

ABC for alpha-stable models

Clémence Chevrier, Félix de Champs, Enzo Lounes, Zakaria Bouliaire

Simulation and Monte Carlo Methods
ENSAE Paris

30 April, 2024



Table des matières

- ① Introduction
- ② RQMC et Monte Carlo Standard
- ③ Méthodes

1 Introduction

2 RQMC et Monte Carlo Standard

3 Méthodes

Distributions alpha-stables et génération

Les distributions alpha-stables permettent de modéliser des phénomènes avec des queues de distributions lourdes, avec variance infinie et asymétrie.

Paramètres de la distribution alpha-stable univariée :

$\alpha \in]0, 2]$: décroissance de la queue

$\beta \in]-1, 1]$: degré et signe de l'asymétrie

$\gamma > 0$: facteur multiplicatif

$\delta \in \mathbb{R}$: l'origine

Aperçu d'une distribution alpha-stable

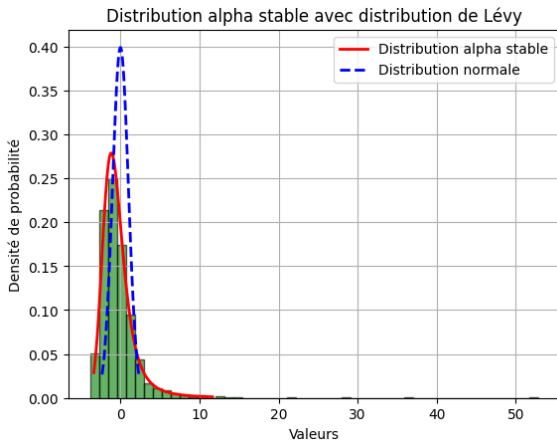


Figure 1: Aperçu d'une distribution alpha-stable. ($\alpha=1.5$, $\beta=1$, $\gamma=1$, $\delta=0$)

Génération

On simule $W \sim \text{Exp}(1)$

On simule $U \sim \mathcal{U}[-\pi/2, \pi/2]$

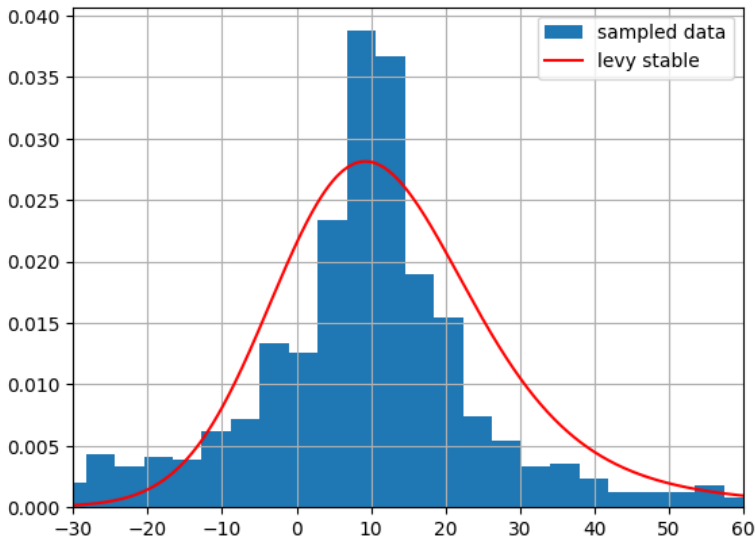
On calcule ceci :

$$\bar{y} = \begin{cases} S_{\alpha,\beta} \frac{\sin \alpha (u + B_{\alpha,\beta})}{(\cos u)^{1/\alpha}} \left[\frac{\cos (u - \alpha (u + B_{\alpha,\beta}))}{w} \right]^{\frac{1-\alpha}{\alpha}} & \text{if } \alpha \neq 1 \\ \frac{2}{\pi} \left[\left(\frac{\pi}{2} + \beta u \right) \tan u - \beta \ln \frac{\frac{\pi}{2} w_i \cos u}{\frac{\pi}{2} + \beta u} \right] & \text{if } \alpha = 1, \end{cases}$$

$$\text{with } S_{\alpha,\beta} = (1 + \beta^2 \tan^2 (\frac{\pi\alpha}{2}))^{1/(2\alpha)} \text{ and } B_{\alpha,\beta} = \frac{1}{\alpha} \arctan (\beta \tan (\frac{\pi\alpha}{2})).$$

