

EST-24107: Simulación

Profesor: Alfredo Garbuno Iñigo — Otoño, 2023 — Reducción de varianza.

Objetivo: En esta sección veremos técnicas para reducir la incertidumbre de nuestros estimadores Monte Carlo. Esto lo logramos con dos estrategias: 1) conociendo mas propiedades de nuestro modelo o 2) haciendo un cambio de medida de referencia.

Lectura recomendada: El contenido de esta sección se puede seguir del Capítulo 5 de Rubinstein and Kroese [3]. Algunos ejemplos son tomados del Capítulo 5 de Asmussen and Glynn [1]. Las propiedades teóricas, si te interesa, del método por importancia las puedes encontrar en el Capítulo 5 de Rubinstein and Kroese [3] o echarle un vistazo a la prueba del Teorema 5.6 en las notas de Sanz-Alonso et al. [4].

1. Introducción

Ya vimos algunos casos donde podemos reducir la varianza de nuestros estimadores Monte Carlo. Esto nos ayuda a mejorar la velocidad y eficiencia estadística de nuestros estimadores y en consecuencia optimizar los recursos computacionales.

Para lograrlo, por ejemplo, consideramos la posibilidad de cambiar la densidad contra la que estamos realizando el proceso de integración. A esta distribución le llamaremos **medida de referencia**.

En esta sección estudiaremos técnicas que nos permitirán reducir la varianza de nuestros estimadores.

1.1. Error Monte Carlo

Lo que estamos tratando de resolver es el problema de cómputo

$$\theta = \mathbb{E}_{\pi}(h(X)), \quad (1)$$

por medio del método Monte Carlo:

1. Generar muestras $X_1, \dots, X_N \stackrel{\text{iid}}{\sim} \pi$.
2. Estimar por medio de $\hat{\theta}_N = (1/N) \sum_{i=1}^N h(X_i)$.

Bajo ciertas condiciones, un intervalo de confianza $(1 - \delta)$ puede construirse por medio

$$[\hat{\theta}_N - z_{1-\delta/2} \text{ee}(\hat{\theta}_N), \hat{\theta}_N + z_{1-\delta/2} \text{ee}(\hat{\theta}_N)], \quad (2)$$

donde podemos calcular el error estándar del estimador (**Error Monte Carlo**).

Hemos jugado con la noción de medir la calidad de nuestro estimador Monte Carlo al observar la longitud del intervalo. Es por esto que utilizamos la **longitud media (HW)** del intervalo de confianza

$$\text{HW} = z_{1-\delta/2} \text{ee}(\hat{\theta}_N). \quad (3)$$

Veremos técnicas de reducción de varianza que nos ayudarán a reducir la longitud media.

El uso de técnicas de reducción de varianza nos obligan a conocer un poco más sobre el modelo que está detrás de nuestro estimador. Esto es, para mejorar nuestra estimación $\hat{\pi}_N^{\text{MC}}(f)$ tenemos que conocer más propiedades tanto de f y/o π . Pues esto nos ayudará a reducir aún mas nuestra varianza.

2. Variables antitéticas

- Lo que buscaremos es inducir una correlación negativa entre secuencias de números pseudo-aleatorios.
- La idea es que al generar números en pares una observación grande en la primera secuencia se compense con una observación pequeña en la segunda.
- El ejemplo típico es sincronizar $u_n \sim \text{U}(0, 1)$ con $u'_n = 1 - u_n$.

Supongamos que queremos estimar

$$\pi(h) = \mathbb{E}[h(X)] = \mathbb{E}[Y], \quad (4)$$

donde tenemos realizaciones de las variables aleatorias Y_1 y Y_2 . El estimador que podemos construir con esos números aleatorios es

$$\hat{\pi}(h) = (Y_1 + Y_2)/2, \quad (5)$$

donde tenemos que

$$\mathbb{V}(\hat{\pi}(h)) = \frac{\mathbb{V}(Y_1) + \mathbb{V}(Y_2) + 2\text{Cov}(Y_1, Y_2)}{4}. \quad (6)$$

Pregunta. En el contexto de las ecuaciones de arriba:

- Si Y_1 y Y_2 son iid. ¿Cuál es el valor de $\mathbb{V}(\hat{\pi}(h))$?
- ¿Cómo reducimos la varianza?

2.1. Intuición

Supongamos que tenemos $(X_1^{(1)}, \dots, X_N^{(1)})$ y $(X_1^{(2)}, \dots, X_N^{(2)})$ en donde, para generar $X_j^{(1)}$, se utilizó u_j y para generar $X_j^{(2)}$ se utilizó $1 - u_j$. Suponemos que tenemos $X_j^{(i)} \stackrel{\text{iid}}{\sim} \mathbb{P}$.

