

Métodos de muestreo para densidades log cóncavas

Javier Roberto Rubalcava Cortes

Jan 16, 2025

Agenda

1. Simulación usando aceptación rechazo
2. Desigualdades para densidades LC
3. Algoritmos de caja negra
4. Adaptive rejection sampling
5. Implementaciones
6. Puntos finales

Simulación usando aceptación rechazo

- El metodo de aceptación rechazo nos permite simular muestras de una densidad f usando una densidad envolvente g que satisface en todo el domino $f(x) \leq Kg(x)$. Se generan muestras candidatas usando g y se aceptan si:

$$U \leq \frac{f(x')}{g(x')} \quad \text{donde} \quad U \sim \text{Uniform}([0, 1]), x' \sim \frac{g}{K} \quad 1.$$

- La eficiencia de AR depende de la elección de g , una buena g cumple:
 - Queremos que $\frac{g}{K}$ sea fácil de muestrear
 - Queremos que g pueda ser evaluada de forma eficiente
 - Queremos que haya la mínima área entre f y $K \cdot g$
- NO es fácil obtener una densidad g que cumpla todos los puntos de arriba

Desigualdades para densidades LC

Desigualdades para la densidad exacta

- Suponemos que la densidad f es log-cóncava con dominio $D = \mathbb{R}$.
- La primera desigualdad que se expone en [1] solo requiere de la localización de la moda m :

$$f(x) \leq g_1(x) := M \min(1, \exp(1 - |x - m|M)) \quad \text{donde } f(m) = M \quad 2.$$

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

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

- Si solo tenemos acceso a la media μ podemos tomar la siguiente desigualdad:

$$f(x) \leq g_3(x) := \begin{cases} f(\mu)e\sqrt{3} & \text{si } |x - \mu| \leq \frac{1 + \frac{1}{e\sqrt{3}}}{f(\mu)} \\ \frac{1}{|x - \mu| - \frac{1}{f(\mu)}} & \text{si } \frac{2}{f(\mu)} \geq |x - \mu| \geq \frac{1 + \frac{1}{e\sqrt{3}}}{f(\mu)} \\ f(\mu) \exp\left(2 - \frac{|x - \mu|}{f(\mu)}\right) & \text{si } |x - \mu| \geq \frac{2}{f(\mu)} \end{cases} \quad 4.$$

- Las ultimas dos desigualdades son resultado de la primera y de las siguientes desigualdades para densidades LC

$$\frac{1}{e\sqrt{3}} \leq \sigma f(\mu) \leq 1 \quad \frac{1}{\sqrt{12}} \leq \sigma f(m) \leq 1 \quad 5.$$

- Las areas de las funciones envloventes g_1, g_2, g_3 son:

$$\int_{\mathbb{R}} g_1(t) dt = 4$$

$$\int_{\mathbb{R}} g_3(t) dt = 2(1 + \sqrt{3}) + 2(\log(\sqrt{12})) + \frac{\sqrt{12}}{\sqrt{3}} \approx 9.94 \quad 6.$$