Finalement, on applique la transformation $y = \gamma \bar{y} + \delta$

Génération : résultats



1 Introduction

2 RQMC et Monte Carlo Standard

3 Méthodes

Estimation avec les méthodes de Monte Carlo et RQMC

Monte Carlo

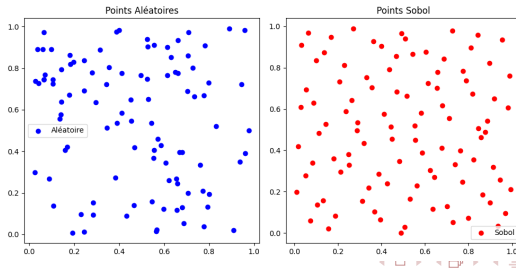
```
def monte_carlo_estimate_manual(f, size, params, u, w):  
    y = univ_alpha_stable_sampler(params, u, w, size)  
    return np.mean(f(y))
```

Random Quasi Monte Carlo (RQMC)

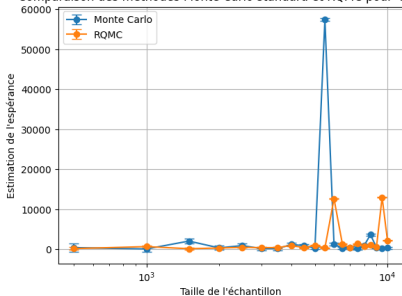
La génération de nombres aléatoires est remplacée par une séquence déterministe de points comme la séquence quasi-aléatoire de Sobol.

```
def rqmc_estimate_manual(f, size, params, u, w):  
    d = 1 # Dimension de la séquence  
    points = sobol_sequence(n, d)  
    u = np.arccos(2 * points - 1) - np.pi/2 # Transformation pour obtenir u entre -pi/2 et pi/2  
    w = -np.log(1 - points) # Transformation pour obtenir w selon une distribution exponentielle  
    y = univ_alpha_stable_sampler(params, u, w, size)  
    return np.mean(f(y))
```

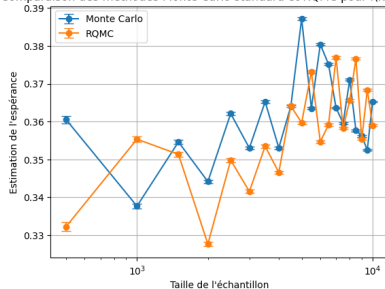
Séquence de Sobol



Comparaison RQMC et Monte Carlo Standard

Comparaison des méthodes Monte Carlo standard et RQMC pour $f(x) = x$ 

Comparaison pour $f(x) = x$

Comparaison des méthodes Monte Carlo standard et RQMC pour $f(x) = \sin(x)$ 

Comparaison pour $f(x) = \sin(x)$

Comparaison RQMC et Monte Carlo Standard

Les deux méthodes convergent vers une valeur similaire, ce qui est attendu avec la loi des grands nombres

La convergence avec RQMC semble être plus rapide

Comparaison RQMC et Monte Carlo Standard

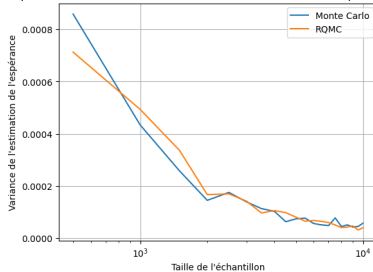
La MSE est plus faible pour RQMC

MSE Monte Carlo pour $f(x) = \exp(-x^2)$: 0.03075766222756377

MSE RQMC pour $f(x) = \exp(-x^2)$: 0.02952922162463651

La variance de l'estimation de l'espérance par la méthode RQMC est généralement plus faible que celle du Monte Carlo standard

Comparaison des variances entre Monte Carlo standard et RQMC pour $f(x) = \sin(x)$



1 Introduction

2 RQMC et Monte Carlo Standard

3 Méthodes

ABC-rejection

MCMC-ABC

SMC sampler PRC-ABC algorithm (Peters et al., 2009)

1 Introduction

2 RQMC et Monte Carlo Standard

3 Méthodes

ABC-rejection

MCMC-ABC

SMC sampler PRC-ABC algorithm (Peters et al., 2009)

ABC-rejection : principe

Un algorithme d'ABC-rejection est une méthode d'inférence bayésienne approximative utilisée lorsque l'on peut simuler, pour des données, la distribution $p(y|\theta)$ mais qu'on ne peut calculer la vraisemblance.