Pregunta. En el contexto de dos colecciones de números aleatorios como se describe arriba:

1. ¿Cuál es el valor esperado de $X_j^{(1)}$ y $X_j^{(2)}$?
2. ¿Cuál es el signo de $\text{Cov}(X_j^{(1)}, X_j^{(2)})$?
3. ¿Son independientes?
4. ¿Qué pasa con los pares $(X_j^{(1)}, X_j^{(2)})$ y $(X_k^{(1)}, X_k^{(2)})$?

Proposición 2.1 (Estimación con variables antitéticas). Por lo anterior, si utilizamos un estimador Monte Carlo con variables antitéticas tendríamos que utilizar

$$X_j = \frac{X_j^{(1)} + X_j^{(2)}}{2}, \quad \bar{X}_N = \frac{1}{N} \sum_{n=1}^N X_n. \quad (7)$$

Por lo tanto, tenemos un estimador que tiene la siguientes propiedades:

1. Es insesgado.
2. Tiene menor varianza que una muestra de $2N$ simulaciones.

2.1.1. Consideraciones:

No siempre se puede lograr el objetivo. Es decir, depende del modelo.

2.2. Fundamento

Queremos estimar

$$\pi(h) = \mathbb{E}[h(U)], \quad (8)$$

donde $U \sim \mathcal{U}(0, 1)$.

- Si suponemos que h es **no decreciente**. Entonces, si U es grande también $h(U)$ será grande. Al mismo tiempo $1 - U$ y $h(1 - U)$ serán pequeños. Esto implica que el término $\text{Cov}(h(U), h(1 - U)) < 0$.
- Si suponemos que h es **no creciente**. Entonces, podemos concluir también que el término $\text{Cov}(h(U), h(1 - U)) < 0$.
- Entonces, una condición suficiente para garantizar que se reduce la varianza es por medio de h una función **monótona**.

Teorema 2.2. Si $h(u_1, \dots, u_m)$ es una función monótona en cada uno de sus argumentos en $[0, 1]^m$, entonces para una colección de variables aleatorias $U_i \stackrel{\text{iid}}{\sim} \mathcal{U}(0, 1)$ tenemos que

$$\text{Cov}(h(U_1, \dots, U_m), h(1 - U_1, \dots, 1 - U_m)) < 0. \quad (9)$$

Proposición 2.3. Podemos extender el resultado anterior bajo el método de la transformada inversa. Es decir, podemos definir

$$h(\mathbb{P}_1^{-1}(U_1), \dots, \mathbb{P}_m^{-1}(U_m)). \quad (10)$$

Por definición la función de acumulación es no decreciente. Y por lo tanto, las inversas también son no decrecientes.

2.3. Ejemplo: Integral en intervalo

Queremos estimar $\int_a^b f(x)dx$. El estimador Monte Carlo sería

$$\hat{\pi}_N^{\text{MC}}(f) = \frac{b-a}{N} \sum_{n=1}^N f(x_n), \quad (11)$$

donde $x_n \sim \mathcal{U}(a, b)$.

```
1 set.seed(108);
2 nsamples <- 10^3;
3 a <- 2; b <- 3;
```

```
4 u ← runif(nsamples, min = a, max = b)
5 x ← dnorm(u)
```

```
1 estimador error.std      N
2 2.153e-02 4.413e-04 1000
```

Si escogemos la mitad (digamos aleatoriamente) y por cada muestra usamos su variable antitética $x'_n = a + (b - x_n)$. Entonces tendríamos

$$\hat{\pi}_N^{\text{AMC}}(f) = \frac{b-a}{N/2} \sum_{n=1}^{N/2} \frac{f(x_n) + f(x'_n)}{2}, \quad (12)$$

```
1 u_ ← a + (b - u)
2 x_ ← dnorm(u_)
3 x ← (x + x_)/2
4 ax ← x[1:(nsamples/2)]
```

```
1 estimador error.st      N
2 2.133e-02 1.113e-04 500
```

Observación (Evaluaciones del método Monte Carlo antitético). Nota como para el estimador $\hat{\pi}_N^{\text{MC}}(f)$ utilizamos N evaluaciones de la función $f(\cdot)$ y para el estimador $\hat{\pi}_N^{\text{AMC}}(f)$ también usamos N evaluaciones de la función $f(\cdot)$. Sin embargo, tenemos que

$$\mathbb{V}(\hat{\pi}_N^{\text{AMC}}(f)) < \mathbb{V}(\hat{\pi}_N^{\text{MC}}(f)). \quad (13)$$

3. Variables de Control

Supongamos que queremos estimar $\mathbb{E}(X)$ y tenemos acceso a una variable aleatoria Y que está correlacionada y se conoce $\nu = \mathbb{E}(Y)$. A Y se le conoce como **variable control** de X .