$$\int_{\mathbb{R}} g_3(t) dt = 6 + 2e\sqrt{3} + 2\log(\sqrt{3}) \approx 16$$

- Tienen area constante sin importar como sea f

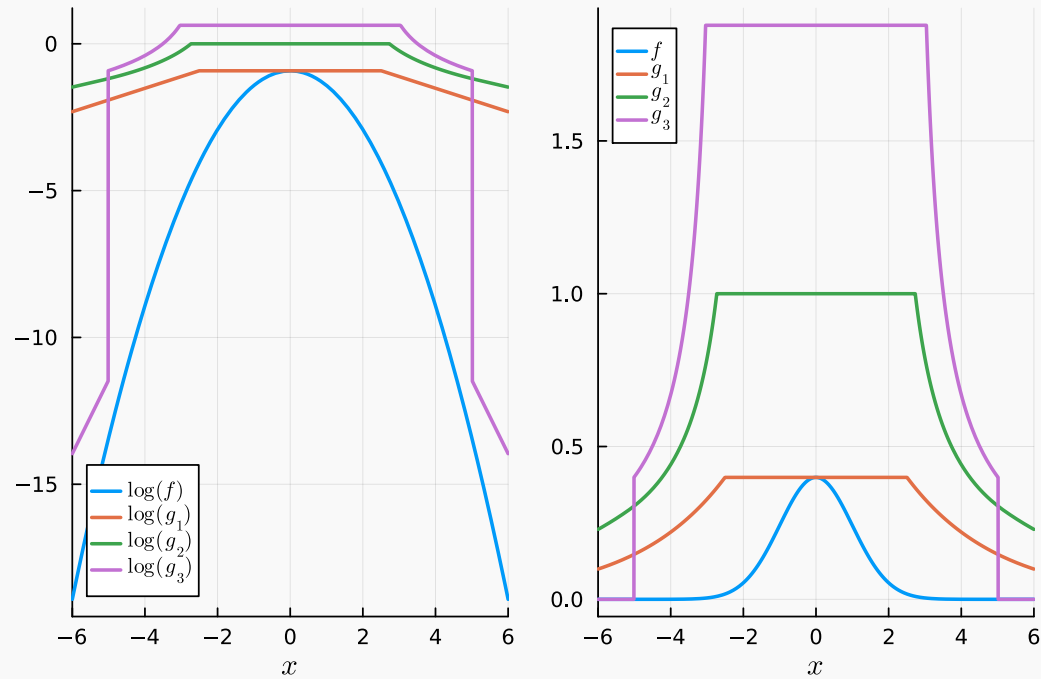


Figura 1: Funciones envolventes definidas para para la densidad exacta f .

Desigualdades para funciones proporcionales

- Tomamos una función $h_1 \propto f$ definida como $h_1(x) = \frac{f(x)}{f(m)}$ de modo que $h_1(m) = 1$
- Si se conoce una constante $M^- \in \mathbb{R}$ que cumple $f(m) = M > M^-$ entonces usando la desigualdad de la Ecuación 2 se obtiene la siguiente función envolvente para h_1

$$h_1 \leq g_4(x) = \min(1, \exp(1 - |x - m|M^-)) \quad 7.$$

El área bajo la función envolvente g_4 es proporcional a $\frac{M}{M^-}$.

- Si solo se tiene acceso σ^2 entonces se propone una envolvente la siguiente para h_1 :

$$h_1 \leq g_6 := \min \left(1, \exp \left(1 - \left(\frac{|x - m|}{\sigma \sqrt{12}} \right) \right) \right) \quad 8.$$