Ici, chercher à calculer la vraisemblance revient à déterminer une estimation des paramètres de la loi stable.

ABC-rejection : prérequis

Utilisation de `prior_sample` : Cette fonction génère un échantillon à partir de la distribution a priori des paramètres du modèle alpha-stable.

Définition des paramètres : Les paramètres α, β, γ et δ sont définis à partir des conseils de l'article conseillé dans le cadre du projet.

Génération des données observées : les données observées y sont générées à partir de la distribution alpha-stable en utilisant la fonction `levy_stable.rvs`.

Définition de `distance_S` : Cette fonction calcule la distance entre les statistiques résumées des données observées et des données simulées via `y_barre` définie plus tôt.

ABC-rejection : l'algorithme

Nombre de samples : Une boucle for est utilisée pour effectuer l'inférence bayésienne approximative sur $N = 100\,000$ itérations.

Générations à chaque sample : Un échantillon de paramètres θ est généré à partir de `prior_sample`. Puis on génère un échantillon x à partir de la fonction transformation et des θ générés.

Acceptation de θ : Si la distance calculée via `distance_S` est inférieure à un certain seuil, on accepte le paramètre θ , qu'on ajoute ensuite ou non à la liste des paramètres acceptés.

ABC-rejection : présentation des résultats

Nous pouvons présenter les paramètres retenus sous la forme d'un boxplot

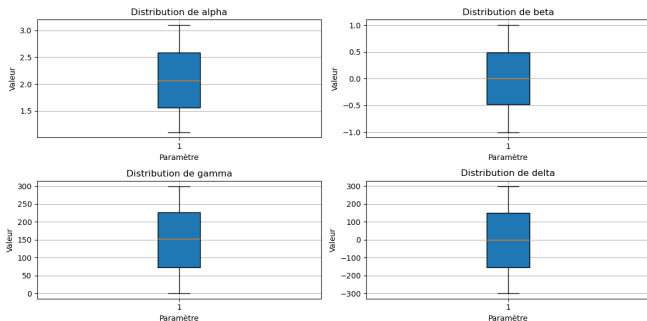


Figure 2: Boxplot des paramètres alpha; beta; gamma; delta

1 Introduction

2 RQMC et Monte Carlo Standard

3 Méthodes

ABC-rejection

MCMC-ABC

SMC sampler PRC-ABC algorithm (Peters et al., 2009)

MCMC-ABC (Marjoram, 2003)

Pseudo-code

On simule $\theta^* \sim q(\cdot | \theta_n)$.

On simule $x^* = ((x^*)_1, \dots, (x^*)_T)$ selon $p(\cdot | \theta^*)$.

Si $x^* = y$ on accepte $(\theta_{n+1}, x_{n+1}) = (\theta^*, x^*)$ avec la probabilité
$$h = \min\left(1, \frac{\pi(\theta^*)q(\theta_n | \theta^*)}{\pi(\theta_n)q(\theta^* | \theta_n)}\right)$$

Sinon $(\theta_{n+1}, x_{n+1}) = (\theta_n, x_n)$

MCMC-ABC (Cao, Zhang, Zhou, 2024)

Pseudo-code

On simule $\theta^* \sim q(\cdot | \theta_n)$.

On simule $x^* = ((x^*)_1, \dots, (x^*)_T)$ selon $p(\cdot | \theta^*)$.

