^{(t)}, x_{i+1}^{(t-1)}, \dots, x_d^{(t-1)})$
   e. ...
   f. Simuler $x_d^{(t)} \sim p(x_d | x_2^{(t)}, \dots, x_{d-1}^{(t)})$

**NB** : si la loi conditionnelle est inconnue pour certaines coordonnées, on peut introduire une étape d'acceptation-rejet pour cette coordonnée uniquement (*Metropolis within gibbs*)

# Méthodes numériques alternatives

- **Bayésien variationnel**

- **Calcul Bayésien Approché** $(ABC)$