

ECN 6338 Cours 10

La simulation Monte Carlo séquentielle

William McCausland

2022-03-24

Une densité à quatre modes

La fonction de Himmelblau :

$$\phi(x) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2$$

- ▶ Difficile à minimiser, un cas type pour les tests de performance.
- ▶ Non-négative, quatre racines : $(-3.779, -3.283)$,
 $(-2.805, 3.131)$, $(3, 2)$, $(3.584, -1.848)$

Considérez une densité cible non-normalisée de la forme

$$h(x) = e^{-k\phi(x)},$$

où $k \in (0, 1]$ est un paramètre qui indique à quel point la fonction est hérissée.

La densité $h(x)$ en R

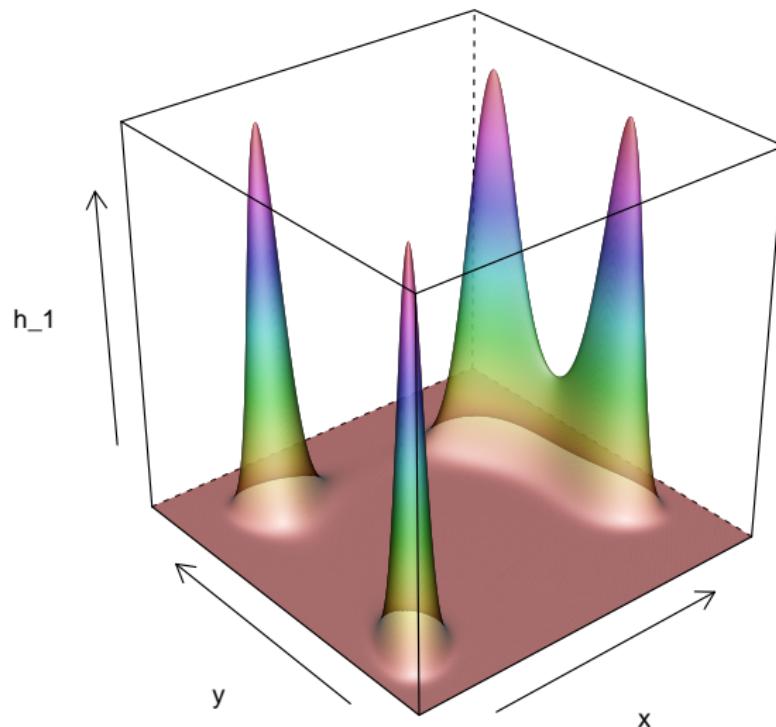
```

# La fonction de densité non-normalisée  $h(x)$ 
h <- function(x, k) {
  ph <- (x[1]^2 + x[2] - 11)^2 + (x[1] + x[2]^2 - 7)^2
  exp(-k * ph)
}

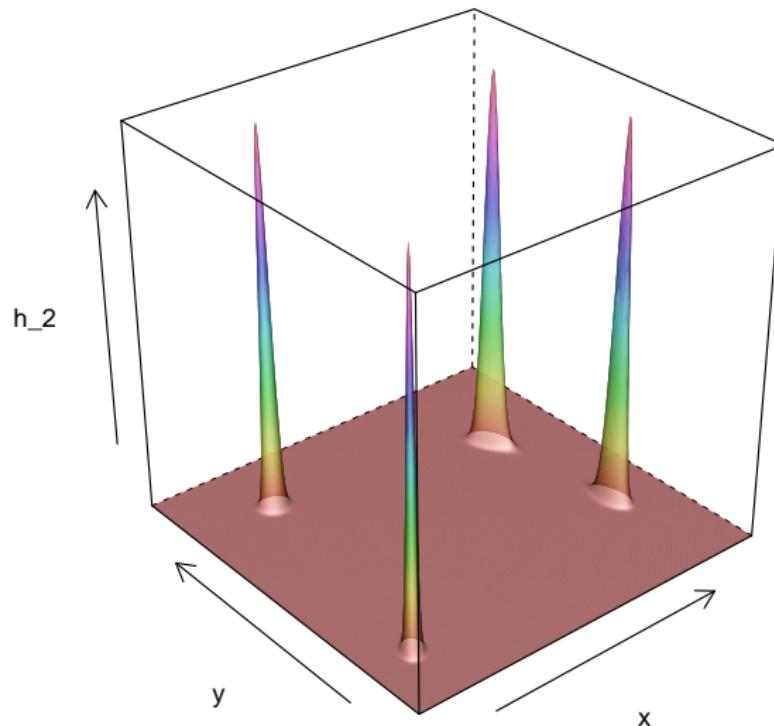
# Évaluations sur une grille pour  $k=0.1, 1.0$ 
library(lattice)
k_1 <- 0.1; k_2 <- 1
x_1 <- seq(-5, 5, by=0.02)
x_2 <- seq(-5, 5, by=0.02)
x_mat <- as.matrix(expand.grid(x_1, x_2))
colnames(x_mat) <- c('x', 'y')
df <- data.frame(h_1 = apply(x_mat, 1, h, k_1),
                  h_2 = apply(x_mat, 1, h, k_2), x_mat)

