

CONVERGENCE DES MÉTHODES MONTE CARLO POUR LE MAXIMUM DE VRAISEMBLANCE

Evenson Auguste

2024-11-25

La méthode du maximum de vraisemblance

La méthode du maximum de vraisemblance est une technique statistique permettant d'estimer les paramètres d'un modèle en maximisant la probabilité des données observées. Supposons que les variables X_1, X_2, \dots, X_n aient une fonction de densité conjointe ou une fonction de masse conjointe donnée par :

$$f(x_1, \dots, x_n | \theta),$$

où θ est le paramètre du modèle.

Définition de la vraisemblance

La vraisemblance de θ pour les observations réalisées x_1, \dots, x_n est définie comme :

$$L(\theta) = f(x_1, \dots, x_n | \theta).$$

En pratique, pour simplifier les calculs, on maximise souvent la **log-vraisemblance** $l(\theta)$, définie par :

$$l(\theta) = \log L(\theta).$$

Pour un échantillon X_1, \dots, X_n iid, la fonction de log-vraisemblance devient :

$$l(\theta) = \sum_{i=1}^n \log f(x_i | \theta).$$

Étapes pour trouver l'estimateur du maximum de vraisemblance (EMV)

1. Calculer la log-densité d'une seule observation :

$$\log f(x | \theta).$$

2. Dériver la log-densité par rapport au paramètre θ :

$$\frac{\partial}{\partial \theta} \log f(x | \theta).$$

3. Faire la somme sur toutes les observations et égaler à zéro :

$$\sum_{i=1}^n \frac{\partial}{\partial \theta} \log f(x_i | \theta) = 0.$$

4. Résoudre pour θ :

- La solution de cette équation est l'estimateur de maximum de vraisemblance $\hat{\theta}$.
-

Exemple : Loi de Poisson

Pour une variable $X \sim \text{Poisson}(\lambda)$, la fonction de masse est donnée par :

$$f(x|\lambda) = \frac{e^{-\lambda} \lambda^x}{x!}.$$

La log-vraisemblance pour un échantillon iid X_1, \dots, X_n est alors :

$$l(\lambda) = \sum_{i=1}^n [x_i \log \lambda - \lambda - \log(x_i!)] .$$

Pour maximiser cette log-vraisemblance, on calcule la dérivée et on résout :

$$\frac{\partial}{\partial \lambda} l(\lambda) = \sum_{i=1}^n \frac{x_i}{\lambda} - n = 0.$$

La solution donne l'estimateur de λ :

$$\hat{\lambda} = \frac{1}{n} \sum_{i=1}^n x_i = \bar{x}.$$

```
# Données des fibres d'amiante
x <- c(31, 29, 19, 18, 31, 28, 34, 27, 34, 30, 16, 18, 26, 27,
      27, 18, 24, 22, 28, 24, 21, 17, 24)

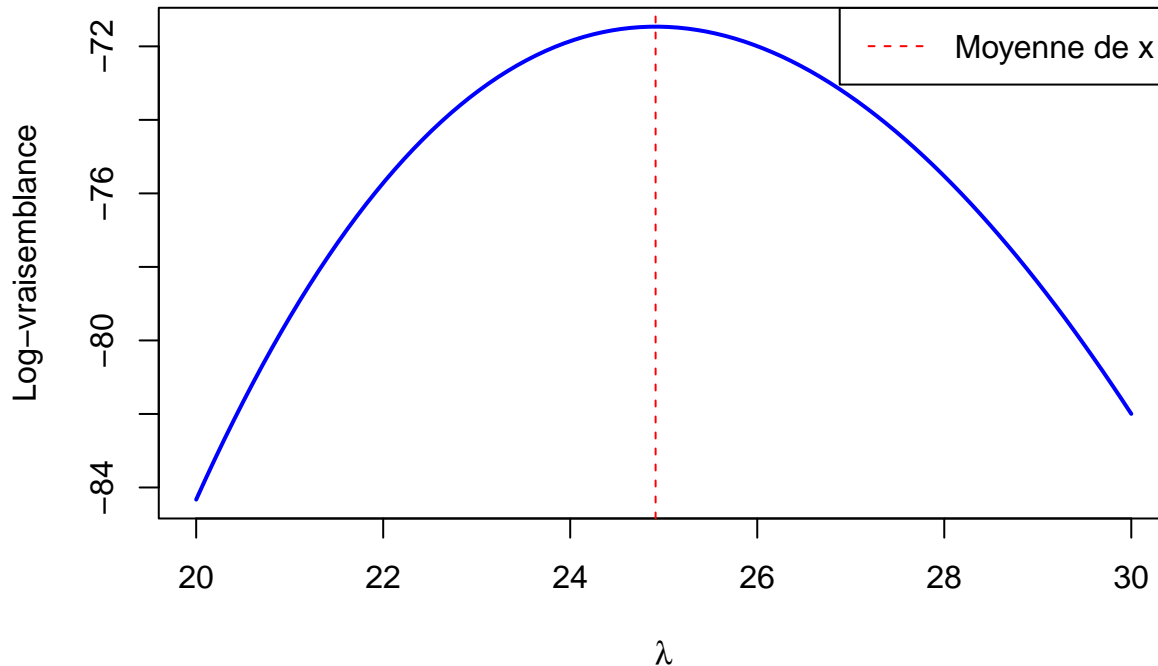
# Taille de l'échantillon
n <- length(x)

# Génération de valeurs de lambda pour calculer la log-vraisemblance
lambda <- seq(20, 30, length = 100)

# Calcul de la log-vraisemblance pour chaque valeur de lambda
log_vraisemblance <- log(lambda) * sum(x) - n * lambda - sum(log(factorial(x)))

# Tracé de la log-vraisemblance
plot(lambda, log_vraisemblance, type = "l", col = "blue",
      lwd = 2, ylab = "Log-vraisemblance", xlab = expression(lambda))

# Ajout d'une ligne pointillée pour la moyenne
abline(v = mean(x), lty = 2, col = "red")
legend("topright", legend = c("Moyenne de x"), col = "red", lty = 2)
```



```
cat("Selon la définition du maximum de vraisemblance :\n",
    "L'estimateur de maximum de vraisemblance est le paramètre qui maximise la probabilité des données o\n",
    "Dans le cas de la loi de Poisson, cet estimateur est la moyenne des observations.\n",
    sprintf("La vraisemblance maximale correspondante est : %.2f\n", mean(x)))

## Selon la définition du maximum de vraisemblance :
## L'estimateur de maximum de vraisemblance est le paramètre qui maximise la probabilité des données o
## Dans le cas de la loi de Poisson, cet estimateur est la moyenne des observations.
## La vraisemblance maximale correspondante est : 24.91
```