On accepte si $(\theta_{n+1}, x_{n+1}) = (\theta^*, x^*)$ avec la probabilité

$$h = \min\left(1, \frac{\pi(\theta^*)q(\theta_n|\theta^*)K_\epsilon(\Delta(x^*, y))}{\pi(\theta_n)q(\theta^*|\theta_n)K_\epsilon(\Delta(x_n, y))}\right)$$

Sinon $(\theta_{n+1}, x_{n+1}) = (\theta_n, x_n)$

MCMC-ABC

Remarques

En pratique, il est difficile d'obtenir $x^* = y$ donc pour l'algorithme de Majoram, on pose $h = \min(1, \frac{\pi(\theta^*)q(\theta_n|\theta^*)}{\pi(\theta_n)q(\theta^*|\theta_n)}) \mathbb{1}(|x^* - y| < \epsilon)$ avec $\lim \epsilon = 0$

Si le proposal q est symétrique (par exemple : un noyau gaussien) , $q(\theta_n|\theta^*) = q(\theta^*|\theta_n)$ et $h = \min(1, \frac{\pi(\theta^*)}{\pi(\theta_n)}) \mathbb{1}(|x^* - y| < \epsilon)$

on choisit K_ϵ tel que $S(y) \sim \mathbb{N}(S(x), \epsilon^2 \hat{\Sigma})$ où $\hat{\Sigma}$ est un estimateur de $\text{cov}(S(x)|\theta^*)$

MCMC-ABC

Statistiques

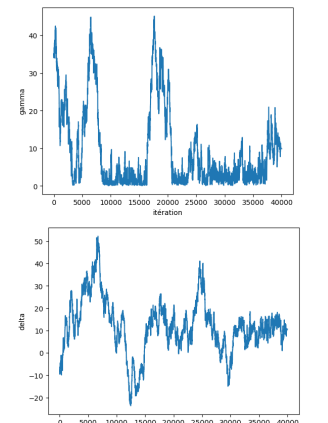
En pratique, utiliser $S(x)$ et $S(y)$ au lieu de x et y où S est une statistique permet de réduire la dimension des calculs sur les data. On choisit la statistique suivante:

$$\hat{v}_\alpha = \frac{\hat{q}_{0.95}(\cdot) - \hat{q}_{0.05}(\cdot)}{\hat{q}_{0.75}(\cdot) - \hat{q}_{0.25}(\cdot)}, \quad \hat{v}_\beta = \frac{\hat{q}_{0.95}(\cdot) + \hat{q}_{0.05}(\cdot) - 2\hat{q}_{0.5}(\cdot)}{\hat{q}_{0.95}(\cdot) - \hat{q}_{0.05}(\cdot)} \quad \text{and} \quad \hat{v}_\gamma = \frac{\hat{q}_{0.75}(\cdot) - \hat{q}_{0.25}(\cdot)}{\gamma},$$

Figure 3: Statistiques 1

MCMC-ABC

Résultats



MCMC-ABC

Résultats

Pour $N=100\ 000$ itérations (50 000 burnin) et $\lim \epsilon = 0$:

Taux d'acceptation : 0.16589

Estimateurs:

$$\hat{\alpha} = 1.8258098218173866$$

$$\hat{\beta} = 0.01577862390082706$$

$$\hat{\gamma} = 9.440643649507724$$

$$\hat{\delta} = 11.846692420476382$$

Root Mean Square Error : 32.96045408144649

MCMC-ABC

Résultats

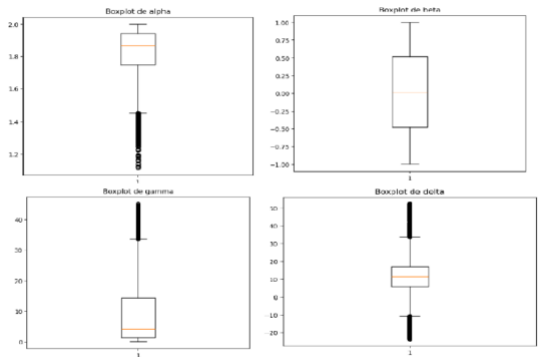


Figure 4: Boxplot des paramètres

1 Introduction

2 RQMC et Monte Carlo Standard

3 Méthodes

ABC-rejection

MCMC-ABC

SMC sampler PRC-ABC algorithm (Peters et al., 2009)

SMC sampler PRC-ABC algorithm (Peters et al., 2009)

Initialization: Set $t = 1$ and specify tolerance schedule $\epsilon_1, \dots, \epsilon_T$.

