Compte rendu

Statistiques des valeurs extrêmes

Pierre Monnin

9 mai 2025

Table des matières

1 Introduction

L'objectif du projet était de reprendre les travaux de Vygantas Paulauskas sur l'estimateur par paquets de l'indice de queue, noté α. Cet estimateur présente de nombreux avantages comparé à l'estimateur de Hill, largement utilisé dans ce contexte. Il semble plus simple à calculer, et il est indiqué que pour de petits échantillons et pour des lois stables ou de Pareto à queue lourde, il pourrait même être plus performant.

2 Exercice 1

L'exercice consistait à créer une fonction `estpaq` en langage R. Cette fonction prend en paramètre `x`, l'échantillon d'observations, et Θ, le paramètre à partir duquel on définit N.

Voici le code utilisé, accompagné de ses commentaires explicatifs :

```
# Fonction estpaq(x, theta)
estpaq <- function(x, theta) {
  N <- length(x)                   # Taille de l'échantillon
  n <- floor(N^theta)              # n = partie entière de N puissance theta
  m <- floor(N / n)                # m = partie entière de N divisé par n

  N_truncate <- m * n
  x <- sample(x)
  x <- x[1:N_truncate]

  X_mat <- matrix(x, nrow = m, ncol = n)

  max1 <- apply(X_mat, 2, function(v) max(abs(v)))
  max2 <- apply(X_mat, 2, function(v) max(abs(v)[abs(v) != max(abs(v))]))

  z <- max2 / max1
  S <- sum(z)
  alpha_hat <- S / (n - S)

  return(alpha_hat)
}
```

L'objectif de cette fonction est de calculer l'estimateur par paquets d'un échantillon x en fonction du paramètre Θ.

3 Exercice 2

L'objectif ici est d'étudier la consistance de notre estimateur par paquets.

3.1 Partie C

Voici le tracé de la courbe demandée :

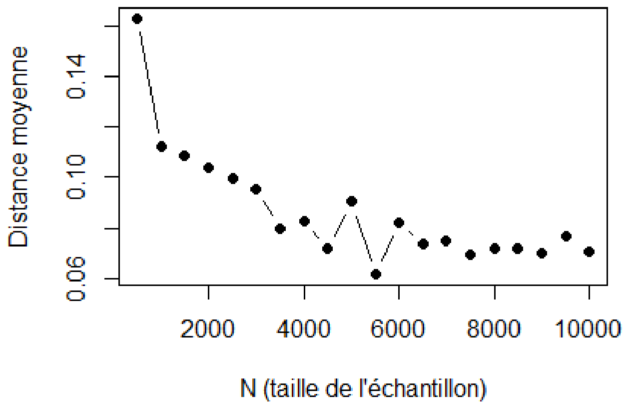

FIGURE 1 – **Courbe de la distance moyenne entre l'estimateur par paquets d'un échantillon de taille** N **et l'indice de queue** $\alpha = 0.7$

On observe que plus l'échantillon est grand, plus la distance moyenne se rapproche de 0.7.

Voici ce que l'on peut en déduire :

— La loi forte des grands nombres nous permet d'écrire, avec $r = 100$ considéré comme suffisamment grand :

$$\bar{\hat{d}}_{N_i,r} \approx \mathbb{E}\left[\hat{d}_{N_i}\right]$$

— De plus, on a :

$$\mathbb{E}\left[\hat{d}_{N_i}\right] = \mathbb{E}\left[|\hat{\alpha}_{N_i} - \alpha|\right]$$

— Enfin, d'après notre graphique :

$$\lim_{N_i \to +\infty} \mathbb{E}\left[|\hat{\alpha}_{N_i} - \alpha|\right] = 0$$

Cela implique donc que notre estimateur $\hat{\alpha}_{N_i}$ est consistant.

3.2 Partie D

Voici les graphiques obtenus en réalisant la même procédure pour $\alpha = 1.5$ et $\alpha = 1.8$:

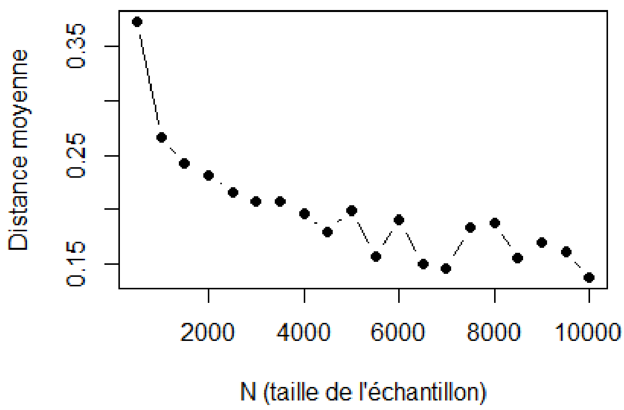

FIGURE 2 – **Courbe de la distance moyenne entre l'estimateur par paquets et l'indice de queue** $\alpha = 1.5$