```

Graphique pour $k = 0.1$



Graphique pour $k = 1.0$



Échantillonage préférentiel

- ▶ Le problème : intégrer des fonctions $g(x)$ par rapport à la densité $f(x)$, lorsqu'on peut seulement évaluer un noyau $h(x) = f(x)/c$, où c est une constante inconnue.
- ▶ Une densité alternative (ou biaisée) est une $p(x)$ telle que
 - ▶ $p(x) > 0 \Leftrightarrow h(x) > 0$,
 - ▶ $\sup_x h(x)/p(x) < \infty$.
- ▶ Pour être utile, il faut pouvoir évaluer la densité alternative et simuler des variables aléatoires de la loi alternative.
- ▶ Identités importantes de la population :

$$E_p \left[\frac{h(x)}{p(x)} \right] = \int \frac{h(x)}{p(x)} p(x) dx = \int h(x) dx = \frac{1}{c}.$$

$$E_p \left[\frac{g(x)h(x)}{p(x)} \right] = \int \frac{g(x)h(x)}{p(x)} p(x) dx = \int \frac{f(x)g(x)}{c} dx = \frac{E_f[g(x)]}{c}$$

Échantillonage préférentiel, cont.

- ▶ Maintenant, supposons qu'on peut tirer $x^{(m)}$, $m = 1, \dots, M$ de la loi alternative (ayant la densité $p(x)$).
- ▶ Si $E_f[g(x)]$ existe,

$$\frac{1}{M} \sum_{m=1}^M \frac{h(x^{(m)})}{p(x^{(m)})} = \frac{1}{M} \sum_{m=1}^M w^{(m)} \xrightarrow{p} \frac{1}{c}$$

$$\frac{1}{M} \sum_{m=1}^M \frac{h(x^{(m)})}{p(x^{(m)})} g(x^{(m)}) = \frac{1}{M} \sum_{m=1}^M w^{(m)} g(x^{(m)}) \xrightarrow{p} \frac{1}{c} E_f[g(x)],$$

où $w^{(m)} \equiv h(x^{(m)})/p(x^{(m)})$.

- ▶ Et alors (par le théorème Mann-Wald)

$$\frac{\sum_{m=1}^M w^{(m)} g(x^{(m)})}{\sum_{m=1}^M w^{(m)}} \xrightarrow{p} E_f[g(x)].$$

Autres notes sur l'échantillonage préférentiel

- ▶ Les $w^{(m)}$ sont pareilles pour chaque choix de $g(\cdot)$.
- ▶ Dans un sens, l'échantillon pondéré $(w^{(m)}, x^{(m)})$ représente la loi cible.
- ▶ On peut faire les simulations de $p(x)$ par
 - ▶ Tirages iid
 - ▶ Méthodes de réduction de variance
 - ▶ Nombres quasi-aléatoires aléatorisés
 - ▶ Méthodes MCMC
- ▶ L'interprétation de l'intégration par l'échantillonage préférentiel comme un changement de variables est très intéressante.

L'erreur numérique de l'échantillonage préférentiel

- ▶ Selon la méthode Monte Carlo utilisée pour tirer de $p(x)$, on obtient la variance numérique Σ du vecteur (n, d) , où n est le numérateur et d est le dénominateur :

$$\Sigma = \begin{bmatrix} \sigma_n^2 & \sigma_{nd} \\ \sigma_{nd} & \sigma_d^2 \end{bmatrix}.$$

- ▶ Selon la méthode delta multivariée, la variance numérique du ratio (l'estimateur numérique de $E_f[g(x)]$) est ($r = n/d$)

$$\sigma_r^2 \approx \frac{1}{d^2}(\sigma_n^2 - 2r\sigma_{nd} + r^2\sigma_d^2).$$

- ▶ Quand les tirages de $p(x)$ sont iid, la taille effective de l'échantillon, (Effective Sample Size, ESS) la taille d'un échantillon iid qui donnent la même erreur numérique, est

$$\frac{(\sum_{m=1}^M w_i)^2}{\sum_{m=1}^M w_i^2} \xrightarrow{p} \frac{\sigma^2/M}{\sigma_r^2}, \quad \text{où } \sigma^2 = \text{Var}_f[g(x)].$$