Sea $X_c = X - a(Y - \nu)$. Entonces

1. $\mathbb{E}(X_c) = \mathbb{E}(X)$.
2. $\mathbb{V}(X_c) = \mathbb{V}(X - a(Y - \nu)) = \mathbb{V}(X) + a^2\mathbb{V}(Y) - 2a\text{Cov}(X, Y)$. Esto implica que

$$\mathbb{V}(X_c) \leq \mathbb{V}(X) \quad \text{si} \quad 2a\text{Cov}(X, Y) > a^2\mathbb{V}(Y). \quad (14)$$

3. El caso particular

$$a^* = \frac{\text{Cov}(X, Y)}{\mathbb{V}(Y)}, \quad (15)$$

que induce la mínima varianza.

4. En este último caso

$$\mathbb{V}(X_c) = (1 - \rho_{X,Y}^2)\mathbb{V}(X). \quad (16)$$

Definición 3.1 (Variable de control). Se conoce a la variable aleatoria Y como variable control si satisface las propiedades descritas anteriormente.

3.1. Consideraciones:

En la práctica no siempre se conoce el valor de $\mathbb{V}(Y)$ y muy difícilmente la $\text{Cov}(X, Y)$, lo que implica que es difícil conocer el valor de a .

En la práctica se puede utilizar un estudio piloto para estimar a [2]. Esto es,

$$\hat{a}_M = \frac{\widehat{\text{Cov}}_M(X, Y)}{\widehat{\mathbb{V}}_M(Y)}. \quad (17)$$

Nota que el estimador resultante para la media de X_c ya no es un estimador insesgado.

3.2. Ejemplo

Supongamos que $X \sim N(0, 1)$ y que $f(X) = \frac{X^6}{1+X^2}$.

- Entonces, utilizando la igualdad

$$\frac{x^6}{1+x^2} = x^4 - x^2 + 1 - \frac{1}{1+x^2}, \quad (18)$$

y podemos aproximar con $Y = g(X) = x^4 - x^2 + 1$.

- Para esta elección tenemos $\mathbb{E}(Y) = 3$.
- Así que el problema se reduce a

$$\mathbb{E}\left[\frac{X^6}{1+X^2}\right] = 3 - \mathbb{E}\left[\frac{1}{1+X^2}\right]. \quad (19)$$

```
1 set.seed(108)
2 x <- rnorm(nsamples)
```

```
1 f_x <- x**6/(1 + x**2)
2 c(estimador = mean(f_x), error.std = sd(f_x)/sqrt(nsamples))
```

```
1 estimador error.std
2 2.3473 0.2798
```

```
1 g_x <- 3 - 1 / (1 + x**2)
2 c(estimador = mean(g_x), error.std = sd(g_x)/sqrt(nsamples))
```

```
1 estimador error.std
2 2.343346 0.008549
```

Pregunta. ¿Por qué estos estimadores dan los mismos números que con el código anterior?

```
1 set.seed(108)
2 x <- rnorm(100 * nsamples)
3 x <- array(x, c(100, nsamples))
4 f_x <- x**6/(1 + x**2)
5 estimadores <- apply(f_x, 1, mean)
6 c(estimador = mean(estimadores), error.std = sd(estimadores))
```

```
1 estimador error.std
2      2.3473      0.2752
```

```
1 g_x <- 3 - 1/(1+x**2)
2 estimadores <- apply(g_x, 1, mean)
3 c(estimador = mean(estimadores), error.std = sd(estimadores))
```

```
1 estimador error.std
2      2.3453      0.0081
```

4. Monte Carlo condicional

Se pueden utilizar algunos resultados teóricos intermedios para algunos casos. A esta técnica también se le conoce como método Rao-Blackwell (por un resultado análogo en inferencia estadística).

Supongamos que nos interesa $\mathbb{E}(f(X))$ y del alguna manera tenemos conocimiento de una variable aleatoria que está relacionada con la original por medio de $\mathbb{E}(f(X)|Z = z)$. Utilizando la propiedad torre podemos calcular

$$\mathbb{E}(f(X)) = \mathbb{E}(\mathbb{E}(f(X)|Z = z)) . \quad (20)$$

Donde además tenemos que

$$\mathbb{V}(f(X)) = \mathbb{V}(\mathbb{E}(f(X)|Z)) + \mathbb{E}(\mathbb{V}(f(X)|Z)) . \quad (21)$$

Lo que buscamos es que:

1. Z pueda ser generado de manera eficiente.
2. Se pueda calcular $\mathbb{E}(f(X)|Z)$.
3. El valor de $\mathbb{E}(\mathbb{V}(f(X)|Z))$ sea grande.