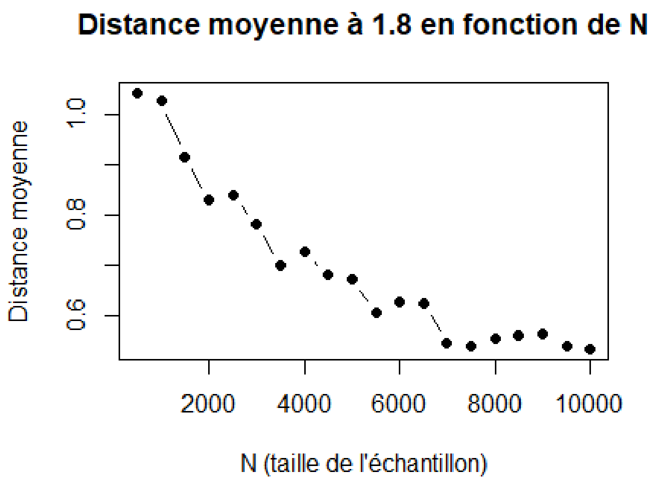

FIGURE 3 – **Courbe de la distance moyenne entre l'estimateur par paquets et l'indice de queue** $\alpha = 1.8$

On observe que les courbes ont la même allure, ce qui s'explique par le fait que l'estimateur est convergent. Ainsi, quelle que soit la valeur de α, la distance entre l'estimateur par paquets et α sera proche de zéro lorsque la taille de l'échantillon N sera suffisamment grande.

Néanmoins, on observe également que la vitesse de convergence décroît lorsque α augmente. Cela s'explique par le fait que, comme indiqué dans l'article de référence, l'estimateur est plus performant pour les lois à queue lourde. Or, plus α est grand, moins la queue est lourde. Voilà pourquoi, à mesure que α

augmente, la convergence de l'estimateur devient plus lente.

3.3 Code

Voici le code R utilisé pour la partie 2 :

```
# Question 2:
# Montrer par simulation la consistance de alpha_hat

library(stabledist)
x <- rstable(10000, alpha = 0.7, beta = 0, gamma = 1, delta = 0)
k <- 1:20
N <- k * 500
alpha_hat <- numeric(length(N))
for (i in seq_along(N)) {
  N_i <- N[i]
  x_Ni <- x[1:N_i]
  alpha_hat[i] <- estpaq(x_Ni, theta = 0.5)
}
distance <- abs(alpha_hat - 0.7)

nb_repetitions <- 100
distance_mat <- matrix(0, nrow = nb_repetitions, ncol = length(N))
for (r in 1:nb_repetitions) {
  x <- rstable(10000, alpha = 0.7, beta = 0, gamma = 1, delta = 0)
  for (i in seq_along(N)) {
    N_i <- N[i]
    x_Ni <- x[1:N_i]
    alpha_hat[i] <- estpaq(x_Ni, theta = 0.5)
  }
  distance <- abs(alpha_hat - 0.7)
  distance_mat[r, ] <- distance
}
moyenne <- colMeans(distance_mat)
plot(N, moyenne, type = "b", main = "Distance moyenne à 0.7",
     xlab = "N", ylab = "Distance moyenne", pch = 19)

# Pour alpha = 1.5 et 1.8, même procédure...
```

4 Exercice 3

L'objectif ici est d'étudier la valeur optimale de θ.

4.1 Graphiques et interprétations

Nous avons réalisé exactement le même procédé que précédemment, mais cette fois-ci pour un seul échantillon de taille $N = 100$, en faisant varier le paramètre θ.

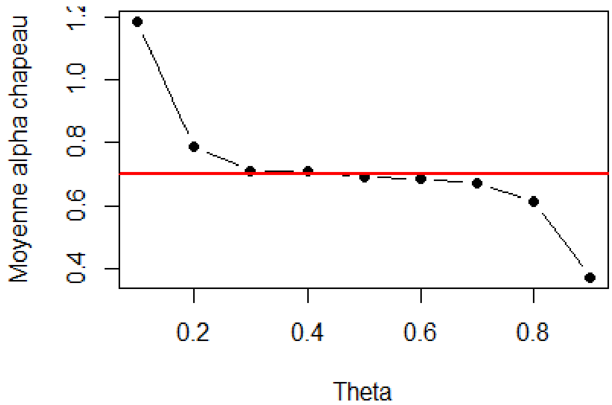

FIGURE 4 – **Courbe de l'estimateur par paquets en fonction de θ, pour un échantillon de taille $N = 100$**

On observe que la première valeur de θ pour laquelle la courbe se rapproche de 0.7 est 0.4, ce qui correspond à la valeur optimale donnée par l'énoncé.