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

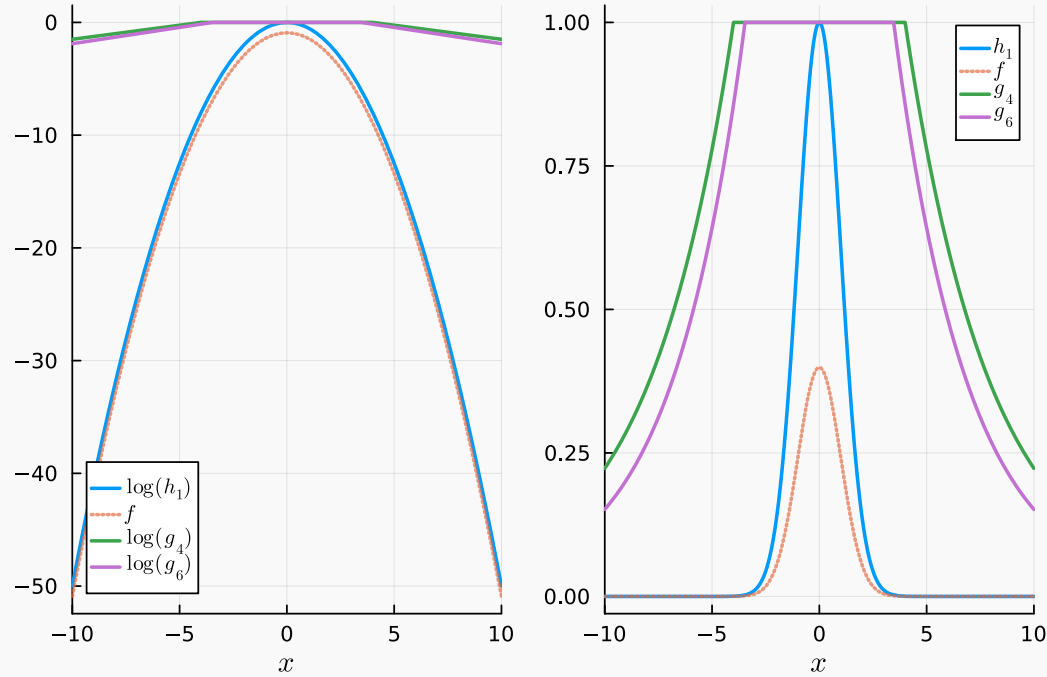


Figura 2: Funciones envolventes definidas para la función proporcional $h_1 \propto f$.

- Si se calibra una función $h_2 \propto f$ como $h_2(x) = \frac{f(x)}{f(\mu)}$ y se tiene acceso a la μ, σ^2 de f entonces usando la 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 :

$$h_2 \leq g_5(x) := e\sqrt{3} \min \left(1, \exp \left(\frac{3}{2} - \frac{|x - \mu|}{\sigma\sqrt{12}} \right) \right) \quad 9.$$

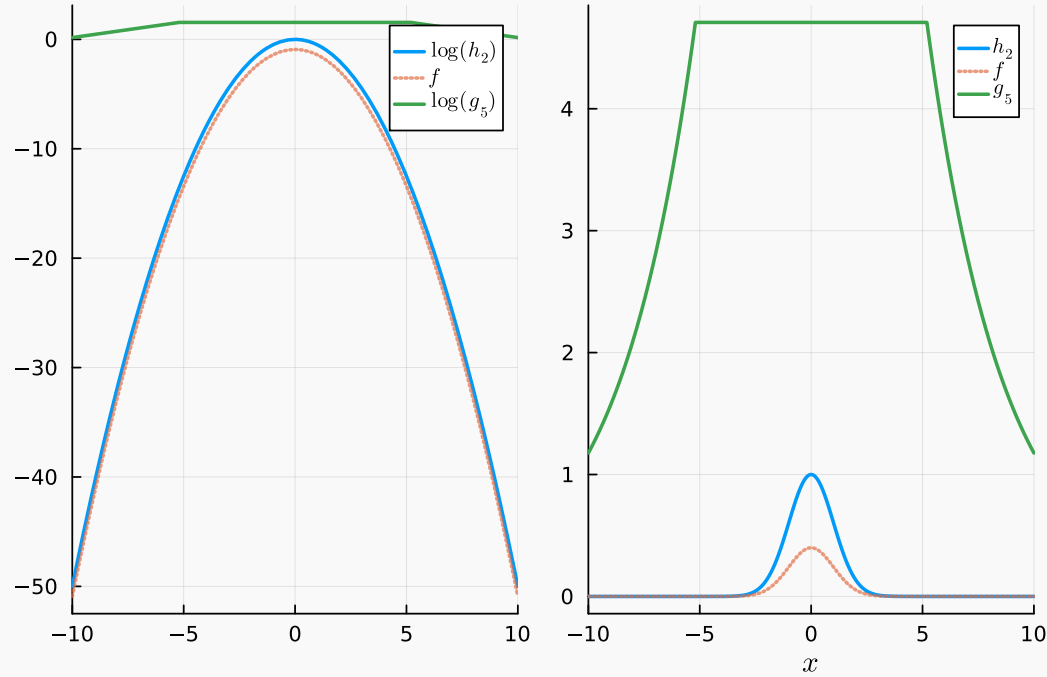


Figura 3: Funciones envolventes definidas para la función proporcional $h_2 \propto f$.