Por lo tanto, el método es:

1. Generar una muestra $Z_1, \dots, Z_N \stackrel{\text{iid}}{\sim} \pi(Z)$.
2. Calcular $\mathbb{E}(f(X)|Z = z_k)$ de manera analítica.
3. Calcular el estimador de $\pi(f) = \mathbb{E}(f(X))$ por medio de

$$\hat{\pi}_N^{\text{CMC}}(f) = \frac{1}{N} \sum_{n=1}^N \mathbb{E}(f(X)|Z = Z_n) . \quad (22)$$

Proposición 4.1 (Varianza del método Monte Carlo condicional). La varianza del estimador Monte Carlo condicional es menor a la varianza del estimador Monte Carlo tradicional y en el peor de los casos se mantiene igual. Es decir,

$$\mathbb{V}\left(\hat{\pi}_N^{\text{CMC}}(f)\right) \leq \mathbb{V}\left(\hat{\pi}_N^{\text{MC}}(f)\right). \quad (23)$$

4.1. Ejemplo: Mezcla Beta-Binomial

Supongamos un modelo Beta-Binomial. Igual que antes asumamos $n = 20$ y $\alpha = 2, \beta = 5$.

```
1 set.seed(108);
2 theta <- rbeta(nsamples, 2, 5)
3 y <- rbinom(nsamples, size = 20, theta)
4 c(estimador = mean(y), error.std = sd(y)/sqrt(nsamples))
```

```
1 estimador error.std
2      5.669      0.053
```

```
1 m_y <- 20 * theta
2 c(estimador = mean(m_y), error.std = sd(m_y)/sqrt(nsamples))
```

```
1 estimador error.std
2      5.668      0.046
```

El porcentaje de reducción de varianza es

```
1 (sd(y) - sd(m_y))/sd(y)
```

```
1 [1] 0.14
```

4.2. Ejemplo: Mezcla Poisson-Beta

Supongamos un modelo de mezcla

```
1 set.seed(108)
2 w <- rpois(nsamples, 10)
3 y <- rbeta(nsamples, w, w**2 + 1)
4 c(estimador = mean(y), error.std = sd(y)/sqrt(nsamples))
```

```
1 estimador error.std
2  0.096535  0.001404
```

```
1 m_y <- w / (w**2 + w + 1)
2 c(estimador = mean(m_y), error.std = sd(m_y)/sqrt(nsamples))
```

```

1  estimador error.std
2  0.098341  0.001019

```

El porcentaje de reducción de varianza es

```

1  [1] 0.2737

```

4.3. Ejemplo: Estimación de densidades (tomado de [1])

Podemos utilizar el método Monte Carlo condicionado para estimar densidades. Por ejemplo, si consideramos que $X_1, \dots, X_k \stackrel{\text{iid}}{\sim} \pi$ y nos interesa $S_k = X_1 + \dots + X_k$. Nos podemos preguntar por la densidad de la suma. Sabemos que la densidad es un objeto infinitesimal $\mathbb{P}(S_k \in dx)$. Y en algunas situaciones no tenemos acceso a éste.

Por ejemplo, consideremos $X_i \sim \text{Pareto}(1, \alpha = 3/2)$. Para este caso, no se puede escribir la densidad de S_k para $k > 1$. Lo que si sabemos es que

$$S_k | S_{k-1} \stackrel{d}{=} X_k | S_{k-1} \sim \text{Pareto}(S_{k-1}, \alpha). \quad (24)$$

Por lo que podemos estimar la densidad de $X_k | S_{k-1}$ para valores, por ejemplo, en $[0, 15]$.

```

1  nsamples <- 5 * 10^3; ngrid <- 1000; k <- 4
2  rpareto <- function(n, alpha) { 1 / runif(n)^(1/alpha) - 1 }
3  dpareto <- function(x, alpha) {
4    ifelse( x >= 0, (alpha / ((x+1)**(alpha + 1))), 0 ) }
5  u <- rpareto( (k-1) * nsamples, alpha = 3/2)
6  u <- array(u, c(k-1, nsamples))
7  S <- apply(u, 2, sum)
8  x <- seq(0.1, 15, length.out = ngrid)

1  estimador <- array(x, c(ngrid,1)) >
2    apply(1, FUN = function(x_){ dpareto(x_ - S, alpha = 3/2) }) >
3    apply(2, mean)
4
5  error.std <- array(x, c(ngrid,1)) >
6    apply(1, FUN = function(x_){ dpareto(x_ - S, alpha = 3/2) }) >
7    apply(2, sd)