Poids préférentiels : code

```
# Nombre de tirages de la loi alternative
M <- 1000
# La loi alternative est N(0_2, sigma^2 I_2)
mu <- 0; sigma <- 3
# Tirage de p(x)
rn <- rnorm(2*M, mean=mu, sd=sigma)
# Évaluation de log p(x)
dn <- dnorm(rn, mean=mu, sd=sigma, log=TRUE)
x_rnd <- matrix(rn, nrow=M, ncol=2)
log_p <- rowSums(matrix(dn, nrow=M, ncol=2))
# Évaluation des densités cibles h1 et h2
log_h1 <- log(apply(x_rnd, 1, h, k_1))    # k = 0.1
log_h2 <- log(apply(x_rnd, 1, h, k_2))    # k = 1.0
# Poids pour les deux densités cibles
w1 = exp(log_h1 - log_p - max(log_h1 - log_p))
w2 = exp(log_h2 - log_p - max(log_h2 - log_p))
```

ESS (Effective Sample Size)

```
f_ESS <- function(w) {  
  sum(w)^2/sum(w^2)  
}
```

```
f_ESS(w1)
```

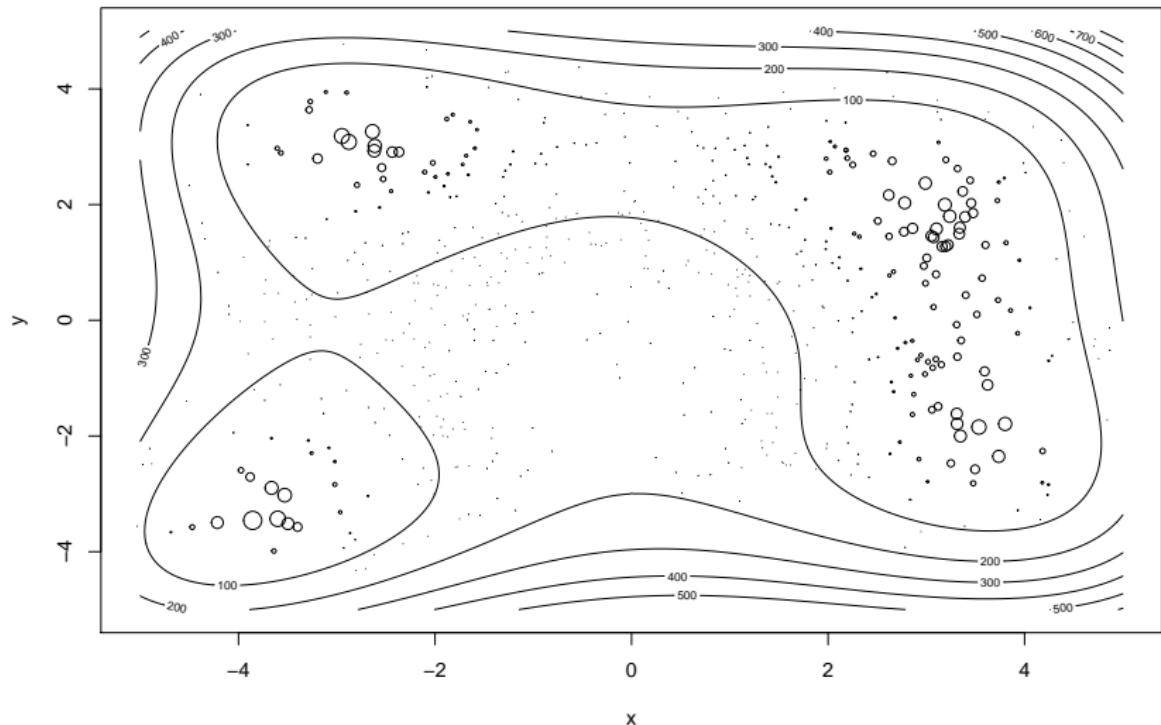
```
## [1] 67.03337
```

```
f_ESS(w2)
```

```
## [1] 8.21487
```

