

Métodos de muestreo de densidades log-concavas

Javier Roberto Rubalcava Cortes

September 2024

1 Introducción

El método de aceptación-rechazo (AR) es uno de los procedimientos fundamentales para el muestreo de variables aleatorias con distribuciones complejas. A diferencia del método de inversión, que requiere que la función de distribución acumulativa (CDF) sea analíticamente invertible o al menos factible de invertir numéricamente—una condición rara vez cumplida en situaciones prácticas—el método AR no impone esta restricción. En cambio, AR se basa en la construcción de una función envolvente g , que satisface la condición $f(x) \leq Kg(x)$ para toda x , siendo f la densidad objetivo y c una constante positiva conocida como constante de aceptación [1].

La eficiencia del método AR depende directamente de qué tan ajustada sea esta función envolvente respecto a la densidad objetivo; cuanto más cercana sea la probabilidad de aceptación $\frac{1}{K}$ a 1 más eficiente será el muestreo.

En este proyecto analizaremos diversas desigualdades útiles para densidades log-cóncavas y exploraremos cómo dichas desigualdades pueden emplearse para construir funciones envolventes adecuadas para el método AR. Adicionalmente, estudiaremos métodos específicos para construir funciones envolventes mediante combinaciones de componentes exponenciales, derivadas directamente de la función logarítmica de la densidad, $\log(f)$. Finalmente, presentaremos un algoritmo que utiliza estas funciones envolventes adaptativas para realizar muestreo eficiente mediante aceptación-rechazo.

1.1 Desigualdades para densidades log-cóncavas

Suponemos que la densidad f es log-cóncava con dominio $D = \mathbb{R}$. La primera desigualdad que se expone en [2] solo requiere de la localización de la moda m , entonces:

$$f(x) \leq g_1(x) := M \min(1, \exp(1 - |x - m|/M)) \quad \text{donde } f(m) = M \quad 1.$$

Si además tenemos acceso a la desviación estándar y la media σ^2, μ entonces se propone la siguiente desigualdad:

$$f(x) \leq g_2(x) := \begin{cases} \frac{1}{\sigma} & \text{si } |x - \mu| \leq (1 + \sqrt{3})\sigma \\ \frac{1}{|x - \mu| - \sigma\sqrt{3}} & \text{si } (\sqrt{3} + \sqrt{12})\sigma \geq |x - \mu| \geq (1 + \sqrt{3})\sigma \\ \frac{1}{\sigma\sqrt{12}} \exp\left(\frac{3}{2} - \frac{|x - \mu|}{\sigma\sqrt{12}}\right) & \text{si } |x - \mu| \geq (\sqrt{3} + \sqrt{12})\sigma \end{cases} \quad 2.$$

La función envolvente g_2 esta compuesta por tres segmentos, un segmento uniforme cuando $|x - \mu| \leq (1 + \sqrt{3})\sigma$ y dos colas decrecientes. El área bajo g_2 es:

$$\int_{\mathbb{R}} g_2 = 2(1 + \sqrt{3}) + 2(\log(\sqrt{12})) + \frac{\sqrt{12}}{\sqrt{3}} \approx 9.94 \quad 3.$$

El peso de cada una de las componentes es de:

$$p_1 = 2\frac{1+\sqrt{3}}{q} \quad p_2 = 2\frac{\log(\sqrt{12})}{q} \quad p_3 = \frac{2}{q} \quad q = p_1 + p_2 + p_3 \quad 4.$$

Por otro lado si solo se tiene la media μ se puede proponer una envolvente similar usando la siguiente desigualdad:

$$f(x) \leq g_3(x) := \begin{cases} f(\mu)e\sqrt{3} & \text{si } |x - \mu| \leq \frac{1+\frac{1}{e\sqrt{3}}}{f(\mu)} \\ \frac{1}{|x-\mu|-\frac{1}{f(\mu)}} & \text{si } \frac{2}{f(\mu)} \geq |x - \mu| \geq \frac{1+\frac{1}{e\sqrt{3}}}{f(\mu)} \\ f(\mu)\exp\left(2 - \frac{|x-\mu|}{f(\mu)}\right) & \text{si } |x - \mu| \geq \frac{2}{f(\mu)} \end{cases} \quad 5.$$

el área bajo g_3 :

$$\int_{\mathbb{R}} g_3 = 6 + 2e\sqrt{3} + 2\log(\sqrt{3}) \quad 6.$$

de modo que los pesos de las componentes serán:

$$p_1 = \frac{2e\sqrt{3}+2}{q} \quad p_2 = \frac{2\log(\sqrt{12})+2}{q} \quad p_3 = \frac{2}{q} \quad q = p_1 + p_2 + p_3 \quad 7.$$