Algoritmos de caja negra

- Usando las funciones envolventes expuestas previamente se proponen varios algoritmos basados en AR, por ejemplo usando la desigualdad de la Ecuación 2 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  $U \sim \text{Uniform}(0, 1)$ 
```

Algoritmo 1

```
7  | if  $B = 1$  then
8  |   Tomar  $V \sim \text{Uniform}(0, 1)$ 
9  |    $X = m + \frac{SV}{M}$ 
10 |   Accept =  $UM \leq f(X)$ 
11 | else then
12 |   Tomar  $E \sim \text{Exponencial}(1)$ 
13 |    $X = m + \frac{S(1+E)}{M}$ 
14 |   Accept =  $UM \exp(-E) \leq f(X)$ 
15 | end
```

Algoritmo 1

16 while accept is false
17 return X

Adaptive rejection sampling

Envolvente construida con tangentes

Las funciones envolventes presentadas previamente dependen de tener accesos a la media o la varianza 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:

$$\begin{aligned}
 u(x) &:= \begin{cases} \ell(x_1) + (x - x_1)\ell'(x_1) & \text{si } x \in (-\infty, z) \\ \ell(x_2) + (x - x_2)\ell'(x_2) & \text{si } x \in (z, \infty) \end{cases} \\
 z &:= \frac{\ell(x_2) - x_2\ell'(x_2) - \ell(x_1) + x_1\ell'(x_1)}{\ell'(x_2) - \ell'(x_1)}
 \end{aligned}
 \tag{10}$$

Y como ℓ es cóncava se tiene que $\ell \leq u$.

- Por la concavidad de ℓ 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:

$$l(x) := \begin{cases} -\infty & \text{Si } x \notin [x_1, x_2] \\ \ell(x_1) + \frac{\ell(x_2) - \ell(x_1)}{x_2 - x_1}(x - x_1) & \text{si } x \in [x_1, x_2] \end{cases}
 \tag{11}$$

- La concavidad de ℓ garantiza que $\ell(x) > l(x)$.
- A partir de estas dos funciones lineales podemos acotar la densidad f de la siguiente forma:

$$\exp(l) \leq f \leq \exp(u) := g(x) \quad 12.$$

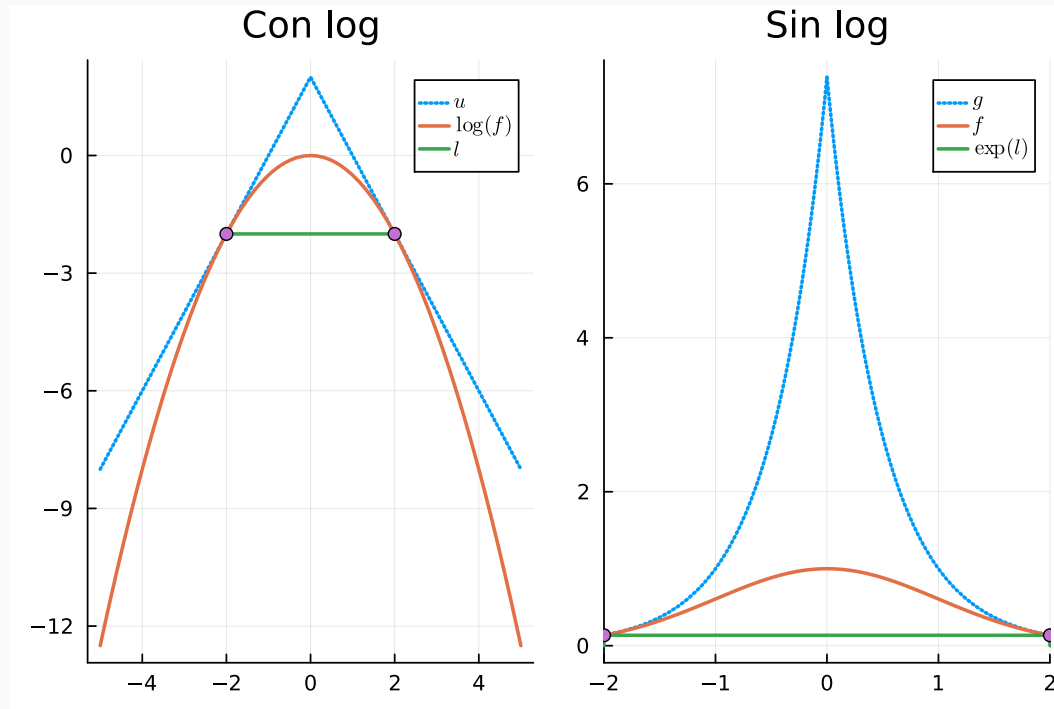