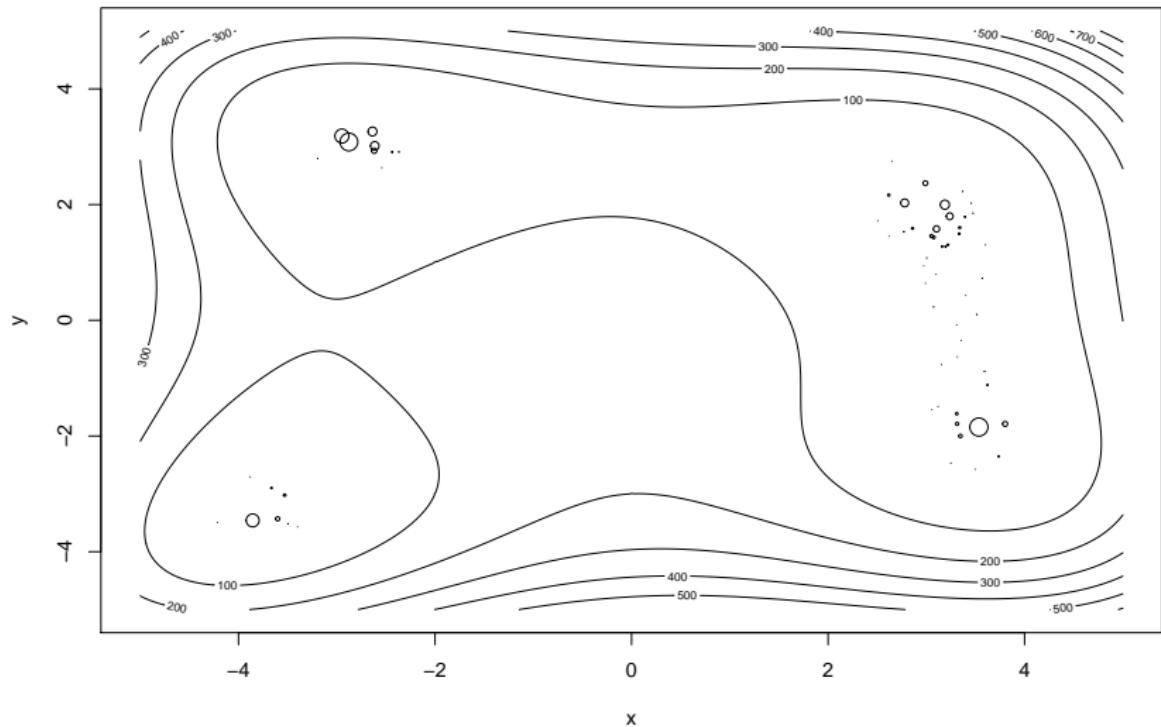
Poids préférentiels : $k = 0.1$

```
contour(x_1, x_2, Himmelblau, xlab='x', ylab='y')  
points(x_rnd, cex=2*sqrt(w1))
```



Poids préférentiels : $k = 1.0$

```
contour(x_1, x_2, Himmelblau, xlab='x', ylab='y')  
points(x_rnd, cex=2*sqrt(w2))
```



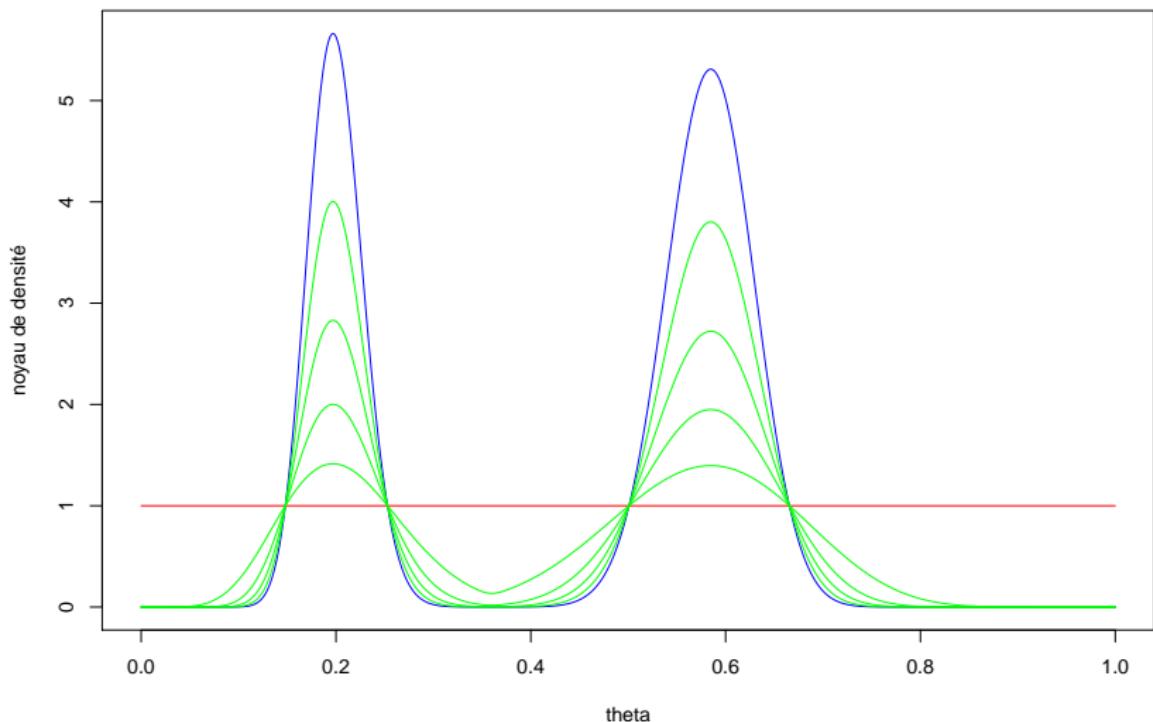
Rééchantillonage (Resampling)