```

4.4. Ejemplo: Constructora (Tomado de las notas de Jorge de la Vega)

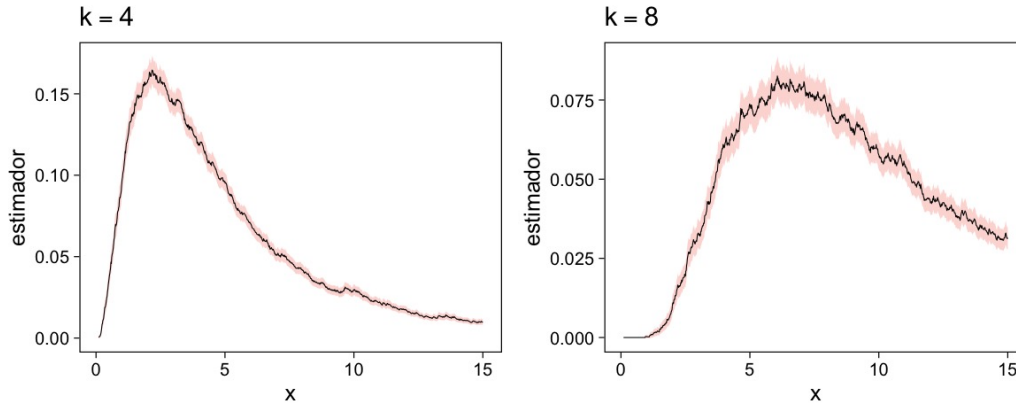
Un proyecto de construcción tiene una duración $X \sim N(\mu, \sigma^2)$ donde, a su vez, $\mu \sim N(10, 16)$ y $\sigma \sim \text{Exp}(1/4)$. La compañía que construye debe pagar 1,000 (USD) por cada día (y prorratea por fracciones del día) que la duración del proyecto excede el contrato de K días. ¿Cuál es el costo esperado por retraso?

Podemos simular

```

1  K <- 20; nsamples <- 10^4
2  sigma <- rexp(nsamples, 1/4)
3  mu <- rnorm(nsamples, mean = 10, sd = 4)
4  x <- rnorm(nsamples, mean = mu, sd = sigma)
5  costo <- 1000 * ifelse( x >= K, 0, x - K)

```


FIGURA 1. Densidad de $x | S_{k-1}$.

```

1  media error.std
2  316.49      21.26

```

Con condicionales, sabemos que podemos considerar $\theta = (\mu, \sigma)$ y evaluar $X|\theta$. Lo que nos lleva a escribir que nuestro estimador será sobre

$$\begin{aligned}
 \mathbb{E}_{X|\theta} [1000 \max\{X - K, 0\}] &= 1000 \int_K^\infty \frac{X - K}{\sqrt{2\pi}\sigma^2} \exp\left[-\frac{1}{2}\left(\frac{x - \mu}{\sigma}\right)^2\right] dx \\
 &= 1000 \int_{K'}^\infty (\sigma\nu + \mu - K) \frac{1}{\sqrt{2\pi}} \exp\left[-\frac{1}{2}\nu^2\right] d\nu \\
 &= 1000 \left[\left(-\frac{\sigma e^{-\frac{\nu^2}{2}}}{\sqrt{2\pi}} \right) \Big|_{K'}^\infty + (\mu - K) \Phi(-K') \right] \\
 &= 1000 [\sigma\phi(K') + (\mu - K)\Phi(-K')] .
 \end{aligned}$$

La estimación utilizando Monte Carlo condicional nos da una estimación (con su métrica de error Monte Carlo).

```

1  media error.std
2  301.920      8.301

```

Lo que lleva a una reducción de varianza

```

1  [1] 0.5645

```

5. Muestreo estratificado

Queremos estimar $\mathbb{E}_\pi[h(X)]$ y supongamos que existe una variable aleatoria discreta Y con soporte y_1, \dots, y_k tal que

1. Las probabilidades $\omega_i = \text{Prob}\{Y = y_i\}$ son conocidas;

2. Para cada i podemos simular de la condicional $\pi_i(X) = \pi(X|Y = y_i)$.

Si queremos usar simulación para estimar $\pi(h)$ entonces utilizaríamos una muestra aleatoria $h(X_1), \dots, h(X_N)$ y utilizaríamos su promedio para estimarlo. La varianza de este estimador sería igual a

$$\mathbb{V}(\hat{\pi}_N^{\text{MC}}(h)) = \frac{\mathbb{V}_{\pi}(h)}{N}. \quad (25)$$

Ahora, si realizamos $N_i = N \times \omega_i$ simulaciones para cada nivel i , y promediamos para cada nivel tendríamos

$$\hat{\pi}_{N,i}^{\text{MC}}(h) = \frac{1}{N_i} \sum_{n=1}^{N_i} h(X_n^{(i)}), \quad X_n^{(i)} \stackrel{\text{iid}}{\sim} \pi_i(X), \quad (26)$$

de tal forma que podemos construir el estimador

$$\hat{\pi}_N^{\text{sMC}}(h) = \sum_{i=1}^k \omega_i \hat{\pi}_{N,i}^{\text{MC}}(h). \quad (27)$$

La varianza de cada término es igual a

$$\mathbb{V}(\hat{\pi}_{N,i}^{\text{MC}}(h)) = \frac{\mathbb{V}(h(X)|Y = y_i)}{N_i}. \quad (28)$$