Figura 4: Ejemplo de función envolvente u y función de squeezing l para una densidad $f \propto \text{Normal}(0, 1)$ con la construcción dada en Ecuación 10 y Ecuación 11 usando $x_1 = -2, x_2 = 2$.

- El área bajo cada uno de los segmentos de la función envolvente g sera:

$$\begin{aligned}
 A_1 &= \int_{-\infty}^{z_1} \exp(u(x)) dx = \frac{\exp(\ell'(x_1)z_1)}{\ell'(x_1)} \exp(b_1) \\
 A_2 &= \int_{z_1}^{\infty} \exp(u(x)) dx = \frac{\exp(\ell'(x_1)z_1)}{\ell'(x_1)} \exp(b_1) \\
 A &= A_1 + A_2
 \end{aligned}
 \tag{13.}$$

- Definimos las distribuciones de los segmentos de $\frac{g}{A}$:

$$G_1(x) = \frac{\int_{-\infty}^x g(x)}{A_1} = 1 - \exp(\ell'(x_1)(x - z_1))$$

$$G_2(x) = \frac{\int_{z_1}^x g(x)}{A_2} = 1 - \exp(\ell'(x_2)(x - z_1))$$

14.

Invertimos cada una de las funciones:

$$\begin{aligned} G_1^{-1}(u) &= z_1 + \frac{\log(1-u)}{\ell'(x_1)} \\ G_2^{-1}(u) &= z_1 + \frac{\log(1-u)}{\ell'(x_2)} \end{aligned} \tag{15.}$$

- 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 $G_i^{-1}(U) = x$.

Generalización de la construcción

- 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 [2] el cual consiste de lo siguiente
- Definimos el conjunto de abscisas $T = x_1, x_2, \dots, x_k$ con la condición de $h'(x_1) > 0$, $h'(x_k) < 0$ y $x_1 < x_2 < \dots < x_k$.
- A partir de este conjunto se extiende la construcción de la función envolvente u con respecto al conjunto T .

- Primero se calculan los puntos de intersección de las rectas tangentes inducidas por cada punto de T :

$$z_j = \frac{\ell(x_{j+1}) - x_{j+1}\ell'(x_{j+1}) - \ell(x_j) + x_j\ell'(x_j)}{\ell'(x_{j+1}) - \ell'(x_j)} \quad \text{para } j = 1, 2, \dots, k-1 \quad 16.$$

- 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 10 para T :

$$u_T(x) := \begin{cases} \ell(x_1) + (x - x_1)\ell'(x_1) & \text{si } x \in (z_0, z_1) \\ \ell(x_2) + (x - x_2)\ell'(x_2) & \text{si } x \in (z_1, z_2) \\ \vdots & \\ \ell(x_k) + (x - x_k)\ell'(x_k) & \text{si } x \in (z_{k-1}, z_k) \end{cases} \quad 17.$$