- ▶ À noter : un changement de notation pour être cohérent avec Geweke et Durham (2019), $\theta^{(m)}$ devient θ_n .
- ▶ Problème : on a un échantillon (w_n, θ_n) , $n = 1, \dots, N$, pour le noyau de densité $k(\theta)$, et on veut un échantillon θ_n^* , $n = 1, \dots, N$ avec poids égaux pour $k(\theta)$.
- ▶ Solution : échantillonnez θ_n^* , $n = 1, \dots, N$, avec remise, de l'échantillon θ_n , $n = 1, \dots, N$, avec les poids de probabilité w_n .
- ▶ Le nombre de fois θ_n se trouve dans l'échantillon θ_n^* , $n = 1, \dots, N$, est binomial : $\text{Bi}(N, w_n / \sum_{n=1}^N w_n)$.
- ▶ θ_n^* , $n = 1, \dots, N$, est un échantillon pour $k(\theta)$.
- ▶ Attention : le rééchantillonage ajoute du bruit. On le fait seulement si on a besoin d'un échantillon non-pondéré.

L'exercice de simulation Monte Carlo séquentielle

- ▶ Simulation d'une suite de lois, avec noyaux de densité $k^{(0)}(\theta)$, $k^{(1)}(\theta)$, ... $k^{(L)}(\theta)$, où $k^{(0)}(\theta) = f^{(0)}$ est normalisée.
- ▶ Il est facile de simuler de la loi avec densité $f(\theta)$.
- ▶ Le dernier noyau de densité $k^{(L)}(\theta)$ est de la loi cible.
- ▶ Les loi intermédiaires peut être ou ne pas être de l'intérêt, mais elles sont utiles pour diviser le problème en parties gérables.
- ▶ Une exemple courant :
 - ▶ $k^{(0)}(\theta) = 1_{[0,1]}(\theta)$
 - ▶ $k^{(L)}(\theta) = \frac{3}{5}f_{\text{Be}}(70, 50) + \frac{2}{5}f_{\text{Be}}(40, 160)$
 - ▶ $k^{(l)}(\theta) = [k^{(L)}(\theta)]^{1/L}$, $l = 1, \dots, L - 1$.

Illustration de l'exemple courant pour $L = 5$



Quatre cas d'importance pratique

1. Problème de simulation a posteriori - revenu/refroidissement à l'exposant (power tempering)

$$k^{(I)}(\theta) = p_0(\theta)[p(y|\theta)]^{(I/L)}, \quad I = 0, 1, \dots, L.$$

2. Problème de simulation a posteriori - revenu/refroidissement aux données (data tempering)

$$k^{(I)}(\theta) = p_0(\theta) \prod_{\tau=1}^{t_I} p(y_\tau | y_{1:\tau-1}, \theta), \quad 0 = t_0 < t_1 < \dots < t_L = T$$

3. Problème de maximum de vraisemblance - recuit simulé

$$k^{(I)}(\theta) = p_0(\theta)[p(y|\theta)]^{\gamma I}, \quad 0 = \gamma_0 < \gamma_1 < \dots$$

4. Problème de filtrage, où x_t est une suite d'états latente

$$p(x_t | y_{1:t}), \quad t = 1, \dots, T.$$

Les lois intermédiaires peuvent être de l'intérêt dans les cas 2 et 4.

L'algorithme SMC (version de Geweke et Durham (2019))

