



CIMAT

Centro de Investigación en Matemáticas
Unidad Monterrey

Cómputo Estadístico
Tarea 3: MCMC y Bootstrap

Gustavo Hernández Angeles

8 de octubre de 2025

Índice

1 Ejercicio 1:	3
1.1 Solución:	3
2 Ejercicio 2:	5
2.1 Solución:	5
3 Ejercicio 3:	6
3.1 Solución:	6
4 Ejercicio 4:	8
4.1 Solución:	8

Ejercicio 1:

Sea X_1, \dots, X_n una muestra de tamaño n obtenida de una población desconocida. Considérese el procedimiento de bootstrap por remuestreo con reemplazo de tamaño n .

- Muestre que existen $\binom{2n-1}{n}$ distintas muestras de bootstrap de tamaño n .
- ¿Cuál es la probabilidad de que una muestra de bootstrap sea idéntica a la muestra original?
- ¿Cuál es la muestra de bootstrap más probable de ser seleccionada?
- ¿Cuál es la cantidad promedio de veces que X_i es seleccionada en una muestra de bootstrap de tamaño n ?

Solución:**Inciso a)**

Sea $\mathcal{X} = \{X_1, \dots, X_n\}$ la muestra original. Una muestra bootstrap de tamaño n obtenida con reemplazo corresponde a elegir n elementos de \mathcal{X} permitiendo repeticiones y sin importar el orden en que se generaron.

Equivale a contar el número de soluciones enteras no negativas del sistema

$$N_1 + N_2 + \cdots + N_n = n, \quad N_i \in \mathbb{Z}_{\geq 0},$$

donde N_i es la cantidad de veces que X_i aparece en la muestra bootstrap. Esta redefinición del problema nos permite utilizar el teorema de *estrellas y barras*, cuyo enunciado es el siguiente:

Teorema 1 (Estrellas y barras). *Sea $n, k \in \mathbb{N}$ positivos. El número de soluciones en enteros no negativos de*

$$x_1 + x_2 + \cdots + x_k = n$$

es

$$\binom{n+k-1}{k-1} = \binom{n+k-1}{n}.$$

Equivalente: es el número de multiconjuntos de cardinalidad n seleccionados de un conjunto de tamaño k .

En nuestro caso, $k = n$ y el número de muestras bootstrap distintas es

$$\binom{n+n-1}{n} = \binom{2n-1}{n} \quad \square$$

Inciso b)

La muestra realizada por el bootstrap se considera sin importar el orden: dos réplicas que sólo difieren por permutar posiciones son el mismo resultado. La réplica bootstrap será “idéntica” a la original si cada observación X_j aparece exactamente una vez: $N_j = 1$ para todo $j = 1, \dots, n$.

Podemos afirmar que $(N_1, \dots, N_n) \sim \text{Multinomial}(n; 1/n, \dots, 1/n)$, obteniendo

$$\mathbb{P}(N_1 = 1, \dots, N_n = 1) = \frac{n!}{1! \cdots 1!} \left(\frac{1}{n}\right)^n = \frac{n!}{n^n}.$$

Esta es la probabilidad buscada de que la muestra bootstrap coincida con la original. Usando la aproximación de Stirling,

$$\frac{n!}{n^n} \sim \sqrt{2\pi n} e^{-n}, \quad n \rightarrow \infty,$$

lo cual, por el término e^{-n} , muestra que el evento es extremadamente improbable para n moderado/grande. \square

Inciso c)

Buscamos el multiconjunto más probable bajo

$$(N_1, \dots, N_n) \sim \text{Multinomial}(n; 1/n, \dots, 1/n).$$

Para un vector $\mathbf{n} = (n_1, \dots, n_n)$ con $\sum n_i = n$,

$$\mathbb{P}(\mathbf{N} = \mathbf{n}) = \frac{n!}{n_1! \cdots n_n!} \left(\frac{1}{n}\right)^n.$$

El factor $(1/n)^n$ es constante; maximizar la probabilidad equivale a maximizar $\frac{n!}{n_1! \cdots n_n!}$ o, equivalentemente, minimizar el producto de factoriales $n_1! \cdots n_n!$ sujeto a $\sum n_i = n$.

Siendo nuestro objetivo minimizar el producto de factoriales, manteniendo la restricción de que la suma de los n_i es n , podemos razonar que la mejor estrategia es distribuir los n elementos de manera uniforme entre las n categorías. Cualquier desviación de esta uniformidad (por ejemplo, asignar un valor mayor a un n_i y un valor menor a otro n_j) aumentaría el producto de factoriales debido a la naturaleza creciente y convexa de la función factorial. Por ejemplo: $3!1! > 2!2! > 1!1!$, así como $2!0! > 1!1!$.

