Ejercicio 5 - Estadística Computacional

Byron Caices Lima

2023-11-19

Método de Máxima Verosimilitud

Ejercicio 1

Paso 0

Crear los datos que se van a utilizar, para que puedan ver que sus resultados estén acertados se establece un lambda y se crea la distribución con eso

```
# Settear seed
# Esto sirve para que se mantengan los mismos números al recargar el código
set.seed(10)

# Tamaño de la muestra N
N = 2000

# Definimos lambda = 10
lambda = 6

# Generamos datos para la distribucion de Poisson
datos = rpois(n = N, lambda = lambda)
```

Paso 1

La función de masa de probabilidad de Poisson para una variable aleatoria X con parámetro λ es:

$$f(X, \lambda) = \frac{e^{-\lambda}\lambda^X}{X!}$$

La distribución conjunta es entonces:

$$f(X_1, X_2, \ldots, X_n; \lambda) = \prod_{i=1}^{n} \frac{e^{-\lambda}\lambda^{X_i}}{X_i!}$$

Al expandir y simplificar usando las propiedades de los exponentes y la independencia de las X_i, podemos escribir la distribución conjunta como:

$$f(X_1, X_2, \ldots, X_n; \lambda) = e^{-n\lambda}\lambda^{\sum_{i=1}^{n} X_i} \prod_{i=1}^{n} \frac{1}{X_i!}$$

Donde n es el número total de variables aleatorias y $\sum_{i=1}^{n} X_i$ es la suma de todas las variables aleatorias X_i.

Paso 2

Aplicando el logaritmo natural ($\ln$) a esta función de verosimilitud conjunta, obtenemos:

$$\ln L(X_1, X_2, \ldots, X_n; \lambda) = \ln\left(e^{-n\lambda}\lambda^{\sum_{i=1}^{n} X_i}\prod_{i=1}^{n}\frac{1}{X_i!}\right)$$

$$\ln L(X_1, X_2, \ldots, X_n; \lambda) = \ln(e^{-n\lambda}) + \ln(\lambda^{\sum_{i=1}^{n} X_i}) + \ln\left(\prod_{i=1}^{n}\frac{1}{X_i!}\right)$$

Usando las propiedades de los logaritmos, esto se simplifica a:

$$\ln L(X_1, X_2, \ldots, X_n; \lambda) = -n\lambda + \left(\sum_{i=1}^{n} X_i\right)\ln(\lambda) - \sum_{i=1}^{n}\ln(X_i!)$$

Dado que la suma de los logaritmos es el logaritmo del producto, la expresión final en sería:

$$lnL(X_1, X_2, \ldots, X_n; \lambda) = -n\lambda + \left(\sum_{i=1}^{n} X_i\right)\ln(\lambda) - \sum_{i=1}^{n}\ln(X_i!)$$

Paso 3

Derivar la funcion

Para encontrar el estimador de máxima verosimilitud de λ, necesitamos tomar la derivada de la función log-verosimilitud con respecto a λ y luego igualarla a cero para resolver.

La derivada de esta función con respecto a λ es:

$$\frac{d}{d\lambda}\ln L(X_1, X_2, \ldots, X_n; \lambda) = \frac{d}{d\lambda}\left(-n\lambda + \left(\sum_{i=1}^{n} X_i\right)\ln(\lambda) - \sum_{i=1}^{n}\ln(X_i!)\right)$$

Al derivar término por término, obtenemos:

$$\frac{d}{d\lambda}(-n\lambda) = -n$$

$$\frac{d}{d\lambda}\left(\left(\sum_{i=1}^{n} X_i\right)\ln(\lambda)\right) = \sum_{i=1}^{n} X_i\frac{1}{\lambda}$$

$$\frac{d}{d\lambda}\left(-\sum_{i=1}^{n}\ln(X_i!)\right) = 0$$

Por lo tanto, la derivada de la función log-verosimilitud con respecto a λ es:

$$\frac{d}{d\lambda}\ln L(X_1, X_2, \ldots, X_n; \lambda) = -n + \frac{\sum_{i=1}^{n} X_i}{\lambda}$$

Paso 4

Para encontrar el valor que maximiza la función log-verosimilitud, igualamos esta derivada a cero y resolvemos para λ:

$$-n + \frac{\sum_{i=1}^{n} X_i}{\lambda} = 0$$

$$\Rightarrow \frac{\sum_{i=1}^{n} X_i}{\lambda} = n$$

$$\Rightarrow \lambda = \frac{\sum_{i=1}^{n} X_i}{n}$$

El estimador de máxima verosimilitud para λ es entonces el promedio de las observaciones X_i.

Sin embargo, existe la funcion optim en R que deriva y maximiza por nosotros, es decir, ejecuta por sí sola el paso 3 y 4

Primero creamos la funcion log:

```
neg_log_likelihood = function(lambda=lambda, x=datos,n=N) {
  sum_lambda = sum(x)
  # la función de log-verosimilitud para la distribución de Poisson
  log_likelihood = -n * lambda + sum_lambda * log(lambda) - sum(log(factorial(x)))
  return(-log_likelihood) # Negamos porque optim minimiza
}
```

Luego procedermos a determinar el máximo verosímil:

- fn: La función que se va a optimizar, en este caso, exponencial_log.
- par: La estimación inicial del parámetro, en este caso, c(1).
- lower: El límite inferior para los parámetros.
- upper: El límite superior para los parámetros.
- hessian: Si se debe calcular o no el hessiano.
- method: El método de optimización, en este caso, "L-BFGS-B".
- n: El tamaño de la muestra.
- x: La entrada X, en este caso, exponential_data.

```
optim_result = optim(
  par = c(1), # valor inicial de lambda
  fn = neg_log_likelihood, # la función a minimizar
  lower = 0, # límite inferior para lambda (no puede ser negativo)
  upper = Inf, # límite superior para lambda
  hessian = TRUE, # queremos el hessiano para la varianza
  method = "L-BFGS-B", # método de optimización
  n = N, # tamaño de la muestra
  x = datos # datos observados
)

# El estimado máximo verosímil para lambda estará en
optim_par = optim_result$par
cat("El valor original de lambda es", lambda, "y el estimador obtenido con máxima verosimilitud es", optim_p
ar)
```

```
## El valor original de lambda es 6 y el estimador obtenido con máxima verosimilitud es 6.014
```

Método de Máxima Verosimilitud

Ejercicio 2

Paso 0

Crear los datos que se van a utilizar, para que puedan ver que sus resultados estén acertados se establece un lambda y se crea la distribución con eso

```
# Establece una semilla para reproducibilidad
set.seed(10)

# Genera datos exponenciales
# Supongamos que lambda (tasa) es 1 por ahora
lambda = 10
N = 2000
exponential_data = rexp(N, rate = lambda)
```

Paso 1

La función de verosimilitud conjunta $L(\mathbf{X}, \alpha)$ es:

$$L(\mathbf{X}, \alpha) = \prod_{i=1}^{n} \alpha e^{-\alpha X_i}$$

Donde $\mathbf{X} = (X_1, X_2, \ldots, X_n)$ es el vector de observaciones.

Paso 2

Para aplicar el logaritmo natural a la función de distribución conjunta que hemos definido previamente, usaríamos la propiedad del logaritmo de la suma de logaritmos cuando aplicamos el logaritmo a un producto:

$$\log L(\mathbf{X}, \alpha) = \log\left(\prod_{i=1}^{n} \alpha e^{-\alpha X_i}\right) = \sum_{i=1}^{n} \log(\alpha e^{-\alpha X_i})$$

Ahora, aplicamos las propiedades del logaritmo para separar los términos:

$$= \sum_{i=1}^{n} (\log(\alpha) - \alpha X_i)$$

Esto se simplifica aún más como:

$$= \sum_{i=1}^{n} \log(\alpha) - \alpha \sum_{i=1}^{n} X_i$$

Y finalmente, como la suma de logaritmos es simplemente n veces el logaritmo de α, obtenemos:

$$= n \log(\alpha) - \alpha \sum_{i=1}^{n} X_i$$

Paso 3

Derivar la función

Dada la función de log-verosimilitud:

$$\ell(\alpha) = n \log(\alpha) - \alpha \sum_{i=1}^{n} X_i$$

La derivada de $\ell(\alpha)$ con respecto a α es:

$$\frac{d\ell(\alpha)}{d\alpha} = \frac{d}{d\alpha}\left(n \log(\alpha) - \alpha \sum_{i=1}^{n} X_i\right)$$

Aplicando la derivada, obtenemos:

$$\frac{d\ell(\alpha)}{d\alpha} = \frac{n}{\alpha} - \sum_{i=1}^{n} X_i$$

Paso 4

Esta derivada se iguala a cero para encontrar el valor de α que maximiza la función de log-verosimilitud:

$$\frac{n}{\alpha} - \sum_{i=1}^{n} X_i = 0$$

Resolviendo para α, obtenemos:

$$\frac{n}{\alpha} = \sum_{i=1}^{n} X_i$$

$$\alpha = \frac{n}{\sum_{i=1}^{n} X_i}$$

Por lo tanto, el estimador de máxima verosimilitud de α es el recíproco del promedio de las observaciones de la muestra X_i. Esto tiene sentido en el contexto de la distribución exponencial, ya que el parámetro α (o la tasa λ) es el inverso de la media.

Sin embargo, existe la funcion optim en R que deriva y maximiza por nosotros, es decir, ejecuta por sí sola el paso 3 y 4

Primero creamos la funcion log:

Luego procedermos a determinar el máximo verosímil:

- fn: La función que se va a optimizar, en este caso, exponencial_log.
- par: La estimación inicial del parámetro, en este caso, c(1).
- lower: El límite inferior para los parámetros.
- upper: El límite superior para los parámetros.
- hessian: Si se debe calcular o no el hessiano.
- method: El método de optimización, en este caso, “L-BFGS-B”.
- n: El tamaño de la muestra.
- x: La entrada X, en este caso, exponential_data.

```r
# Función log-verosimilitud para la distribución exponencial
exponencial_log = function(alpha=lambda, x=exponential_data, n=N) {
  exp_log = (n * log(alpha))-(alpha*sum(x))
  return(-exp_log) # El signo negativo es porque optim() minimiza por defecto
}

# Usa optim para encontrar el estimador de máxima verosimilitud de alpha
resultado_optim = optim(fn=exponencial_log,
                       par=c(1),
                       lower = c(-Inf, -Inf),
                       upper = c(Inf, Inf),
                       hessian= TRUE,
                       method = "L-BFGS-B",
                       n = N,
                       x = exponential_data
)


resultado_optim_par = resultado_optim$par
cat("El valor original de lambda es", lambda, "y el estimador obtenido con máxima verosimilitud es", resulta
do_optim_par)
```

```
## El valor original de lambda es 10 y el estimador obtenido con máxima verosimilitud es 9.936868
```