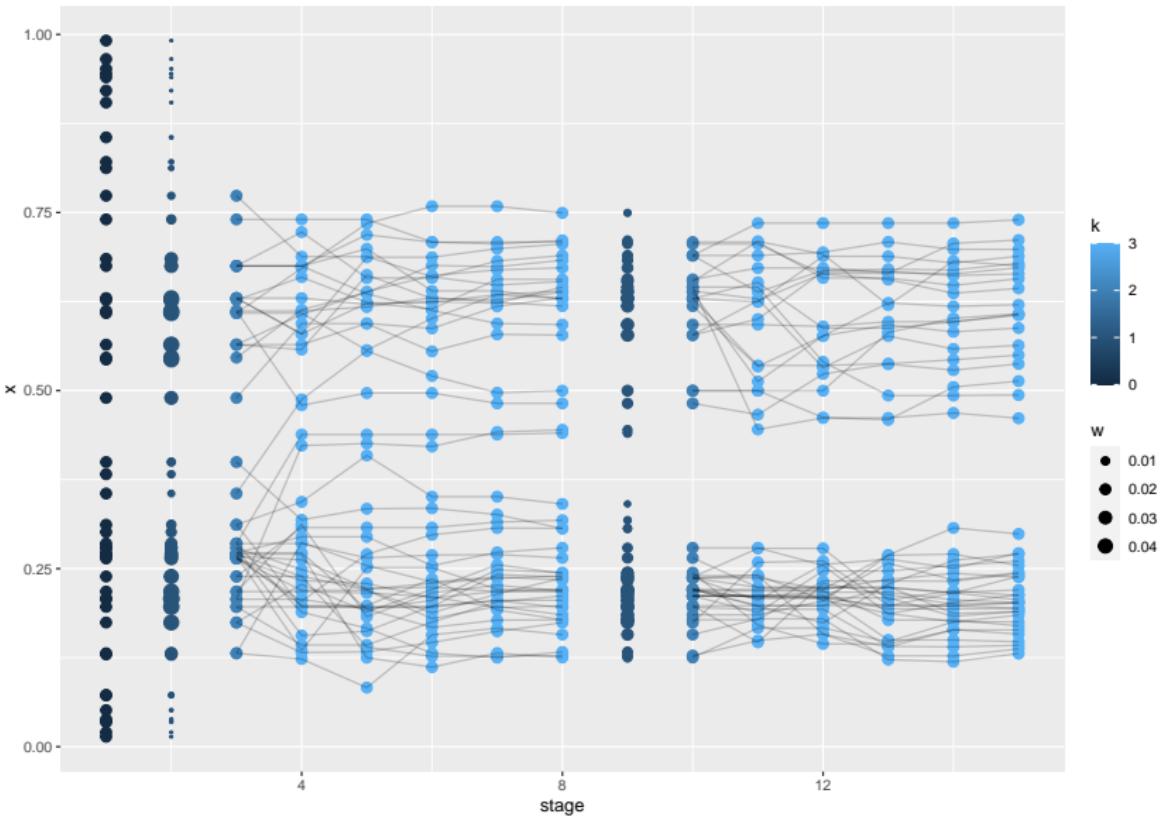
Initialisation

Tirer $\theta_n^{(0)} \sim f^{(0)}(\theta)$, $n = 1, \dots, N$ pour avoir un échantillon $(N^{-1}, \theta_n^{(0)})$ qui représente $f^{(0)}(\theta)$.

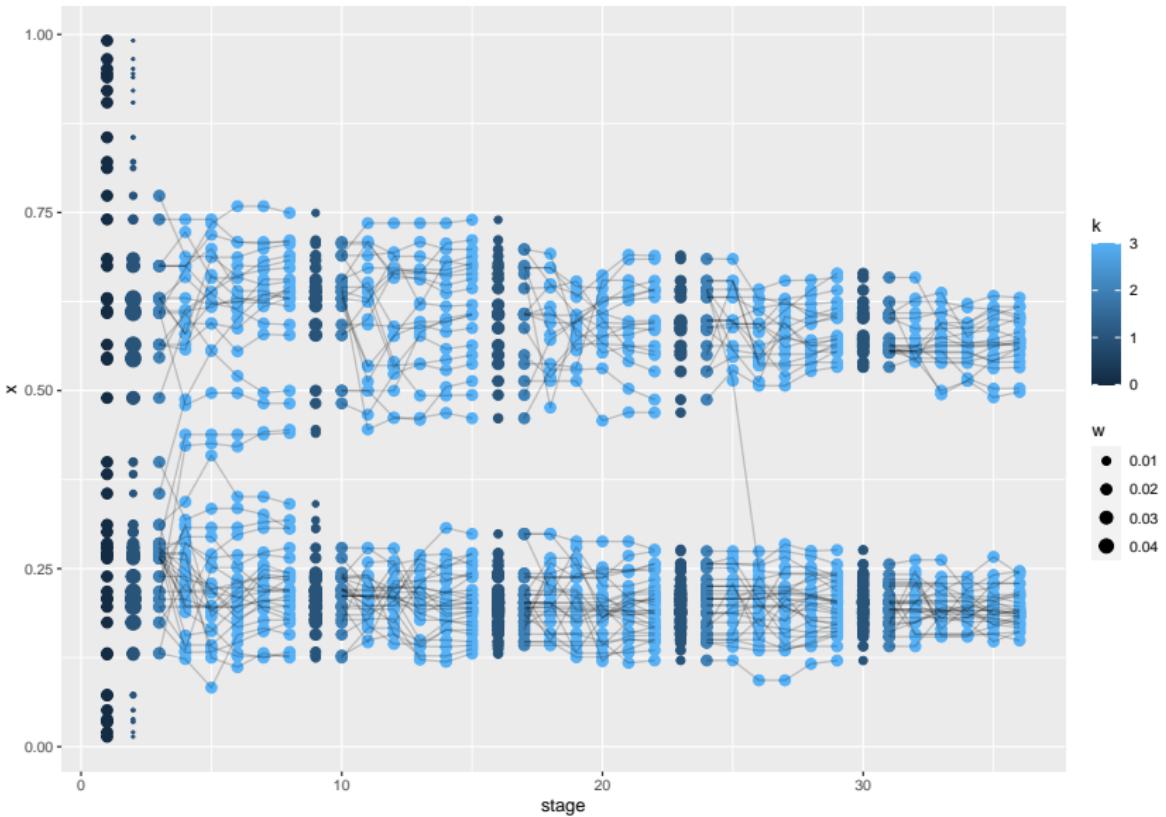
Itération à faire pour $I = 1, \dots, L$