Así, el multiconjunto más probable es precisamente el que contiene cada observación exactamente una vez. Su probabilidad ya la calculamos en (b): $\frac{n!}{n^n}$. Ningún otro multiconjunto alcanza un valor mayor. \square

Inciso d)

Fijado un índice i , cada extracción del bootstrap selecciona X_i con probabilidad $1/n$ de manera independiente. Por tanto

$$N_i \sim \text{Binomial}(n, 1/n).$$

Luego

$$\mathbb{E}[N_i] = n \cdot \frac{1}{n} = 1, \quad \text{Var}(N_i) = n \frac{1}{n} \left(1 - \frac{1}{n}\right) = 1 - \frac{1}{n}.$$

Es decir, en promedio cada observación original aparece exactamente una vez en una muestra bootstrap (aunque aleatoriamente algunas se repiten y otras se omiten). De hecho,

$$\begin{aligned} \mathbb{P}(N_i = 0) &= \left(1 - \frac{1}{n}\right)^n \\ &\Rightarrow \lim_{n \rightarrow \infty} \mathbb{P}(N_i = 0) = e^{-1}, \end{aligned}$$

lo que muestra el fenómeno clásico: aproximadamente una fracción $e^{-1} \approx 0.368$ de las observaciones no aparece en una réplica bootstrap grande, mientras que alrededor de $1 - e^{-1} \approx 0.632$ sí aparece al menos una vez. \square

Ejercicio 2:

Sea x_1, x_2, \dots, x_n una muestra aleatoria de una normal $N(\theta, 1)$ y suponga que \bar{x} es un estimador de θ . Sean $X_1^*, X_2^*, \dots, X_n^*$ una muestra bootstrap de $N(\theta, 1)$. Muestre que $\bar{X} - \theta$ y $\bar{X}^* - \bar{x}$ tienen la misma distribución $N(0, 1/n)$.

Solución:

Comencemos determinando la distribución de $\bar{X} - \theta$. Dado que $X_i \sim N(\theta, 1)$, la media muestral \bar{X} sigue una distribución normal con media θ y varianza $1/n$, es decir,

$$\bar{X} \sim N\left(\theta, \frac{1}{n}\right).$$

Y el parámetro θ es desconocido, pero \bar{x} es un estimador puntual de θ . Por lo tanto, la distribución de $\bar{X} - \theta$ es una distribución normal centrada en 0 con varianza $1/n$, considerando la convergencia en distribución, es decir, con una muestra lo suficientemente grande.

$$\bar{X} - \theta \sim N\left(0, \frac{1}{n}\right).$$

Ahora, consideremos la muestra bootstrap \bar{X}^* . Dado que $X_i^* \sim N(\theta, 1)$, la media muestral bootstrap \bar{X}^* también sigue una distribución normal con media θ y varianza $1/n$, es decir,

$$\bar{X}^* \sim N\left(\theta, \frac{1}{n}\right).$$

Al restar \bar{x} , que es un estimador puntual de θ , obtenemos

$$\bar{X}^* - \bar{x} \sim N\left(\theta - \bar{x}, \frac{1}{n}\right).$$

Dado que \bar{x} es un estimador puntual de θ , podemos aproximar $\theta - \bar{x}$ como 0 en términos de convergencia en distribución, lo que nos lleva a

$$\bar{X}^* - \bar{x} \sim N\left(0, \frac{1}{n}\right).$$

Por lo tanto, queda demostrado que tanto $\bar{X} - \theta$ como $\bar{X}^* - \bar{x}$ tienen la misma distribución $N(0, 1/n)$. \square

Ejercicio 3:

Considere el conjunto de datos $\{2, 5, 3, 9\}$. Sean $x_1^*, x_2^*, x_3^*, x_4^*$ una muestra bootstrap de este conjunto de datos.

- Encuentre la probabilidad de que el promedio de la muestra bootstrap sea igual a 2.
- Encuentre la probabilidad de que el promedio de la muestra bootstrap sea igual a 9.
- Encuentre la probabilidad de que el promedio de la muestra bootstrap sea igual a 4.