Pour vérifier la stabilité des résultats et s'assurer qu'ils ne dépendent pas uniquement des 100 premières observations, nous avons réitéré l'expérience pour $r = 1000$ et $r = 10\,000$. Voici les graphiques obtenus :

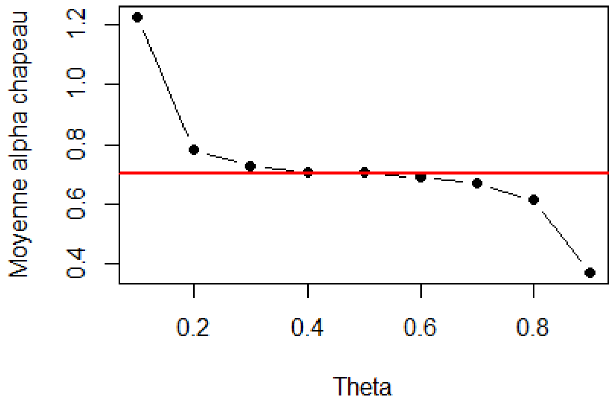

FIGURE 5 – **Courbe de l'estimateur par paquets en fonction de θ, moyenne sur $r = 1000$ répétitions**

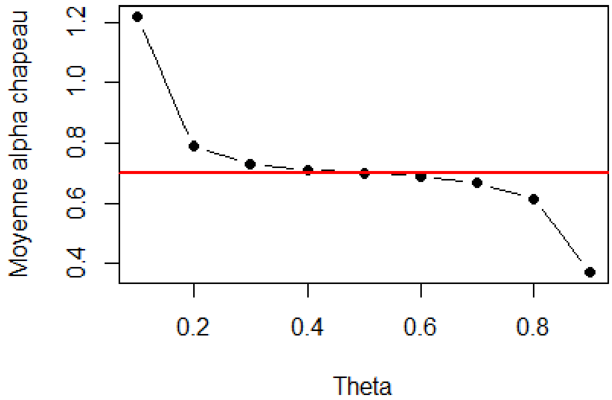

FIGURE 6 – **Courbe de l'estimateur par paquets en fonction de θ, moyenne sur $r = 10\,000$ répétitions**

On constate que plus le nombre de répétitions est élevé, plus la courbe est proche de α, quelle que soit la valeur de θ. Cela s'explique par une convergence plus forte lorsque le nombre de répétitions est grand. La valeur optimale reste $\theta = 0.4$, mais d'autres valeurs de θ donnent également de bons résultats.

4.2 Code

Voici le code R utilisé pour la partie 3, correspondant aux étapes précédentes :

```
# Question 3: Étudier par simulation la valeur optimale de theta

# Génération d'un échantillon stable
x <- rstable(10000, alpha = 0.7, beta = 0, gamma = 1, delta = 0)
Theta <- seq(0.1, 0.9, by = 0.1)
alpha_hat_theta <- numeric(length(Theta))
x_N <- x[1:10000]
for (i in seq_along(Theta)) {
  alpha_hat_theta[i] <- estpaq(x_N, theta = Theta[i])
}

# Répéter 100 fois
nb_repetitions <- 100
alpha_hat_mat <- matrix(0, nrow = nb_repetitions, ncol = length(Theta))
for (r in 1:nb_repetitions) {
  x <- rstable(10000, alpha = 0.7, beta = 0, gamma = 1, delta = 0)
  for (i in seq_along(Theta)) {
    alpha_hat_mat[r, i] <- estpaq(x, theta = Theta[i])
  }
}
moyenne <- colMeans(alpha_hat_mat)
plot(Theta, moyenne, type = "b",
     main = "Moyenne d'alpha chapeau en fonction de Theta (100 répétitions)",
     xlab = "Theta", ylab = "Moyenne alpha chapeau", pch = 19)
abline(h = 0.7, col = "red", lwd = 2)

# Répéter 1000 fois
nb_repetitions2 <- 1000
alpha_hat_mat <- matrix(0, nrow = nb_repetitions2, ncol = length(Theta))
for (r in 1:nb_repetitions2) {
  x <- rstable(10000, alpha = 0.7, beta = 0, gamma = 1, delta = 0)
  for (i in seq_along(Theta)) {
    alpha_hat_mat[r, i] <- estpaq(x, theta = Theta[i])
  }
}
moyenne <- colMeans(alpha_hat_mat)
plot(Theta, moyenne, type = "b",
     main = "Moyenne d'alpha chapeau en fonction de Theta (1000 répétitions)",
     xlab = "Theta", ylab = "Moyenne alpha chapeau", pch = 19)
```

```
abline(h = 0.7, col = "red", lwd = 2)

# Répéter 10 000 fois
nb_repetitions3 <- 10000
alpha_hat_mat <- matrix(0, nrow = nb_repetitions3, ncol = length(Theta))
for (r in 1:nb_repetitions3) {
  x <- rstable(10000, alpha = 0.7, beta = 0, gamma = 1, delta = 0)
  for (i in seq_along(Theta)) {
    alpha_hat_mat[r, i] <- estpaq(x, theta = Theta[i])
  }
}
moyenne <- colMeans(alpha_hat_mat, na.rm = TRUE)
plot(Theta, moyenne, type = "b",
     main = "Moyenne d'alpha chapeau en fonction de Theta (10 000 répétitions)",
     xlab = "Theta", ylab = "Moyenne alpha chapeau", pch = 19)
abline(h = 0.7, col = "red", lwd = 2)
```

Note finale : Les trois parties de code (Exercice 1, Exercice 2, Exercice 3) sont interdépendantes et doivent être mutualisées pour obtenir les résultats attendus.