1. Phase C (Correction, avec échantillonage préférentiel)
 - ▶ L'intrant est un échantillon $(N^{-1}, \theta_n^{(I-1)})$ représente $f^{(I-1)}(\theta)$.
 - ▶ On calcule $w_n^{(I)} = k^{(I)}(\theta_n^{(I-1)})/k^{(I-1)}(\theta_n)$.
 - ▶ Le résultat est un échantillon $(w_n^{(I)}, \theta_n^{(I-1)})$ qui représente $k^{(I)}(\theta)$.
2. Phase S (Sélection, avec rééchantillonage)
 - ▶ On rééchantillonne l'échantillon pondéré $(w_n^{(I)}, \theta_n^{(I-1)})$, qui représente $k^{(I)}(\theta)$.
 - ▶ Le résultat est un échantillon $(N^{-1}, \theta_n^{I,0})$ qui représente $k^{(I)}(\theta)$.
3. Phase M (Mutation, avec transitions MCMC)
 - ▶ Pour chaque $\theta_n^{(I,0)}$, effectuer K_I itérations d'une transition markovienne qui préserve la loi avec noyau de densité $k^{(I)}(\theta)$: $\theta_n^{(I,\kappa)}$, $\kappa = 1, \dots, K_I$. Garder $\theta_n^{(I)} = \theta_n^{(I,K_I)}$.
 - ▶ Le résultat est un échantillon $(N^{-1}, \theta_n^{(I)})$ qui représente $k^{(I)}(\theta)$.

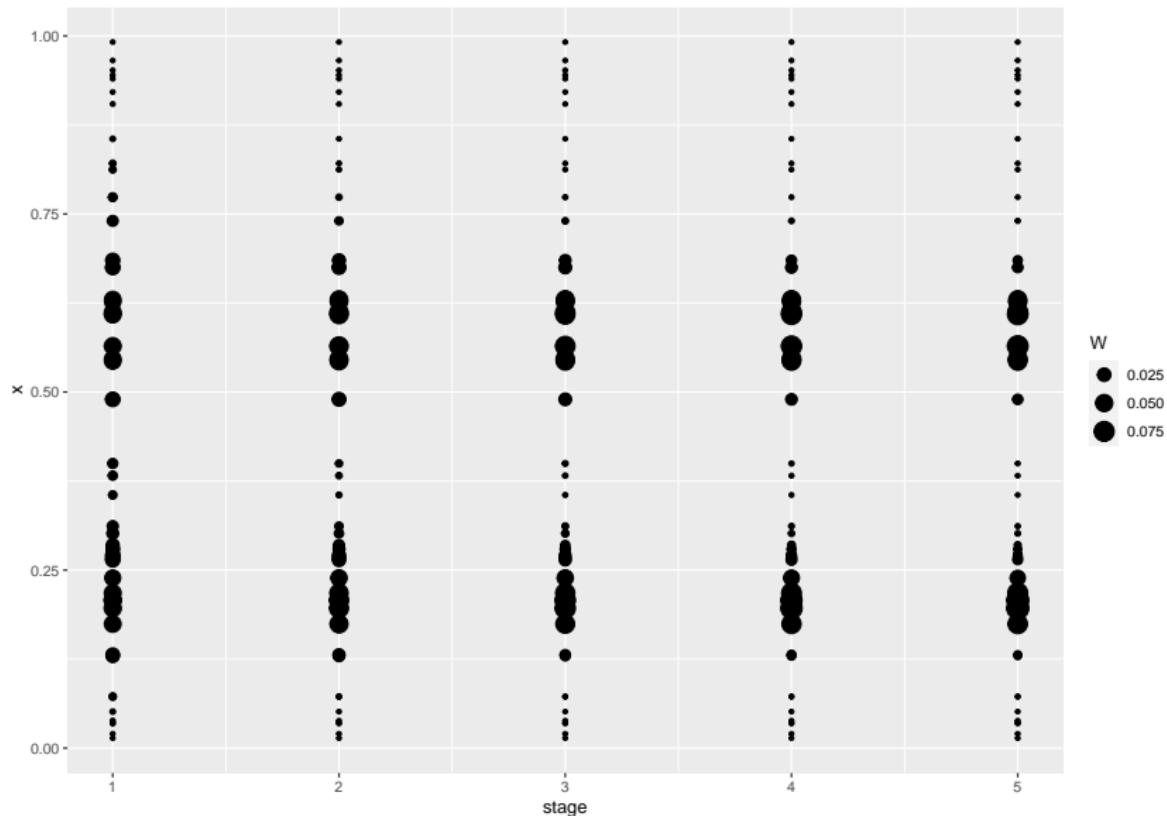
Deux premiers cycles: $(C, S, M \times 5) \times 2$