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

$$l_T(x) = \begin{cases} h(x_1) + m_j(x - x_1) & \text{si } x \in [x_1, x_2] \\ h(x_2) + m_j(x - x_2) & \text{si } x \in [x_2, x_3] \\ \vdots \\ h(x_{k-1}) + m_{k-1}(x - x_{k-1}) & \text{si } x \in [x_{k-1}, x_k] \\ -\infty & \text{si } x \notin [x_1, x_k] \end{cases} \quad 18.$$

$$m_j = \frac{h(x_{j+1}) - h(x_j)}{x_{j+1} - x_j} \quad \text{para } j = 1, 2, \dots, k - 1$$

- Para calcular el área 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

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

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

19.

$$A = \sum_j A_j$$

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

los pesos de cada componente de g_T estarán dados por:

$$p_j = \frac{\int_{z_j}^{z_{j+1}} g_{T(y)} \, dy}{A} \quad 20.$$

La CDF de cada componente de g_T esta dada por

$$G_{j(x)} = \frac{\int_{z_j}^x g_{T(y)} \, dy}{\int_{z_j}^{z_{j+1}} g_{T(y)} \, dy} = \begin{cases} \frac{\exp(x\ell'(x_j)) - \exp(z_j\ell'(x_j))}{\exp(z_{j+1}\ell'(x_j))} - \exp(z_j\ell'(x_j)) & \text{si } \ell'(x_j) \neq 0 \\ \frac{x - z_j}{z_{j+1} - z(j)} & \text{si } \ell'(z_j) = 0 \end{cases} \quad 21.$$

Invirtiéndola tenemos:

$$G_j^{-1}(u) = \begin{cases} \frac{1}{\ell'(x_j)} \log(u \exp(\ell'(x_j)z_{j+1}) + (1 - u) \exp(\ell'(x_j)z_j)) & \text{si } \ell'(z_j) \neq 0 \\ z_j + u(z_{j+z} - z_j) & \text{si } \ell'(z_j) = 0 \end{cases} \quad 22.$$

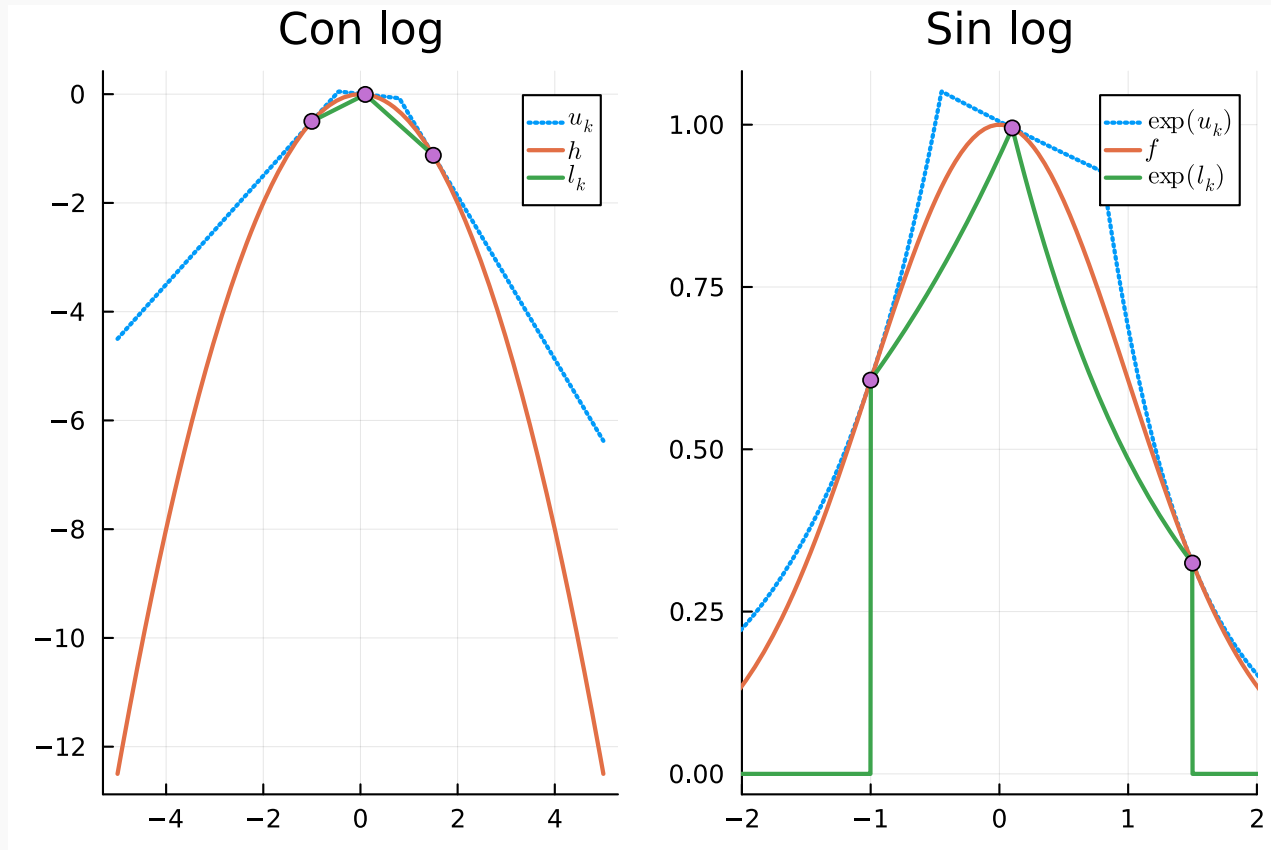


Figura 5: Ejemplo de función envolvente u_T y función de squeezing l_T para una densidad $f \propto \text{Normal}(0, 1)$ usando $T = (-1, 0.1, 1)$.

Definimos 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
```

