LSTAT 2040 - TP 6

L'estimateur du maximum de vraisemblance et optimisation numérique

Exercice 1

Soit $X_1, X_2, \dots$ une suite iid dont la distribution commune admet la fonction de répartition F définie par

$$F(x;\alpha,\beta) = \begin{cases} 0 & \text{if } x < 0 \\ (x/\beta)^\alpha & \text{if } 0 \le x \le \beta \\ 1 & \text{if } x > \beta \end{cases},$$

où $(\alpha, \beta) \in (0,\infty)^2$. Trouver le MLE de (α, β) par maximisation de la vraisemblance profilée.

Exercice 2

Soit $X_1, \dots, X_n$ un échantillon iid de distribution commune $\text{Beta}(\theta, 2)$ où $\theta > 0$. On propose d'étudier le MLE de θ de manière numérique et analytique.

Dans ce but, on choisit, par exemple, $\theta_0 = 4$ et l'on génère un échantillon X de taille $n = 100$ de la distribution associée $\text{Beta}(4, 2)$ comme suit :

```
theta_true <- 4
set.seed(2023) # afin que tout le monde obtienne le même échantillon
n <- 100
X <- rbeta(n, theta_true, 2)
```

(a) Faire un graphique de la fonction de log-vraisemblance associée à l'échantillon que vous venez de générer. Cette fonction est-elle strictement concave ? Vérifier votre observation par le calcul.

(b) Afin de maximiser numériquement la log-vraisemblance de l'échantillon généré, on aura besoin d'une valeur de départ pour θ dans notre algorithme de maximisation. On propose d'utiliser l'estimateur des moments $\hat{\theta}_{\text{MM}}$ comme valeur initiale. Donner une formule analytique pour celui-ci et calculer ce dernier sur l'échantillon généré.

(c) Implémenter l'algorithme de Newton–Raphson afin de maximiser la log-vraisemblance de l'échantillon. Utiliser $\hat{\theta}_{\text{MM}}$ comme point de départ par défaut et $\epsilon = 10^{-6}$ comme seuil de tolérance absolue par défaut entre deux itérations successives de l'algorithme. Quelle valeur obtenez-vous ?

(d) Appliquer à nouveau votre algorithme du point (c) avec comme point de départ $\theta = 10$. Que constatez-vous ? Comment pouvez-vous corriger votre algorithme afin d'éviter ce genre de problème ? Appliquer à nouveau votre algorithme dans sa version corrigée.

(e) Dans le cas présent, il est en fait possible d'optimiser la log-vraisemblance analytiquement. Donner une formule analytique pour le MLE $\hat{\theta}_{\text{MLE}}$ et calculer ce dernier sur l'échantillon généré. Est-il comparable aux valeurs obtenues aux points (c) et (d) ? Qu'en conclure sur la nature de l'erreur entre les valeurs calculées et la vraie valeur du paramètre sous-jacent à la génération de l'échantillon X ?

(f) On sait de la théorie que l'estimateur $\hat{\theta}_{\text{MLE}}$ est asymptotiquement normal avec

$$\sqrt{n}(\hat{\theta}_{\text{MLE}} - \theta_0) \xrightarrow{d} \text{N}(0, I^{-1}(\theta_0)),$$

où θ_0 est la vraie valeur du paramètre. Le problème est qu'en pratique, ce dernier est inconnu et donc la variance asymptotique du MLE l'est également. Sur base de l'échantillon généré, proposer deux méthodes différentes pour estimer la variance asymptotique de $\hat{\theta}_{\text{MLE}}$ et comparer les résultats avec la vraie variance asymptotique, qui est ici connue car on sait que $\theta_0 = 4$. Que constatez-vous ?

Exercice 3

Soit $X_1, \ldots, X_n$ un échantillon iid de distribution commune Beta$(\theta, 2)$ où $\theta > 0$.

(a) Trouver l'estimateur du maximum de vraisemblance de $\mu = \mathrm{E}[X_1]$. Quelle est la distribution asymptotique de cet estimateur ?

(b) La moyenne empirique $\overline{X}_n$ est un autre candidat pour l'estimation de μ. Comparer la variance asymptotique de cette dernière avec l'estimateur obtenu en (a).

Exercice 4

Soit $X_1, \ldots, X_n$ un échantillon iid dont la distribution commune a pour densité

$$f(x; \mu, \lambda) = \sqrt{\frac{\lambda}{2\pi x^3}} \exp\left(-\frac{\lambda(x-\mu)^2}{2\mu^2 x}\right) \mathrm{I}(0 < x < \infty),$$

où $(\mu, \lambda) \in (0, \infty)^2$. Cette distribution est appelée inverse-Gaussienne. On propose d'étudier le MLE de (μ, λ) de manière numérique et analytique.