Introduction aux Méthodes Monte Carlo par Chaînes de Markov (MCMC)

Les méthodes Monte Carlo par chaînes de Markov (MCMC) sont des algorithmes utilisés pour échantillonner des distributions complexes, particulièrement lorsque ces distributions ne peuvent pas être échantillonnées directement ou évaluées analytiquement. Ces méthodes sont largement employées en statistique, en apprentissage automatique et en physique statistique pour résoudre des problèmes impliquant des distributions de probabilités complexes.

Algorithme de Metropolis-Hastings

En 1970, quelques années après les travaux de Metropolis et ses collaborateurs, Hastings (1970) s'est penché sur le sujet. Il a étendu l'application de l'algorithme de Metropolis à des cas plus généraux.

La grande différence entre l'algorithme de Metropolis et celui dit de Metropolis-Hastings réside essentiellement dans la relaxation de l'hypothèse exigeant une distribution instrumentale symétrique, i.e., $q(\tilde{x} | x) = q(x | \tilde{x})$.

L'algorithme, qui ressemble beaucoup à celui de Metropolis, peut être décrit comme suit :

1. Initialiser $x^{(0)}$, soit le premier élément de la chaîne.
2. Poser $i \leftarrow 1$.
3. Simuler un candidat $\tilde{x} \sim q(\cdot | x^{(i-1)})$, qui dépend de la valeur précédente $x^{(i-1)}$.
4. Calculer la valeur :

$$\alpha = \min \left\{ 1, \frac{f(\tilde{x})}{f(x^{(i-1)})} \cdot \frac{q(x^{(i-1)} | \tilde{x})}{q(\tilde{x} | x^{(i-1)})} \right\}.$$

5. Accepter \tilde{x} avec une probabilité α , telle que :

$$x^{(i)} = \begin{cases} \tilde{x} & \text{avec la probabilité } \alpha, \\ x^{(i-1)} & \text{sinon.} \end{cases}$$

En pratique, on peut utiliser une loi uniforme sur $[0, 1]$ pour effectuer cette étape.

6. Incréments $i \leftarrow i + 1$ et revenir à l'étape 3.

Différences avec l'algorithme de Metropolis

Essentiellement, la différence pratique entre les deux algorithmes réside dans le calcul de α . L'algorithme de Metropolis-Hastings est plus général car il autorise des distributions instrumentales $q(\cdot | \cdot)$ asymétriques.

Variantes de l'algorithme

Il existe plusieurs variantes dérivées de l'algorithme de Metropolis-Hastings. Par exemple, une variante utilise une distribution instrumentale indépendante de $x^{(i-1)}$, i.e., $q(\tilde{x})$. Cette version est appelée **Independence Chain Metropolis-Hastings**.

Objectif de l'étape suivante

Dans l'étape suivante, nous allons appliquer l'algorithme de Metropolis-Hastings pour résoudre le problème jouet précédent basé sur une distribution de Poisson. L'objectif est de : 1. Échantillonner le paramètre λ de la loi de Poisson à partir de la log-vraisemblance, qui est notre distribution cible. 2. Utiliser ces échantillons pour estimer λ via le maximum de vraisemblance (MLE). 3. Visualiser la distribution des échantillons et vérifier la convergence vers l'estimation théorique.