- Cuando tomamos un candidato $x' \sim \frac{g}{A}$ se realiza una prueba de squeezing entre s_T y g_T antes de hacer la prueba de rechazo entre f y g_T .
-

```
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  $X$ 
```

Implementaciones

Benchmarking

- Se realizó una implementación de cada uno de los algoritmos descritos en [1] y de ARS en Julia.
- Para probar los algoritmos de caja negra 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 usó el método `rand()` del paquete de `Distributions.jl`[3]. Se registró el tiempo promedio y uso de memoria.
- En la Tabla 1 podemos ver que todos los algoritmos de caja negra requieren mucho más 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

- 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 [4] el cual destaca por su eficiencia tanto en velocidad como en memoria.

- 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 Tabla 2 muestran que el tiempo y las alocaiones son altas para (ARS), al igual que con el Algoritmo 1

	Alocaciones	Tiempo promedio
Algoritmo base	0	$3.245 \mu\text{s} \pm 19.783 \mu\text{s}$
ARS	21	$9.973 \text{ ms} \pm 2.190 \text{ ms}$
Algoritmo 1	21	$5.775 \text{ ms} \pm 1.117 \text{ ms}$

Comparación del numero de rechazos en ARS

- Otra comparación que se realizo entre el algoritmo 1 y (ARS) fue el numero de muestras que se rechazan cuando se simula una muestra de tamaño $N = 10000$.
- En la Tabla 3 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 $\frac{g_T}{A}$.

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

Puntos finales



- ARS NO es optimo ya que en la mayoría de los casos es posible diseñar un algoritmo especializado.
- La ventaja principal de ARS es su universalidad. Solo requerimos la forma cerrada de f y su derivada para poder simular muestras
- Se han propuesto generalización de ARS que no requieren de la derivada de f [5] y que pueden ser usados en densidades que no son log convexas [6] .
Tambien se han diseñado métodos para optimizar la construcción del conjunto de abscisas inicial para aumentar la eficiencia del muestreo [7] .

- [1] L. Devroye, «Inequalities and simulation methods for univariate log-concave densities», *Communications in Statistics-Simulation and Computation*, pp. 1-21, 2025.
- [2] 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.
- [3] D. Lin *et al.*, «JuliaStats/Distributions.Jl: V0.25.119». abril de 2025. doi: 10.5281/ZENODO.2647458.
- [4] G. Marsaglia, «Xorshift RNGs», *Journal of Statistical Software*, vol. 8, pp. 1-6, jul. 2003, doi: 10.18637/jss.v008.i14.

- [5] W. R. Gilks, «Derivative-free adaptive rejection sampling for Gibbs sampling», *Bayesian statistics*, vol. 4, n.º 2, pp. 641-649, 1992.
- [6] L. Martino y J. M\iguez, «A generalization of the adaptive rejection sampling algorithm», *Statistics and Computing*, vol. 21, pp. 633-647, 2011.
- [7] 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.