Si es costoso evaluar exactamente la densidad f entonces se pueden considerar funciones proporcionales $h_1, h_2 \propto f$ calibradas de la siguiente forma:

$$h_1(M) = 1 \quad h_2(\mu) = 1 \quad 8.$$

Esta forma resulta útil por ejemplo en el caso de la distribución log-gamma donde para la densidad $f(x) = \frac{\exp(ax-e^x)}{\gamma(a)}$, se puede definir una función proporcional:

$$h_1(x) = \frac{f(x)}{f(m)} = \frac{\frac{\exp(ax-e^x)}{\gamma(a)}}{\frac{1}{\gamma(a)}\left(\frac{a}{e}\right)^a} = \exp(a(x-m) + a - e^x) \quad 9.$$

Usando la función h_1 se evita evaluar la función gamma en cada prueba del paso de rechazo.

Si se tiene una función proporcional h_1 y además se conoce una constante $M^- \in \mathbb{R}$ tal que $f(m) = M > M^-$ entonces usando la desigualdad de la Ecuación 1 se obtiene la siguiente función envolvente para h_1

$$h_1 \leq g_4(x) = \min(1, \exp(1 - |x - m|M^-)) \quad 10.$$

El área bajo la función envolvente g_4 es proporcional a $\frac{M}{M^-}$, dependiendo de que tan ajustada sea la desigualdad $M^- < M$ se podrá muestrear usando AR de forma mas eficiente. Si solo se tiene acceso σ^2 entonces se propone una envolvente la siguiente para h_1 :

$$h_1 \leq g_6 := \min\left(1, \exp\left(1 - \left(\frac{|x - m|}{\sigma\sqrt{12}}\right)\right)\right) \quad 11.$$

El área bajo esta curva es de $4\sigma\sqrt{12}$.

Si se calibra una función $h_2 \propto f$ y se tiene acceso a la μ, σ^2 de f entonces usando la desigualdad anterior junto a la desigualdad de Johnson-Rogers [3], la cual nos da la siguiente cota inferior: $|x - m| \geq |z - \mu| - \sigma\sqrt{3}$, y el hecho de que $h_2(m) \leq h_2(\mu)e\sqrt{3}$ entonces se propone la siguiente envolvente para h_2

$$h_2 \leq g_5(x) := e\sqrt{3} \min\left(1, \exp\left(\frac{3}{2} - \frac{|x - \mu|}{\sigma\sqrt{12}}\right)\right) \quad 12.$$

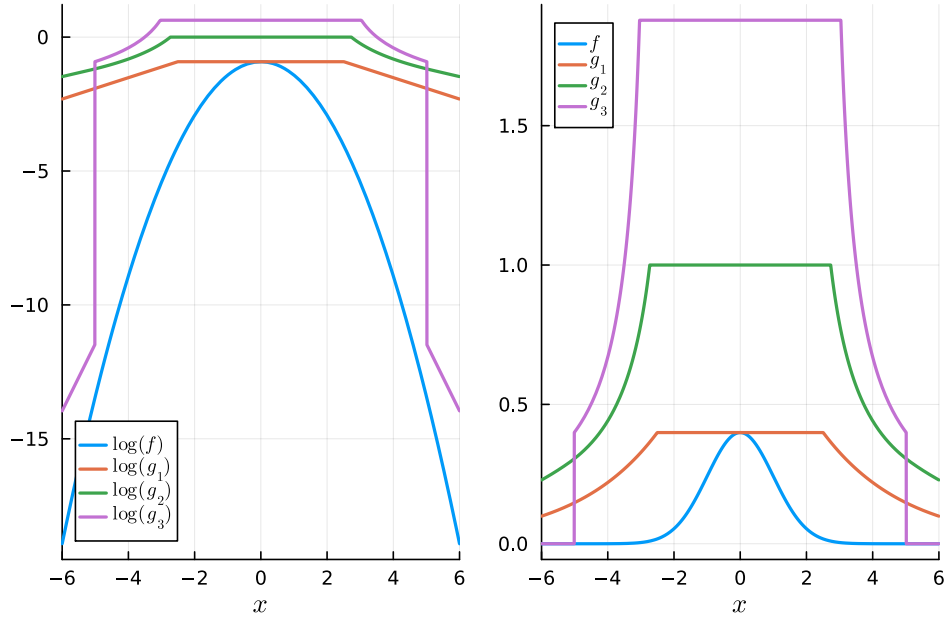


Abb. 1: Funciones envolventes definidas para para la densidad exacta f . A la izquierda se muestran las envolventes sobre $\log(f)$, mientras que a la derecha se muestran sobre f .