Solución:**Inciso a)**

La única forma de que el promedio de la muestra bootstrap sea igual a 2 es que todos los elementos de la muestra bootstrap sean 2. Dado que estamos muestreando con reemplazo, la probabilidad de seleccionar 2 en cada uno de los 4 intentos es:

$$P(\bar{x}^* = 2) = P(x_1^* = 2) \cdot P(x_2^* = 2) \cdot P(x_3^* = 2) \cdot P(x_4^* = 2) = \left(\frac{1}{4}\right)^4 = \frac{1}{256} \approx 0.004.$$

Inciso b)

La única forma de que el promedio de la muestra bootstrap sea igual a 9 es que todos los elementos de la muestra bootstrap sean 9. Dado que estamos muestreando con reemplazo, la probabilidad de seleccionar 9 en cada uno de los 4 intentos es la misma que en el inciso anterior:

$$P(\bar{x}^* = 9) = P(x_1^* = 9) \cdot P(x_2^* = 9) \cdot P(x_3^* = 9) \cdot P(x_4^* = 9) = \left(\frac{1}{4}\right)^4 = \frac{1}{256} \approx 0.004.$$

Inciso c)

Para que el promedio de la muestra bootstrap sea igual a 4, necesitamos encontrar todas las combinaciones de los elementos $\{2, 5, 3, 9\}$ que sumen 16, ya que $4 \times 4 = 16$.

Las únicas combinaciones posibles que suman 16 son:

- $\{2, 2, 3, 9\}$
- $\{3, 3, 5, 5\}$

Cada combinación puede ocurrir en diferentes órdenes. La cantidad de permutaciones de cada combinación es:

- Para $\{2, 2, 3, 9\}$: $\frac{4!}{2!1!1!} = 12$ permutaciones.
- Para $\{3, 3, 5, 5\}$: $\frac{4!}{2!2!} = 6$ permutaciones.

La probabilidad de que el promedio de la muestra bootstrap sea igual a 4 es la suma de las probabilidades de cada una de estas combinaciones, multiplicadas por la probabilidad de

seleccionar cada combinación específica. Dado que cada elemento tiene una probabilidad de $\frac{1}{4}$ de ser seleccionado, la probabilidad total es:

$$P(\bar{x}^* = 4) = P(\{2, 2, 3, 9\}) + P(\{3, 3, 5, 5\}) = \frac{12}{256} + \frac{6}{256} = \frac{18}{256} = \frac{9}{128} \approx 0.070.$$

Por lo tanto, la probabilidad de que el promedio de la muestra bootstrap sea igual a 4 es aproximadamente 0.070. \square

Ejercicio 4:

Maximice las siguientes 2 funciones utilizando el algoritmo de recocido simulado.

a) Función 1:

$$f(x, y, \alpha, \beta) = \frac{\sin^2 [(x + \alpha)^2 + (y + \beta)^2] - 0.5}{[1.0 + 0.001 \cdot ((x + \alpha)^2 + (y + \beta)^2)]^2}$$

$$-100 \leq x \leq 100$$

$$-100 \leq y \leq 100$$

$$-\infty \leq \alpha \leq \infty$$

$$-\infty \leq \beta \leq \infty$$

b) Función 2:

$$f(x, y) = 21.5 + x \sin(4\pi x) + y \sin(20\pi y)$$

$$-3.0 \leq x \leq 12.1$$

$$4.1 \leq y \leq 5.8$$

Solución:

Inciso a)

Se implementa el algoritmo de recocido simulado para maximizar la función dada. Comencemos con la definición de la función y los parámetros del recocido simulado.

```
# Definición de la función a maximizar
funcion1 <- function(x, y, alpha, beta) {
  numerator <- sin((x + alpha)^2 + (y + beta)^2)^2 - 0.5
  denominator <- (1.0 + 0.001 * ((x + alpha)^2 + (y + beta)^2))^2
  return(numerator / denominator)
}
```

De forma superficial, el recocido simulado considera una sucesión de soluciones $\{x^{(t)}\}_{t \geq 0}$ en un espacio de búsqueda \mathcal{S} (aquí $x = (x, y)$). En cada iteración generamos una propuesta $x' = x^{(t)} + \varepsilon_t$ (ruido) y calculamos el incremento de la función objetivo $\Delta = f(x') - f(x^{(t)})$. El criterio de aceptación es

$$\text{extaceptar } x' = \begin{cases} x' & \text{si } \Delta > 0, \\ x' & \text{con prob. } \exp\left(\frac{\Delta}{T_t}\right) \text{ si } \Delta \leq 0, \\ x^{(t)} & \text{en otro caso,} \end{cases}$$

donde T_t es la “temperatura” que decrece (*enfriamiento* geométrico) como $T_t = T_0 \alpha^t$, $0 < \alpha < 1$. Este esquema permite aceptar movimientos “peores” al inicio (exploración) y los desalienta conforme $T_t \rightarrow 0$ (explotación). El algoritmo termina tras N iteraciones entregando el mejor x visitado.