Cette approche illustrera comment les MCMC peuvent être utilisés pour résoudre des problèmes simples et complexes en statistique.

```
set.seed(123)

# Données des fibres d'amiante
x <- c(31, 29, 19, 18, 31, 28, 34, 27, 34, 30, 16, 18, 26, 27,
      27, 18, 24, 22, 28, 24, 21, 17, 24)
n <- length(x)

# Log-vraisemblance pour la loi de Poisson
log_likelihood <- function(lambda) {
  if (lambda <= 0) return(-Inf) # Contraindre lambda > 0
  sum(x) * log(lambda) - n * lambda
}

# Metropolis-Hastings
metropolis_hastings <- function(log_likelihood, init, n_iter, proposal_sd) {
  samples <- numeric(n_iter)
  samples[1] <- init

  for (i in 2:n_iter) {
    # Proposer une nouvelle valeur
    proposal <- rnorm(1, mean = samples[i - 1], sd = proposal_sd)

    # Calcul du ratio d'acceptation
    log_acceptance_ratio <- log_likelihood(proposal) - log_likelihood(samples[i - 1])
```

```

    # Accepter ou rejeter
    if (log(runif(1)) < log_acceptance_ratio) {
      samples[i] <- proposal
    } else {
      samples[i] <- samples[i - 1]
    }
  }

  return(samples)
}

# Paramètres de l'algorithme
init <- mean(x) # Initialisation proche de la moyenne des données
n_iter <- 10000 # Nombre d'itérations
proposal_sd <- 1 # Écart-type de la distribution de proposition

# Exécuter Metropolis-Hastings
samples <- metropolis_hastings(log_likelihood, init, n_iter, proposal_sd)

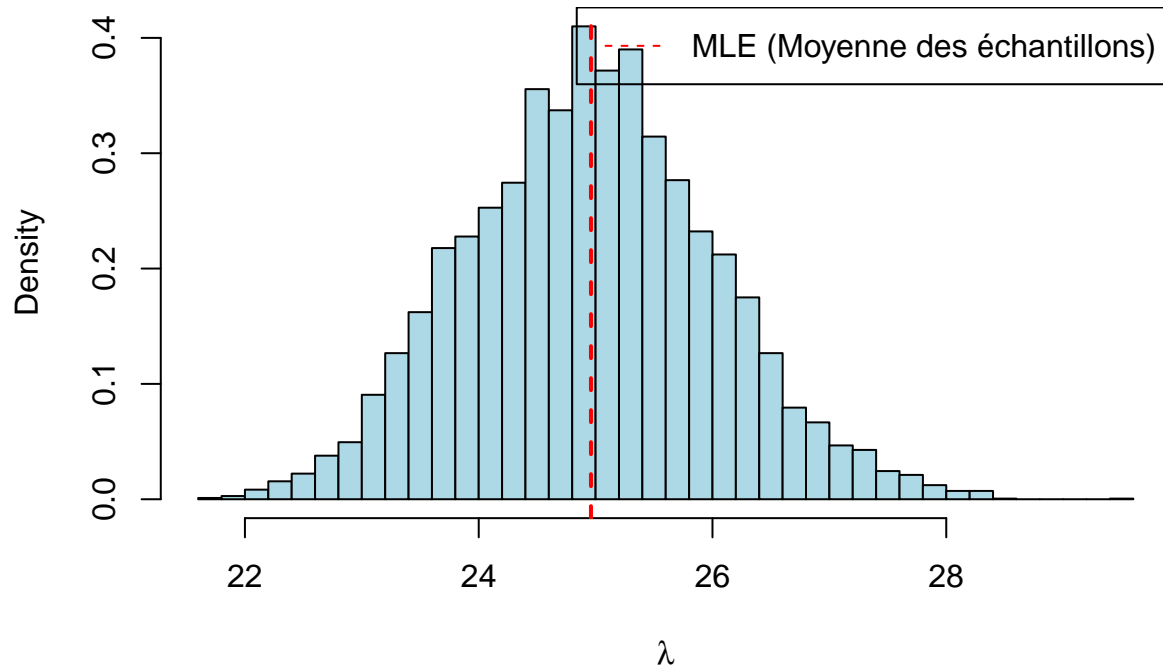
# Retirer les premières valeurs (burn-in)
burn_in <- 1000
samples_post_burnin <- samples[-(1:burn_in)]

# Estimation du MLE par la moyenne des échantillons
lambda_mle <- mean(samples_post_burnin)

# Tracer l'histogramme des échantillons
hist(samples_post_burnin, breaks = 30, probability = TRUE, col = "lightblue",
      main = expression(paste("Échantillons de ", lambda, " via Metropolis-Hastings")),
      xlab = expression(lambda))
abline(v = lambda_mle, col = "red", lwd = 2, lty = 2)
legend("topright", legend = c("MLE (Moyenne des échantillons)"), col = "red", lty = 2)

```

Échantillons de λ via Metropolis-Hastings



```
# Afficher les résultats
cat("Estimation de lambda via Metropolis-Hastings :\n",
    sprintf("Lambda estimé (MLE) : %.2f\n", lambda_mle))
```

```
## Estimation de lambda via Metropolis-Hastings :
## Lambda estimé (MLE) : 24.96
```