Por lo tanto la varianza de nuestro estimador es

$$\mathbb{V}(\hat{\pi}_N^{\text{sMC}}(h)) = \sum_{i=1}^k \omega_i^2 \mathbb{V}(\hat{\pi}_{N,i}^{\text{MC}}(h)) \quad (29)$$

$$= \frac{1}{N} \sum_{i=1}^k \omega_i \mathbb{V}(h(X)|Y = y_i) \quad (30)$$

$$= \frac{1}{N} \mathbb{E}[\mathbb{V}(h(X)|Y)]. \quad (31)$$

Utilizando lo que sabemos de Monte Carlo condicional sabemos que tendremos una ganancia de

$$\mathbb{V}(\hat{\pi}_N^{\text{MC}}(h)) - \mathbb{V}(\hat{\pi}_N^{\text{sMC}}(h)) = \frac{1}{N} \mathbb{V}(\mathbb{E}(h(X)|Y)). \quad (32)$$

La ganancia será mayor mientras más afecte el valor de Y el valor esperado de $h(X)$.

5.1. Diseño de experimentos

Notemos que el estimador estratificado tiene una varianza igual a

$$\mathbb{V}(\hat{\pi}_N^{\text{sMC}}(h)) = \sum_{i=1}^k \frac{\omega_i^2}{N_i} \mathbb{V}(h(X)|Y = y_i). \quad (33)$$

Lo cual asume que conocemos los términos individuales.

Sin embargo, usualmente no conoceremos $\mathbb{V}(h(X)|Y = y_i)$ lo que nos lleva a que podríamos usar un pequeño piloto de simulación para poder estimarlos. Denotaremos por s_i^2 dichos estimadores.

Si sabemos que tenemos un presupuesto de N simulaciones y queremos distribuir nuestras simulaciones entre la partición. Entonces podemos resolver el problema de

$$\begin{aligned} & \min \sum_{i=1}^k \omega_i^2 s_i^2 / N_i \\ & \text{sujeto a } \sum_{i=1}^k N_i = N. \end{aligned}$$

El cual tiene una solución

$$\frac{N_i^*}{N} = \frac{\omega_i s_i}{\sum_{j=1}^k \omega_j s_j}. \quad (34)$$

5.2. Ejemplo:

Supongamos que queremos resolver la integral

$$\pi(h) = \int_0^1 h(x) dx. \quad (35)$$

Si definimos

$$Y = j \quad \text{si } \frac{j-1}{N} \leq U < \frac{j}{N}, \quad j = 1, \dots, N, \quad (36)$$

entonces podemos calcular

$$\pi(h) = \frac{1}{N} \sum_{j=1}^N \mathbb{E}[h(U)|Y = j] \quad (37)$$

$$= \frac{1}{N} \sum_{j=1}^N \mathbb{E}[h(U^{(j)})], \quad (38)$$

donde $U^{(j)} \sim U((j-1)/N, j/N)$.

Por lo que en lugar de generar $U_1, \dots, U_N \sim U(0, 1)$ para calcular $\sum_j h(U_j)/N$, podemos construir un mejor estimador por medio de

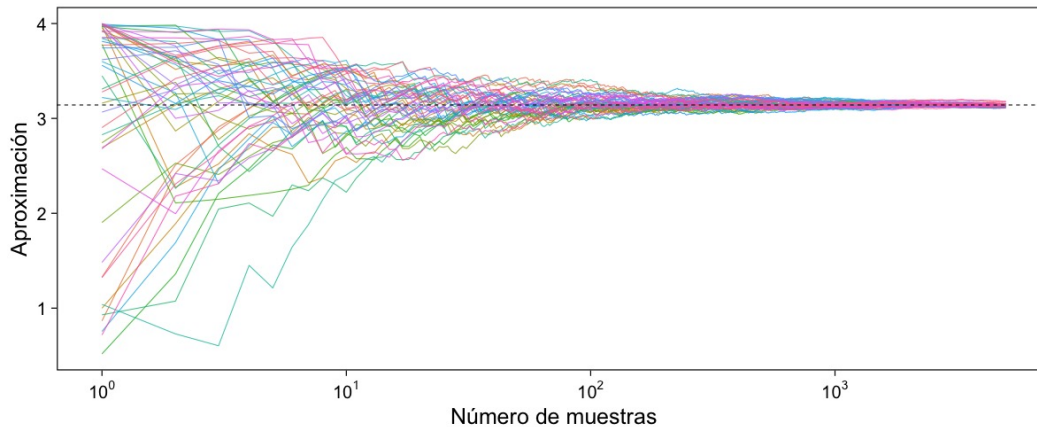
$$\hat{\pi}_N^{sMC}(h) = \frac{1}{N} \sum_{j=1}^N h\left(\frac{U_j + j - 1}{N}\right). \quad (39)$$