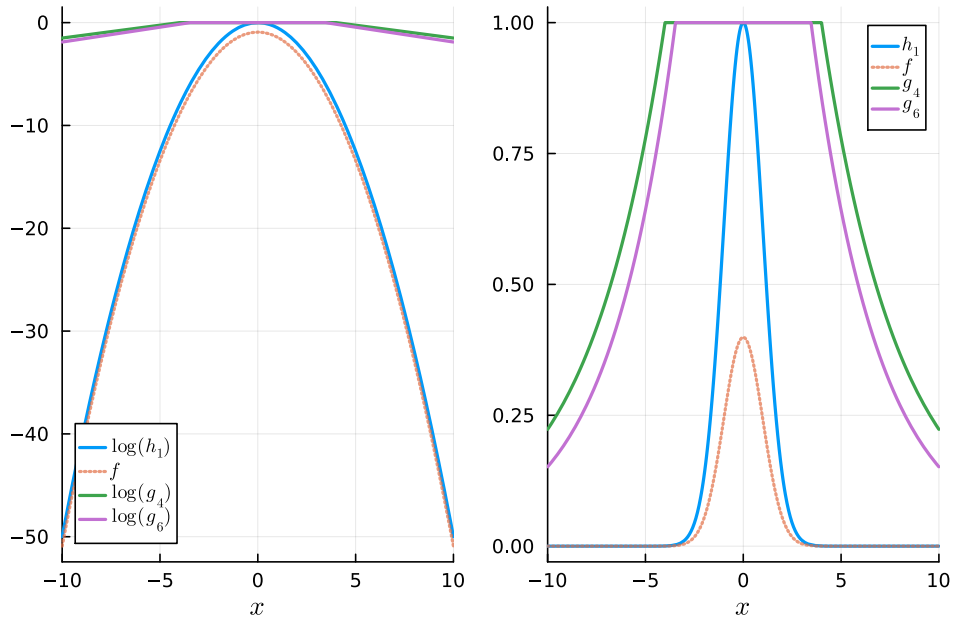


Abb. 2: Funciones envolventes definidas para la función proporcional $h_1 \propto f$. A la izquierda se muestran las envolventes sobre $\log(h_1)$, mientras que a la derecha se muestran sobre h_1 .

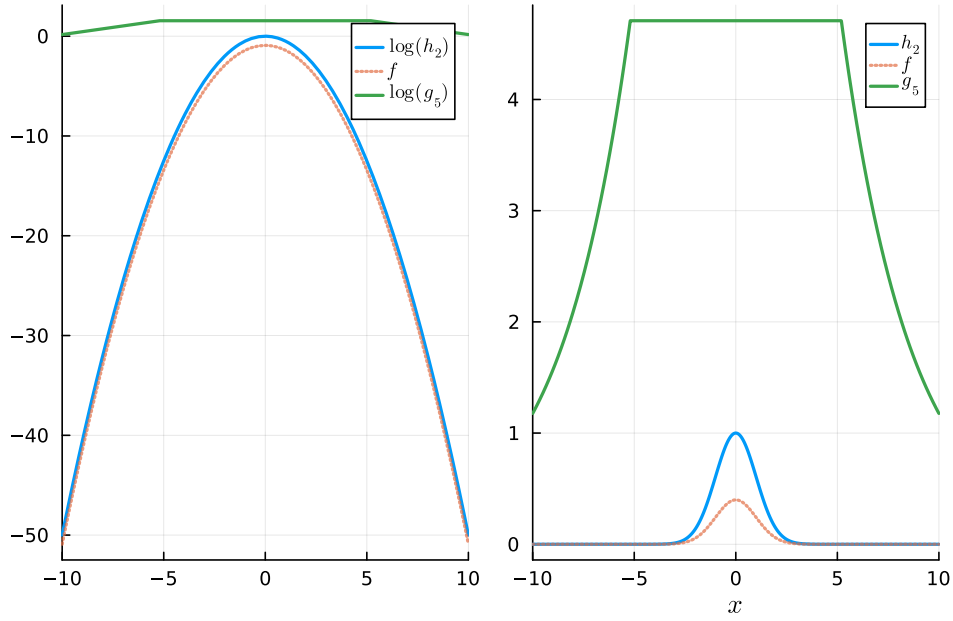


Abb. 3: Función envolvente definidas para h_2 . A la izquierda se muestran las envolventes sobre $\log(h_2)$, mientras que a la derecha se muestran sobre h_2 .

1.2 Metodos blackbox

En [2] se proponen varios algoritmos de muestreo a partir de las envolventes presentadas previamente. Para simplificar el proceso de muestreo se hacen las siguientes observaciones:

- Si la densidad es trasladada de modo que la moda se encuentra en $m = 0$ entonces cada una de las funciones envolventes se vuelve simétrica.