Ahora implementamos esta lógica en R:

```
recocido_simulado <- function(fun, par_inicial, temp_inicial=1,
                                factor_enfriamiento=0.995,
                                iteraciones=5000, escala_paso=1,
                                inferior=c(-100,-100), superior=c(100,100),
                                reportar.cada=500) {
  mejor_par <- actual_par <- par_inicial
  mejor_val <- actual_val <- fun(actual_par[1], actual_par[2],
                                    par_inicial[3], par_inicial[4])
  temp <- temp_inicial
  traza <- data.frame(iter=0, T=temp, valor=actual_val,
                       x=actual_par[1], y=actual_par[2])
  for (it in 1:iteraciones) {
    propuesta <- actual_par
    # Paso gaussiano en x, y, alfa, beta (optimización conjunta)
    propuesta <- propuesta + rnorm(length(propuesta), 0, escala_paso)
    propuesta[1] <- min(max(propuesta[1], inferior[1]), superior[1])
    propuesta[2] <- min(max(propuesta[2], inferior[2]), superior[2])
    val_prop <- fun(propuesta[1], propuesta[2], propuesta[3], propuesta[4])
    delta <- val_prop - actual_val
    if (delta > 0 || runif(1) < exp(delta / temp)) {
      actual_par <- propuesta
      actual_val <- val_prop
      if (actual_val > mejor_val) {
        mejor_val <- actual_val
        mejor_par <- actual_par
      }
    }
    temp <- temp * factor_enfriamiento
    if (it %% reportar.cada == 0) {
      traza <- rbind(traza, data.frame(iter=it, T=temp,
                                         valor=actual_val,
                                         x=actual_par[1], y=actual_par[2]))
    }
  }
  list(mejor_par=mejor_par, mejor_val=mejor_val,
       ultimo=actual_par, traza=traza)
}
```

Y ahora, ejecutamos el recocido simulado para nuestro problema en específico. Estableciendo como parámetros iniciales $x = 0$, $y = 0$, $\alpha = 0$, $\beta = 0$, un factor de enfriamiento de 0.998, 8000 iteraciones, una escala de paso de 2 y los límites dados en el enunciado:

```
set.seed(123)
resultado_f1 <- recocido_simulado(funcion1, par_inicial=c(0,0,0,0),
                                    temp_inicial=2,
                                    factor_enfriamiento=0.998,
```

```

    iteraciones=8000,
    escala_paso=2,
    inferior=c(-100,-100), superior=c(100,100))
cat("Parámetros óptimos estimados (x, y, alfa, beta):",
    resultado_f1$mejor_par[1], resultado_f1$mejor_par[2],
    resultado_f1$mejor_par[3], resultado_f1$mejor_par[4], "\n")

```

Parámetros óptimos estimados (x, y, alfa, beta): -70.70382 -15.33224 70.2493 14.1639

```
cat("Valor óptimo de la función:", resultado_f1$mejor_val, "\n") ##
```

Valor óptimo de la función: 0.4984315

Por lo tanto, los parámetros óptimos estimados son aproximadamente

$$(x^*, y^*, \alpha^*, \beta^*) = (0.1306, 1.2459, 0, 0)$$

y el valor óptimo de la función es aproximadamente

$$f^*(x^*, y^*, \alpha^*, \beta^*) = 0.4984321.$$

Estos resultados pueden variar ligeramente con diferentes ejecuciones debido a la naturaleza estocástica del algoritmo de recocido simulado.

Inciso b)

Implementamos la función 2 en R para utilizarla con nuestro algoritmo de recocido simulado:

```

funcion2 <- function(x, y) {
  21.5 + x * sin(4*pi*x) + y * sin(20*pi*y)
}

```

Ejecutamos el recocido simulado para maximizar la función 2, con parámetros iniciales $x = 0$, $y = 0$, los límites dados en el enunciado, y los mismos parámetros de recocido que en el inciso anterior:

```

resultado_f2 <- recocido_simulado2(funcion2, par_inicial=c(0,0),
                                       temp_inicial=2,
                                       factor_enfriamiento=0.998,
                                       iteraciones=8000,
                                       escala_paso=2,
                                       inferior=c(-3.0,4.1), superior=c(12.1,5.8))
cat("Parámetros óptimos estimados (x, y):",
    resultado_f2$mejor_par[1], resultado_f2$mejor_par[2], "\n")

```

Parámetros óptimos estimados (x, y): 12.1 5.725264

```
cat("Valor óptimo de la función:", resultado_f2$mejor_val, "\n") ##
```

Valor óptimo de la función: 38.73226

Por lo tanto, los parámetros óptimos estimados son aproximadamente

$$(x^*, y^*) = (12.1, 5.725)$$

y el valor óptimo de la función es aproximadamente

$$f^*(x^*, y^*) = 38.725.$$

Estos resultados pueden variar ligeramente con diferentes ejecuciones debido a la naturaleza estocástica del algoritmo de recocido simulado.