5.2.1. Aplicación:

Estimemos π por medio de

$$\frac{\pi}{4} = \mathbb{E}[\sqrt{1 - U^2}]. \quad (40)$$

```
1 nsamples <- 5000
2 h <- function(u) { 4 * sqrt(1 - u**2) }
3 u <- runif(100 * nsamples)
4 u <- array(u, c(100, nsamples))
5 h_u <- h(u)
6 estimador_MC <- apply(h_u, 1, cummean) > t()
```



```

1 runif_estrat <- function(u){
2   x <- c()
3   for (jj in 1:nsamples){
4     x[jj] <- (u[jj] + jj - 1)/nsamples
5   }
6   return(x)
7 }
8 u_strat <- apply(u, 1, runif_estrat) > t()

```

```

1 calcula_antitetic <- function(u){
2   x <- c()
3   for (jj in 1:nsamples){
4     x[jj] <- h((u[jj] + jj - 1)/nsamples) + h((jj - u[jj])/nsamples)
5   }
6   return(0.5 * x)
7 }
8 h_anti <- apply(u, 1, calcula_antitetic) > t()

```

```

1      metodo estimador      error.mc
2 1      vainilla  3.141487 1.400903e-02
3 2      estratificado  3.141592 7.065134e-06
4 3 anti-estratificado  3.141593 6.824089e-07

```

5.3. Post estratificación

Supongamos que hemos simulado N réplicas independientes de una variable aleatoria $X \sim \mathbb{P}$. Supongamos que podemos identificar los grupos, es decir, podemos decir cuando una simulación está asociada a una Y con categoría y_i de tal forma que con N_i contamos cuántas simulaciones pertenecen a la categoría i . Entonces, podríamos escribir

$$\begin{aligned}
 \bar{X}_N &= \frac{1}{N} \sum_{n=1}^N X_n \\
 &= \frac{1}{N} \sum_{i=1}^k N_i \bar{X}_{N_i}^{(i)} \\
 &= \sum_{i=1}^k \frac{N_i}{N} \bar{X}_{N_i}^{(i)}.
 \end{aligned}$$

6. Muestreo por importancia

Supongamos que queremos estimar

$$\pi(h) = \int h(x)\pi(x)dx. \quad (41)$$

Sin embargo, consideremos que evaluar h es ineficiente debido a:

1. Es difícil simular un valor[†] aleatorio de la densidad π .
2. La varianza de h es muy grande.
3. Una combinación de 2. y 3.

Podemos utilizar una distribución ρ tal que para $\rho(x) = 0$ tenemos $\pi(x) = 0$. Entonces, podemos reescribir

$$\pi(h) = \int \frac{h(x)\pi(x)}{\rho(x)}\rho(x)dx \quad (42)$$

$$= \mathbb{E}_\rho \left[\frac{h(x) \cdot \pi(x)}{\rho(x)} \right] \quad (43)$$

$$= \rho(h\omega), \quad (44)$$

donde

$$\omega(x) = \frac{\pi(x)}{\rho(x)}. \quad (45)$$

En aplicaciones usualmente operamos bajo el supuesto que conocemos los pesos **hasta una constante de normalización**. Esto es, podemos **evaluar**

$$\omega(x) = \frac{\pi(x)}{\rho(x)} = \frac{1}{Z}v(x). \quad (46)$$

donde $Z = \int v(x)\rho(x)dx = \rho(v)$. Lo cual nos deja

$$\pi(h) = \frac{\rho(hv)}{\rho(v)}. \quad (47)$$

Muestreo por importancia se basa en aproximar ambas integrales por el método Monte Carlo. Es decir, utilizamos

$$\pi(h) \approx \sum_{n=1}^N \omega_n h(x_n) \quad x_n \stackrel{\text{iid}}{\sim} \rho \quad (48)$$

$$= \hat{\pi}_N^{\text{IS}}(h), \quad (49)$$

donde

$$\omega_n = \frac{v(x_n)}{\sum_{m=1}^N v(x_m)}. \quad (50)$$

Definición 6.1 (Estimador Monte Carlo por importancia). El estimador por importancia $\hat{\pi}_N^{\text{IS}}(h)$ que aproxima $\theta = \pi(h)$ se obtiene al considerar una colección de variables aleatorias $X_i \stackrel{\text{iid}}{\sim} \rho$ tal que el soporte de ρ contiene el soporte de la distribución original π . El estimador

se construye a través de una estimación Monte Carlo de la forma

$$\hat{\pi}_N^{\text{IS}}(h) = \sum_{n=1}^N \omega_n h(x_n) = \frac{\hat{\rho}_N^{\text{MC}}(hv)}{\hat{\rho}_N^{\text{MC}}(v)}, \quad (51)$$

donde a $v(\cdot)$ se le conoce como el cociente no-normalizado de las densidades $\pi(\cdot)/\rho(\cdot)$ y ω_n representan los pesos normalizados de las muestras que se usan en la estimación Monte Carlo por importancia.

La construcción de nuestro estimador de esta manera tiene algunas propiedades interesantes. El estimador $\hat{\pi}_N^{\text{IS}}(h)$ en general **no** es un estimador insesgado (aunque asintóticamente sí).