- Cuando se toma una muestra de la distribución dada por la envolvente normalizada, entonces basta con tomar la muestra del lado derecho y asignar el signo de forma aleatoria.
- Se puede muestrear de cada una de las componentes de las funciones envolventes usando muestras de distribuciones exponenciales y uniformes, y aplicándoles el escalado apropiado

A partir de esto se proponen varios algoritmos basados en AR usando las funciones envolventes expuestas previamente, por ejemplo usando la desigualdad de la Ecuación 1 se propone:

Algoritmo 1

```

1  $M = f(m)$ 
2  $\text{Accept} = \text{false}$ 
3 do
4   Tomar  $B \sim \text{Bernoulli}(\frac{1}{2})$ 
5   Tomar un signo  $S \in \{-1, 1\}$  de forma aleatoria
6   Tomar  $U \sim \text{Uniform}(0, 1)$ 
7   if  $B = 1$  then
8     Tomar  $V \sim \text{Uniform}(0, 1)$ 
9      $X = m + \frac{SV}{M}$ 
10     $\text{Accept} = UM \leq f(X)$ 
11  else then
12    Tomar  $E \sim \text{Exponencial}(1)$ 
13     $X = m + \frac{S(1+E)}{M}$ 
14     $\text{Accept} = UM \exp(-E) \leq f(X)$ 
15  end
16 while  $\text{accept}$  is false
17 return  $X$ 

```

El resto de funciones envolventes se pueden usar para definir algoritmos usando aceptación rechazo. Dependiendo de la función envolvente que se usa usamos la siguiente nomenclatura para los resultados de la siguiente sección:

Tab. 1: Nomenclatura de algoritmos.

Numero	Evaluación f ?	Función envolvente	Algoritmo 1	exacta	g_1
Algoritmo 2	exacta	g_2	Algoritmo 3	exacta	g_3
Algoritmo 4	h_1	g_4	Algoritmo 5	h_2	g_5
Algoritmo 6	h_1	g_6			

1.3 Envolvente construida con tangentes

Las funciones envolventes presentadas previamente dependen de tener accesos a la media o la desviación estándar o la moda de la densidad f . En esta sección consideramos una función envolvente cuya construcción solo depende de la log concavidad de f . Denotamos $\ell(x) := \log(f)$, como f es log-cóncava entonces ℓ es cóncava. Por la concavidad sabemos que toda recta tangente a ℓ se mantiene encima de la curva. Si tenemos dos puntos $x_1 < x_2$ tales que $\ell'(x_1) > 0$ $\ell'(x_2) < 0$ entonces podemos definir la siguiente función lineal a trozos:

$$\begin{aligned}
u(x) &:= \begin{cases} \ell(x_1) + (x - x_1)\ell'(x_1) & \text{si } x \in (-\infty, z) \\ \ell(x_2) + (x - x_2)\ell'(x_2) & \text{si } x \in (z, \infty) \end{cases} \\
z &:= \frac{\ell(x_2) - x_2\ell'(x_2) - \ell(x_1) + x_1\ell'(x_1)}{\ell'(x_2) - \ell'(x_1)}
\end{aligned} \tag{13}$$

Y como ℓ es cóncava se tiene que $\ell \leq u$. Además como ℓ es cóncava entonces el segmento de recta secante que conecta a los puntos x_1, x_2 estará bajo la curva de ℓ . Definimos la siguiente función lineal a trozos:

$$l(x) := \begin{cases} -\infty & \text{Si } x \notin [x_1, x_2] \\ \ell(x_1) + \frac{\ell(x_2) - \ell(x_1)}{x_2 - x_1}(x - x_1) & \text{si } x \in [x_1, x_2] \end{cases} \tag{14}$$

Al estar compuesta por el segmento de la tangente que conecta x_1 con x_2 y rectas constantes iguales a $-\infty$, tenemos la garantía de que $\ell(x) > l(x)$. A partir de estas dos funciones lineales podemos acotar la densidad f de la siguiente forma:

$$\exp(l) \leq f \leq \exp(u) := g(x) \tag{15}$$

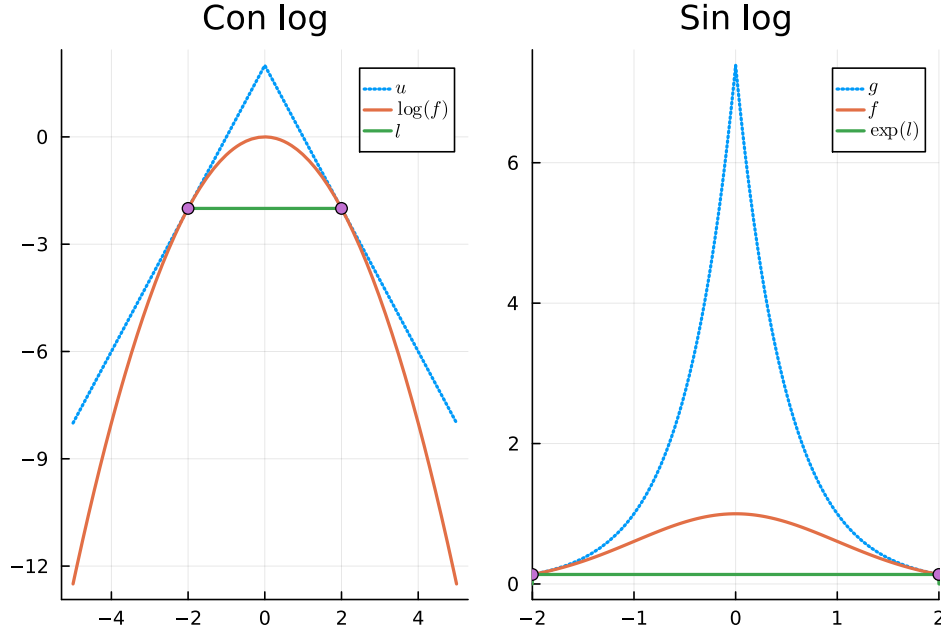


Abb. 4: Ejemplo de función envolvente u y función de squeezing l para una densidad $f \propto \text{Normal}(0, 1)$ con la construcción dada en Ecuación 13 y Ecuación 14 usando $x_1 = -2, x_2 = 2$.

El área bajo cada uno de los segmentos de g sera:

$$\begin{aligned}
A_1 &= \int_{-\infty}^{z_1} \exp(u(x)) dx = \frac{\exp(\ell'(x_1)z_1)}{\ell'(x_1)} \exp(b_1) \\
A_2 &= \int_{z_1}^{\infty} \exp(u(x)) dx = \frac{\exp(\ell'(x_2)z_1)}{\ell'(x_2)} \exp(b_1) \\
A &= A_1 + A_2
\end{aligned} \tag{16}$$

Definimos las CDF de los segmentos de $\frac{g}{A}$ de la siguiente forma:

$$\begin{aligned}
G_1(x) &= \frac{\int_{-\infty}^x g(x)}{A_1} = 1 - \exp(\ell'(x_1)(x - z_1)) \\
G_2(x) &= \frac{\int_{z_1}^x g(x)}{A_2} = 1 - \exp(\ell'(x_2)(x - z_1))
\end{aligned} \tag{17}$$

Invertimos cada una de las funciones:

$$\begin{aligned}
G_1^{-1}(u) &= z_1 + \frac{\log(1-u)}{\ell'(x_1)} \\
G_2^{-1}(u) &= z_1 + \frac{\log(1-u)}{\ell'(x_2)}
\end{aligned} \tag{18}$$

De modo que si queremos tomar una muestra $x \sim \frac{g}{A}$ basta con seleccionar una una de las componentes con probabilidad $p_1 = \frac{A_1}{A}$, $p_2 = 1 - p_1$ y tomar $U \sim \text{Uniform}(0, 1)$ entonces $G_i^{-1}(U) = x$.

Usando la construcción de la función envolvente a partir de las rectas tangentes y de las secantes de ℓ se propone el método de aceptación rechazo adaptativo (ARS) en [4] el cual consiste de lo siguiente

Se define un conjunto de abscisas $T = x_1, x_2, \dots, x_k$ con la condicon de que $h'(x_1) > 0$, $h'(x_k) < 0$ y $x_1 < x_2 < \dots < x_k$. A partir de este conjunto se extiende la construccion de la funcion envolvente u con respecto al conjunto T . Primero se calculan los puntos de intersección de las rectas tangentes a cada punto de T :

$$z_j = \frac{\ell(x_{j+1}) - x_{j+1}\ell'(x_{j+1}) - \ell(x_j) + x_j\ell'(x_j)}{\ell'(x_{j+1}) - \ell'(x_j)} \quad \text{para } j = 1, 2, \dots, k-1 \tag{19}$$

Si el dominio de f es \mathbb{R} entonces asignamos $z_0 = -\infty$ $z_k = \infty$, si el dominio esta acotado entonces $z_0 = \min(D)$ $z_k = \max(D)$ A partir de estos puntos se extiende la construccion de u en la Ecuación 13 para T :

$$u_T(x) := \begin{cases} \ell(x_1) + (x - x_1)\ell'(x_1) & \text{si } x \in (z_0, z_1) \\ \ell(x_2) + (x - x_2)\ell'(x_2) & \text{si } x \in (z_1, z_2) \\ \vdots & \\ \ell(x_k) + (x - x_k)\ell'(x_k) & \text{si } x \in (z_{k-1}, z_k) \end{cases} \tag{20}$$