Tous les cinq cycles



Déplétion des particules (ESS = 26.8, 21.0, 17.6, 15.5, 14.1) si on fait seulement la phase C



Quelques détails

Deux questions à résoudre pour mettre en oeuvre l'algorithme :

1. Comment sélectionner la séquence $k^{(l)}(\theta)$, $l = 1, \dots, L - 1$, entre $f^{(0)}(\theta)$ et la cible $k^{(L)}(\theta)$?
2. À la phase M de l'itération l , comment décider le nombre d'itérations K_l dans la phase M et quelle transitions markoviennes à utiliser?

L'idée d'adaptation (version naive)

1. Rappelons les deux cas : revenu à l'exposant et revenu aux données. On choisit $k^{(l)}(\theta)$ dans la phase C en allant vers la cible $k^{(L)}(\theta)$ jusqu'au point que la ESS atteint un seuil toujours acceptable.
2. On choisit les paramètres des transitions (par exemple, la variance de la proposition Metropolis) et leur nombre pour atteindre un seuil d'efficacité numérique jugé acceptable.

Quelques détails (cont.)

Pourquoi cette version de l'idée est-elle naïve?

- ▶ Les théorèmes qui justifient l'algorithme (loi de grands nombres, théorème central limite) reposent sur la nature markovienne de la simulation.
- ▶ Les décisions à faire dépendent de toute l'histoire de simulation alors la version adaptative n'est pas markovienne.

La solution :

- ▶ Executer deux versions de l'algorithme, en ordre :
 1. Une version adaptative, où on enregistre toutes les décisions, comme les $k^{(l)}(\theta)$, les K_l , les variances des propositions Metropolis.
 2. Une version non-adaptive où on utilise toutes les décisions utilisées dans la version adaptative.
- ▶ Utiliser seulement les résultats de la deuxième version.

Calcul des erreurs numéristiques

- ▶ Geweke et Durham (2019) modifie l'algorithme pour organiser les particules en J groupes de N particules.
- ▶ À la fin de l'itération I , les particules sont $\theta_{jn}^{(I)}$, $j = 1, \dots, J$ et $n = 1, \dots, N$, pour un total de JN particules.
- ▶ À la phase S, le rééchantillonage se fait groupe par groupe : pour chaque groupe j , on sélectionne $\theta_{jn}^{(I,0)}$, $n = 1, \dots, N$, seulement parmi les particules pondérées $(w_{jn}, \theta_{jn}^{(I)})$ du groupe j .
- ▶ En conséquence, l'efficacité numérique diminue mais les J échantillons de taille N sont indépendants.
- ▶ Pour $j = 1, \dots, J$, soit $\bar{g}_j \equiv N^{-1} \sum_{n=1}^N g(\theta_{jn}^{(L)})$ l'estimation de $E_{f^L}[g(\theta)]$ à partir des particules $\theta_{jn}^{(L)}$, $n = 1, \dots, N$, du groupe j .
- ▶ Alors l'estimation globale $\bar{g} = J^{-1} \sum_{j=1}^J \bar{g}_j$ a une variance numéristique asymptotique (en J et N) estimée par

$$\hat{\sigma}_g^2 = J^{-1} \sum_{j=1}^J (\bar{g}_j - \bar{g})^2.$$