El estimador que hemos construido asume que conocemos los pesos *hasta* una constante de normalización. En aplicaciones esto es usual, pues la densidad π que define nuestros problemas de integración suele ser muy complicada. Es por esto que la discusión en esta sección lo ha tratado de esta manera.

Para que muestreo por importancia tenga éxito necesitamos que $h(x)\pi(x)/\rho(x)$ tenga una varianza pequeña. Es decir, el objetivo de Muestreo por importancia es lograr que

$$\mathbb{V}(\hat{\pi}_N^{\text{IS}}(h)) \leq \mathbb{V}(\hat{\pi}_N^{\text{MC}}(h)), \quad (52)$$

al utilizar una $\rho(\cdot)$ adecuada.

Proposición 6.2 (Varianza del estimador por importancia). La varianza del estimador por importancia está dado por

$$\mathbb{V}(\hat{\pi}_N^{\text{IS}}(h)) = \frac{1}{N} \left(\mathbb{E} \left[h^2(X) \frac{\pi(X)}{\rho(X)} \right] - \pi(h)^2 \right). \quad (53)$$

De donde vemos que la elección de la densidad de muestreo por importancia, ρ , está asociada a la varianza del estimador $\pi_N^{\text{IS}}(h)$ y por lo tanto la elección es crucial para que tenga éxito.

Observación (Minimizador de muestreo por importancia). Para tener el estimador por importancia, $\mathbb{V}(\hat{\pi}_N^{\text{IS}}(h))$, de varianza mínima *basta* con encontrar el minimizador de

$$\rho^*(x) = \arg \min_{\rho} \mathbb{E} \left[h^2(X) \frac{\pi(X)}{\rho(X)} \right], \quad (54)$$

cuya solución es

$$\rho^*(x) \propto |h(x)|\pi(x). \quad (55)$$

El problema de escoger la mejor ρ^* es que tendríamos que escoger dependiendo del problema (la elección de h y π) y esto no presenta estrategias muy aplicables en la práctica.

1 sessionInfo()

1 R version 4.3.1 (2023-06-16)

```

2 Platform: x86_64-apple-darwin20 (64-bit)
3 Running under: macOS Ventura 13.5.1
4
5 Matrix products: default
6 BLAS: /Library/Frameworks/R.framework/Versions/4.3-x86_64/Resources/lib/
  libRblas.0.dylib
7 LAPACK: /Library/Frameworks/R.framework/Versions/4.3-x86_64/Resources/lib/
  libRlapack.dylib; LAPACK version 3.11.0
8
9 locale:
10 [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
11
12 time zone: America/Mexico_City
13 tzcode source: internal
14
15 attached base packages:
16 [1] stats      graphics  grDevices  datasets  utils      methods
17 [7] base
18
19 other attached packages:
20 [1] scales_1.2.1      patchwork_1.1.2  lubridate_1.9.2  forcats_1.0.0
21 [5] stringr_1.5.0     dplyr_1.1.2      purrr_1.0.1      readr_2.1.4
22 [9] tidyr_1.3.0       tibble_3.2.1     ggplot2_3.4.2    tidyverse_2.0.0
23
24 loaded via a namespace (and not attached):
25 [1] crayon_1.5.2      vctrs_0.6.3      cli_3.6.1
26 [4] rlang_1.1.1       stringi_1.7.12   renv_1.0.0
27 [7] generics_0.1.3    labeling_0.4.2   glue_1.6.2
28 [10] colorspace_2.1-0  hms_1.1.3        fansi_1.0.4
29 [13] grid_4.3.1        munsell_0.5.0    tzdb_0.4.0
30 [16] lifecycle_1.0.3   compiler_4.3.1   timechange_0.2.0
31 [19] pkgconfig_2.0.3   farver_2.1.1     R6_2.5.1
32 [22] tidyselect_1.2.0  utf8_1.2.3       pillar_1.9.0
33 [25] magrittr_2.0.3    tools_4.3.1      withr_2.5.0
34 [28] gtable_0.3.3

```

Referencias

- [1] S. Asmussen and P. W. Glynn. *Stochastic Simulation: Algorithms and Analysis*. Number 57 in Stochastic Modelling and Applied Probability. Springer, New York, 2007. ISBN 978-0-387-30679-7 978-0-387-69033-9. [1](#), [8](#)
- [2] S. S. Lavenberg, T. L. Moeller, and P. D. Welch. Statistical Results on Control Variables with Application to Queueing Network Simulation. *Operations Research*, 30(1):182–202, 1982. ISSN 0030-364X. [5](#)
- [3] R. Y. Rubinstein and D. P. Kroese. *Simulation and the Monte Carlo Method*. [1](#)
- [4] D. Sanz-Alonso, A. M. Stuart, and A. Taeb. Inverse Problems and Data Assimilation. *arXiv:1810.06191 [stat]*, jul 2019. [1](#)