Extendemos también la construcción de la función de squeezing l para el conjunto T de la siguiente forma

$$\begin{aligned}
l_T(x) &= \begin{cases} h(x_1) + m_j(x - x_1) & \text{si } x \in [x_1, x_2] \\ h(x_2) + m_j(x - x_2) & \text{si } x \in [x_2, x_3] \\ \vdots & \\ h(x_{k-1}) + m_{k-1}(x - x_{k-1}) & \text{si } x \in [x_{k-1}, x_k] \\ -\infty & \text{si } x \notin [x_1, x_k] \end{cases} \\
m_j &= \frac{h(x_{j+1}) - h(x_j)}{x_{j+1} - x_j} \quad \text{para } j = 1, 2, \dots, k-1
\end{aligned} \tag{21}$$

¹Este caso solo se da cuando $x = m$, en la construcción de dos puntos se supuso que ninguna de las abscisas era la ubicación de la moda

Para calcular el area debajo de cada una de las componentes de $g_T = \exp(u_T)$ usamos la misma formula que con el caso de dos abscisas cuando $\ell'(x) \neq 0$ y agregamos el caso de cuando $\ell'(x) = 0$ ¹ de la siguiente forma

$$A_j = \int_{z_j}^{z_{j+1}} g_{T(y)} dy = \begin{cases} (z_{j+1} - z_j) \exp(B_j) & \text{si } \ell'(x_j) = 0 \\ \frac{\exp(\ell'(x_j)z_{j+1}) - \exp(\ell'(x_j)z_j)}{\ell'(x_j)} & \text{si } \ell'(x_j) \neq 0 \end{cases} \quad 22.$$

$$B_j = \ell(x_j) - \ell'(x_j)x_j$$

$$A = \sum_j A_j$$

los pesos de cada componente de g_T estarán dados por:

$$p_j = \frac{\int_{z_j}^{z_{j+1}} g_{T(y)} dy}{A} \quad 23.$$

La CDF de cada componente de g_T esta dada por

$$G_{j(x)} = \frac{\int_{z_j}^x g_{T(y)} dy}{\int_{z_j}^{z_{j+1}} g_{T(y)} dy} = \begin{cases} \frac{\exp(x\ell'(x_j)) - \exp(z_j\ell'(x_j))}{\exp(z_{j+1}\ell'(x_j))} - \exp(z_j\ell'(x_j)) & \text{si } \ell'(x_j) \neq 0 \\ \frac{x - z_j}{z_{j+1} - z_j} & \text{si } \ell'(x_j) = 0 \end{cases} \quad 24.$$

Invirtiéndola tenemos:

$$G_j^{-1}(u) = \begin{cases} \frac{1}{\ell'(x_j)} \log(u \exp(\ell'(x_j)z_{j+1}) + (1 - u) \exp(\ell'(x_j)z_j)) & \text{si } \ell'(z_j) \neq 0 \\ z_j + u(z_{j+1} - z_j) & \text{si } \ell'(z_j) = 0 \end{cases} \quad 25.$$

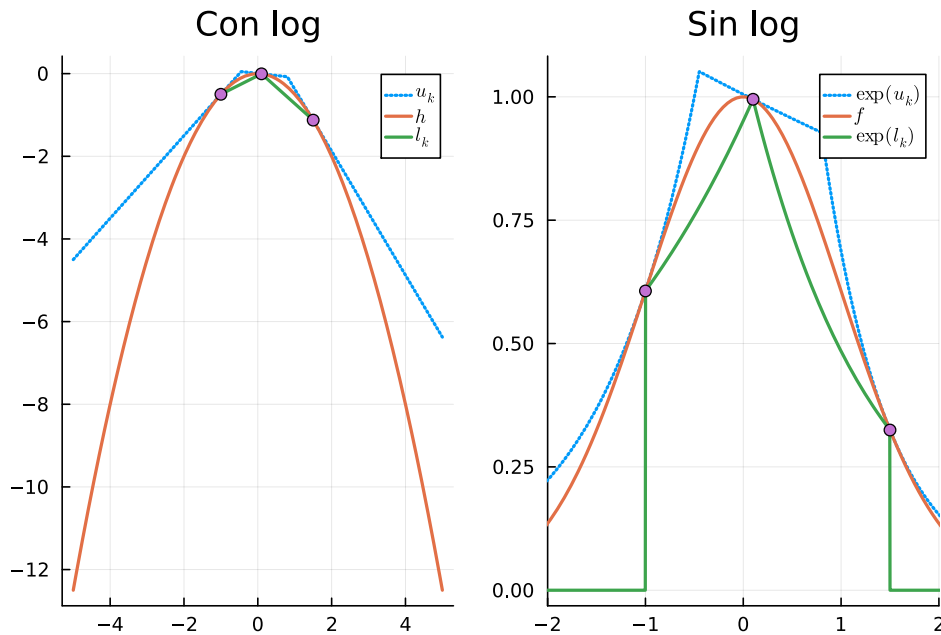


Abb. 5: Ejemplo de función envolvente u_T y función de squeezing l_T para una densidad $f \propto \text{Normal}(0, 1)$ usando $T = (-1, 0.1, 1)$.