Dans ce but, on choisit, par exemple, $(\mu_0, \lambda_0) = (2, 5)$ et l'on génère un échantillon X de taille $n = 100$ de la distribution inverse Gaussienne associée comme suit :

```
mu_true <- 2
lambda_true <- 5
set.seed(2023)
n <- 100
# install.packages("statmod") # exécuter si le package "statmod" n'est pas installé
library(statmod)
X <- rinvgauss(n, mean = mu_true, shape = lambda_true)
```

(a) Représenter graphiquement la fonction de log-vraisemblance via la fonction `filled.contour()` de `R` afin de se faire une première impression sur la concavité de celle-ci. Un example basique de l'utilisation de `filled.contour()` se présente comme suit :

```
x <- seq(from = 0, to = 1, by = 0.01)
y <- seq(from = 0, to = 1, by = 0.01)
myfun <- function(x,y) x^2+y^2
z <- outer(X = x, Y = y, FUN = myfun)
filled.contour(x,y,z)
```

Bien entendu, vous pouvez consulter l'aide de `R` sur cette fonction afin d'obtenir plus de détails sur celle-ci ainsi que la liste des options pour améliorer votre graphique. Noter que la fonction prise en argument dans `outer()` doit être vectorisée dans chaque argument. Si elle ne l'est pas par défaut, vous pouvez utiliser `Vectorize(FUN = myfun)` afin de la vectoriser. Pour alléger le temps de calcul, ne considérer que la fonction sur $(\mu, \lambda) \in (0, 20)^2$ pour le graphique. La log-vraisemblance vous semble-t-elle strictement concave ? Vérifier votre observation par le calcul.

(b) Afin de maximiser numériquement la log-vraisemblance de l'échantillon généré, on aura besoin d'une valeur de départ pour (μ, λ) dans notre algorithme de maximisation. On propose d'utiliser l'estimateur des moments $(\hat{\mu}_{\mathrm{MM}}, \hat{\lambda}_{\mathrm{MM}})$ comme valeur initiale. Donner une formule analytique pour celui-ci et calculer ce dernier sur l'échantillon généré.

(c) Utiliser la fonction `optim()` avec la méthode "L-BFGS-B" afin de maximiser la log-vraisemblance avec l'estimateur des moments obtenu au point (b) comme point de départ. Noter que la fonction `optim()` effectue une minimisation de la fonction qui lui est donnée comme argument. L'algorithme converge-t-il ? Si oui, quelle valeur obtenez-vous pour (μ, λ) ?

(d) Dans le cas présent, il est en fait possible d'optimiser la log-vraisemblance analytiquement. Donner une formule analytique pour le MLE $(\hat{\mu}_{\mathrm{MLE}}, \hat{\lambda}_{\mathrm{MLE}})$ et calculer ce dernier sur l'échantillon généré. Est-il comparable à la valeur obtenue au point (c) ? Qu'en conclure sur la nature de l'erreur entre les valeurs calculées et la vraie valeur du paramètre sous-jacent à la génération de l'échantillon X ?

(e) Une manière consistante d'estimer la variance asymptotique du MLE est de calculer l'inverse de l'information de Fisher observée. Calculer cette dernière à partir des sorties de `optim()` en spécifiant les bons arguments lors de l'exécution de la fonction (consulter l'aide de la fonction si besoin).

(f) Calculer l'information de Fisher $I_n(\mu_0, \lambda_0)$ évaluée en le vrai paramètre $(\mu_0, \lambda_0) = (2, 5)$ et comparer son inverse avec votre résultat obtenu au point (e).

Exercice 5

Soit $X_1, \dots, X_n$ un échantillon iid de distribution commune Gamma(k, θ) où $(k, \theta) \in (0, \infty)^2$. On propose de calculer le MLE de (k, θ) de manière analytique et numérique à l'aide de la log-vraisemblance profilée.

Dans ce but, on choisit, par exemple, $(k_0, \theta_0) = (5, 1/3)$ et l'on génère un échantillon X de taille $n = 100$ de la distribution associée Gamma$(5, 1/3)$ comme suit :

```
k_true <- 5
theta_true <- 1/3
set.seed(2023)
n <- 100
X <- rgamma(n, shape = k_true, scale = theta_true)
```

(a) Montrer que la log-vraisemblance profilée en k est donnée par

$$\ell_n^p(k) = -n\log(\Gamma(k)) - nk\log(\overline{X}_n/k) + (k-1)\sum_{i=1}^n \log(X_i) - nk.$$

Visualiser ℓ_n^p sur un graphique afin d'étudier sa concavité. Cette fonction vous semble-t-elle concave ?

(b) Maximiser ℓ_n^p en utilisant la fonction `mle()` du package de base `stats4`. Utiliser l'estimateur des moments comme point de départ pour votre algorithme (noter que l'expression analytique pour l'estimateur a déjà été obtenue au TP précédent et qu'il n'est pas nécessaire de la redémontrer ici) et la méthode "L-BFGS-B" pour l'optimisation.

(c) A partir de vos calculs pour le point (a) et de la valeur obtenue au point (b), déduire une valeur pour le MLE de (k, θ).