For $i = 1, \dots, N$, sample $\theta_1^{(i)} \sim \pi(\theta)$, and set weights

$$W_1(\theta_1^{(i)}) = \pi_{LF,1}(\theta_1^{(i)}|y)/\pi(\theta_1^{(i)}).$$

Resample: Resample N particles with respect to $W_t^{(i)}(\theta_t^{(i)})$ and set $W_t^{(i)}(\theta_t^{(i)}) = \frac{1}{N}$, $i = 1, \dots, N$.

SMC sampler PRC-ABC algorithm (Peters et al., 2009) (suite)

- Mutation and correction:** Set $t = t + 1$ and $i = 1$: (a) Sample $\theta_t^{(i)} \sim M_t(\theta_t)$ and set weight for $\theta_t^{(i)}$ to $W_t^{(i)}(\theta_t^{(i)}) = \frac{\pi_{LF,t}(\theta_t^{(i)}|y)}{M_t(\theta_t^{(i)})}$
- (b) With probability $1 - p^{(i)} = 1 - \min(1, W_t^{(i)}(\theta_t^{(i)})/c_t)$, reject $\theta_t^{(i)}$ and go to (a).
- (c) Otherwise, accept $\theta_t^{(i)}$ and set $W_t^{(i)}(\theta_t^{(i)}) = \frac{W_t^{(i)}(\theta_t^{(i)})}{p^{(i)}}$.
- (d) Increment $i = i + 1$. If iN , go to (a).
- (e) If $t < T$ then go to Resample.

Remarques sur l'implémentation

Définition du noyau K (par exemple le calcul de $\hat{\Sigma}$)

Approximation de l'espérance avec une seule valeur ($P=1$).

$$E_{\pi(x|\theta)}[K_{\epsilon}(y - x)] \approx 1/P \sum_{p=1}^P K_{\epsilon}(y - x^p)$$

Valeurs extrêmes.

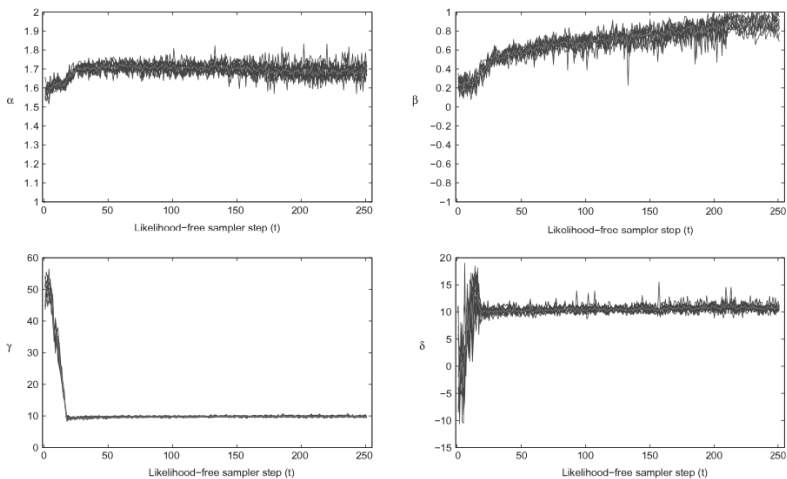


Figure 4: Résultats du SMC sampler

Bibliographie

Likelihood-free Bayesian inference for alpha-stable models,
G.W. Peters, S.A. Sisson, Y. Fan, 2012

Markov chain Monte Carlo without likelihoods, P. Marjoram,
J. Molitor, V. Plagnol, S. Tavaré, (2003)

Sisson, S.A., Fan, Y., 2010. Likelihood-free Markov chain
Monte Carlo. In: Brooks, S.P., Gelman, A., Jones, G., Meng,
X.L. (Eds.), Handbook of Markov Chain Monte Carlo.
Chapman and Hall/CRC.

Peters, G.W., Fan, Y., Sisson, S.A., 2009. On sequential
Monte Carlo, partial rejection control and approximate
Bayesian computation. Tech. Rep. UNSW.

Wikipedia contributors. "Approximate Bayesian computation."
Wikipedia, The Free Encyclopedia. Wikipedia, The Free
Encyclopedia, 26 Apr. 2024. Web. 29 Apr. 2024.

Lien Github

Lien vers notre github :

<https://github.com/ClemenceCVR/Projet-Monte-Carlo>