De modo que proponemos el siguiente algoritmo para muestrear de $\frac{g_T}{A}$

Muestreo de g_T

```

1 Tomar  $j$  tal que  $\mathbb{P}(j = i) = p_i$ 
2 Tomar  $U \sim \text{Uniform}(0, 1)$ 
3 if  $\ell'(x_j) == 0$ 
4   | return  $\frac{1}{\ell'(x_j)} \log(U \exp(\ell'(x_j)z_{j+1}) + (1 - U) \exp(\ell'(x_j)z_j))$ 
5 else
6   | return  $z_j + U(z_{j+1} - z(j))$ 
7 end
```

Otro elemento en el que difiere ARS de método de rechazo convencional es que se considera una prueba de *squeezing* para esto se utiliza el cociente $\frac{l_K}{g_K}$, cuando las muestras son rechazadas por la prueba de *squeezing* entonces se añaden a T_k , de modo que $T_{k+1} = T_k \cup \{x\}$, este nuevo conjunto de abscisas induce a nuevas funciones l_{K+1}, u_{K+1} , entre mas puntos tenga el conjunto de abscisas entonces g_K estará mas ajustada sobre f lo cual reduce las muestras rechazadas.

```

1 function ARS( $N, T_K$ )
2 Definir  $g_T, s_T$ 
3 Inicializar  $X = \{\}$ 
4 while  $|X| < N$ 
5   | Tomar  $x \sim \frac{g}{A}$ 
6   |  $U \sim \text{Uniform}(0, 1)$ 
7   | if  $U \leq \frac{s_T(x)}{g_T(x)}$  then
8     | Agregar  $x$  a  $X$ 
9   | else then
10    | if  $U \leq \frac{s_T(x)}{g_T(x)}$  then
11      | Agregar  $x$  a  $X$ 
12    | end
13    | Actualizar el conjunto de abscisas  $T$  añadiendo  $x$ 
14    | Re definir  $g_T, s_T$ 
15  | end
16 end
17 return  $X$ 
```

2 Implementaciones

Se realizo una implementación de cada uno de los algoritmos descritos en Tab. 1 y de ARS en Julia. Para probar los algoritmos de la Tab. 1 utilizamos una distribución normal f con parámetros $\mu = 2, \sigma^2 = 3$. Usando las funciones de benchmarking de Julia calculamos el costo en tiempo y memoria (alocaciones de memoria) de generar una sola muestra de f . Para comparar se implementaron funciones para tomar una muestra usando el método `rand()` del paquete de **Distributions.jl**[5], se registro el tiempo promedio y uso de memoria. En la Tab. 2 podemos ver que todos los algoritmos de la Tab. 1 requieren mucho mas tiempo que el algoritmo base, esta diferencia es menor en los algoritmos **1** y **2**, pero aun así no deja de ser significativamente mas lento que el algoritmo original de muestreo

	Alocaciones	Tiempo promedio
Algoritmo base	0	4.401 ns \pm 0.518 ns
Algoritmo 1	21	543.539 ns \pm 957.936 ns
Algoritmo 2	33	1.155 μ s \pm 1.253 μ s
Algoritmo 3	45	2.217 μ s \pm 5.941 μ s
Algoritmo 4	28	768.602 ns \pm 654.707 ns
Algoritmo 5	45	1.098 μ s \pm 812.810 ns
Algoritmo 6	39	1.077 μ s \pm 1.162 μ s

Para probar la implementación del algoritmo ARS se considero la misma densidad f , solo que en este caso para la prueba se generaron $N = 10000$ muestras, inicializando el conjunto de abscisas como $T_k = \{-1, 0.5, 1\}$. Los resultados en la tabla Tab. 3 muestran que el tiempo y las alocaciones son altas para (ARS), al igual que con el Algoritmo 1 Otra comparación que se realizo entre el algoritmo 1 y (ARS) fue el numero de muestras que se rechazan cuando se intenta tomar una muestra de tamaño $N = 10000$, en la Tab. 4 se puede ver que (ARS) en promedio rechaza muy pocas muestras candidatas. Esto se debe a que el conjunto final de abscisas T generado por (ARS) tiene en promedio un tamaño grande, lo cual resulta en una función envolvente g_T extremadamente ajustada a f , lo cual parece ser también la causa de que requiera tiempo extra, ya que al agregar múltiples puntos a T durante el *loop* de muestreo de (ARS) entonces aumenta la complejidad de tomar muestras de $\frac{g_T}{A}$.

	Alocaciones	Tiempo promedio
Algoritmo base	0	3.245 μ s \pm 19.783 μ s
ARS	21	9.973 ms \pm 2.190 ms
Algoritmo 1	21	5.775 ms \pm 1.117 ms

	# Rechazos (promedio)	# Abscisas (promedio)
Algoritmo base	NA	NA
ARS	5	100
Algoritmo 1	5964	NA

Las diferencias tan marcadas entre el desempeño de los algoritmos implementados en este proyecto contra el algoritmo base usado en Julia también es importante considerar que el algoritmo de muestreo que se utiliza es **Xorshiro256** el cual destaca por su eficiencia tanto en velocidad como en memoria.

3 Discusión

Los algoritmos de caja negra y adaptativo, se aprovechan de la log concavidad para poder realizar simulación sobre f aun cuando no tenemos acceso a esta función directamente o a la locación de la moda. Esto los pone en ventaja sobre aceptación-rechazo estándar, ya que si se tratara de simular de una distribución f para la cual se desconoce la locación de su moda m entonces se tendría que hacer un paso previo de optimización, lo cual es costoso computacional mente.

El algoritmo adaptativo ademas nos permite ir mejorando la función envolvente que se utiliza para tomar muestras candidatas en el paso de aceptación rechazo, por lo que cada vez que simulamos una variable X y la rechazamos estamos mejorando la capacidad de muestreo del algoritmo, un ejemplo de esto podemos ver en la Abb. 6 donde tomamos la densidad f , y vemos como el cociente del área de la

envolvente entre la constante de normalización de f reduce entre mas abscisas se usan para construir la envolvente.

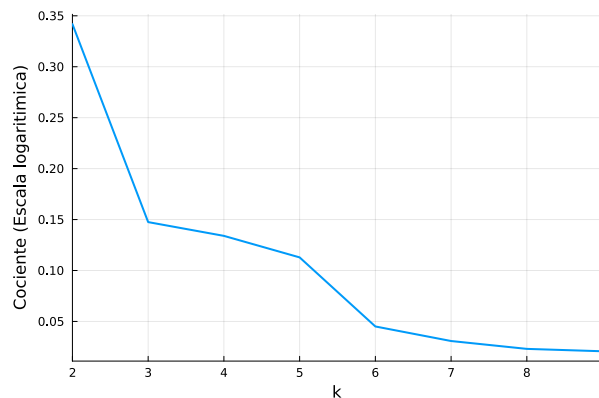


Abb. 6: Relación entre el cociente del área de la envolvente entre la constante de normalización de f . Se tomaron conjuntos de abscisas con tamaño k , se construyo la función envolvente a partir de estas abscisas y se calculo su área. Se calculo el cociente entre el área de la envolvente y la constante de normalización de f . El comportamiento decreciente indica que entre mas grande sea el conjunto de el conjunto de abscisas entonces menor sera la probabilidad de generar una muestra de g normalizada que tenga que ser rechazada.

La versatilidad de estos algoritmos, especialmente en el caso de ARS donde solo se requiere la su posición de que f es log cóncava, da una clara ventaja que compensa su alto costo computacional que tienen.

Unos anos despues del articulo original donde se propuso (ARS) Gilks y Walters propusieron una version que no requiere la derivada de f [6]. Tambien se ha propuesto una generalizacion para densidades que no son log convacavs [7], la cual requiere de consideraciones extra. Tambien se ha diseñado un metodo para optimizar la construcción del conjunto de abscisas inicial para aumentar la eficiencia del muestreo [8].

Bibliografía

- [1] L. Devroye, *Non-Uniform Random Variate Generation*. en SpringerLink Bücher. New York, NY: Springer, 1986. doi: 10.1007/978-1-4613-8643-8.
- [2] L. Devroye, «Inequalities and simulation methods for univariate log-concave densities», *Communications in Statistics-Simulation and Computation*, pp. 1-21, 2025.
- [3] N. L. Johnson y C. A. Rogers, «The Moment Problem for Unimodal Distributions», *The Annals of Mathematical Statistics*, vol. 22, n.º 3, pp. 433-439, sep. 1951, doi: 10.1214/aoms/1177729590.
- [4] W. R. Gilks y P. Wild, «Adaptive Rejection Sampling for Gibbs Sampling», *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 41, n.º 2, pp. 337-348, 1992, doi: 10.2307/2347565.
- [5] D. Lin *et al.*, «JuliaStats/Distributions.Jl: V0.25.119». abril de 2025. doi: 10.5281/ZENODO.2647458.
- [6] W. R. Gilks, «Derivative-free adaptive rejection sampling for Gibbs sampling», *Bayesian statistics*, vol. 4, n.º 2, pp. 641-649, 1992.
- [7] L. Martino y J. M\iguez, «A generalization of the adaptive rejection sampling algorithm», *Statistics and Computing*, vol. 21, pp. 633-647, 2011.
- [8] J. James, «Automated generation of initial points for adaptive rejection sampling of log-concave distributions», *Statistics and Computing*, vol. 34, n.º 3, p. 109, 2024.