

# Inferencia Bayesiana en modelos de ecuaciones diferenciales: Un enfoque computacional

Erick Iván Guerrero Flores

Universidad Autónoma Metropolitana, Unidad Iztapalapa  
Maestría en Ciencias (Matemáticas Aplicadas e Industriales)  
Departamento de Matemáticas

Un travesía por la ciencia de datos.  
Abril, 2021

Asesores:

Dr. Gabriel Núñez Antonio

Dr. José Héctor Morales Bárcenas

# Contenido

- 1 Objetivo General
- 2 Introducción
- 3 Métodos Numéricos y Estocásticos
- 4 Bibliografía

# Objetivo General

Análisis y comparación de algunos métodos numéricos MCMC que se emplean en la inferencia bayesiana, particularmente en el caso donde las soluciones a los sistemas de ecuaciones diferenciales no son analíticas o donde no se cuenta con una verosimilitud que describa la relación entre los parámetros del modelo y la solución del sistema.

Inferencia guiada por datos aplicada a soluciones numéricas de sistemas de ecuaciones diferenciales y modelos composicionales, para así poder lograr modelar el comportamiento real de los datos.

# Estadística Bayesiana

La estadística Bayesiana es una metodología que emplea los elementos de la teoría de la decisión y el enfoque subjetivo de la probabilidad para construir una metodología unificada y coherente. Además de su interpretación formal, la metodología Bayesiana también permite las siguientes bondades:

Incorporar la información que el investigador tenga del problema, al análisis formal de inferencia estadística.

Distribuciones de probabilidad para describir el grado de creencia sobre los parámetros de un modelo.

El análisis natural de la distribución de la variable de respuesta en un modelo (la distribución predictiva).

# Estadística Bayesiana

La **distribución final** emplea el teorema de Bayes y es el punto de partida para realizar inferencias bajo un enfoque bayesiano. El teorema de Bayes contiene la verosimilitud, información a priori e información a posteriori (J.M. Bernardo and A.F.M. Smith, 2007):

$$p(\theta|y) = \frac{p(y|\theta)p(\theta)}{p(y)}$$

$$p(\theta|y) = \frac{p(y|\theta)p(\theta)}{\int_{\Omega} p(y|\theta)p(\theta)d\theta}$$

Una equivalencia a la ecuación de la **distribución final** sería considerar factor  $p(y)$  como un valor constante, con lo cual obtendríamos:

$$p(\theta|y) \propto p(y|\theta)p(\theta)$$

# Estadística Bayesiana

El punto de partida para todos los resúmenes de inferencia posteriores es la distribución final de la formula:

$$p(\theta|y) = \frac{p(y|\theta)p(\theta)}{\int_{\Omega} p(y|\theta)p(\theta)d\theta}$$

Por ejemplo:

$$P(\theta \in A) = \int_{\theta \in A} p(\theta|y)$$

Para estas situaciones tenemos 2 opciones:

1. Integración numérica.
2. Aproximarlo vía muestreo estocástico (métodos de simulación, procesos estocásticos, etc.)

# Modelos de EDO

Las ecuaciones diferenciales ordinarias tienen aplicaciones importantes y son una herramienta poderosa en el estudio de muchos problemas en las ciencias naturales y en la tecnología, esto nos invita a tener un interés especial en los sistemas de ecuaciones diferenciales para la **modelación matemática**.

El modelo FitzHugh–Nagumo es una versión simplificada del modelo de Hodgkin y Huxley (Keener and Sneyd, 2009), (Ramsay and Hooker, 2017) y (Allen, 2007), que describe la dinámica de la neurona más en detalle. Su sistema de ecuaciones diferenciales es:

$$\frac{dV}{dt} = \gamma \left( V - \frac{V^3}{3} + R \right) \qquad \frac{dR}{dt} = -\frac{1}{\gamma} (V - \alpha + \beta R)$$

Donde los parámetros  $\alpha, \beta, \gamma \in \mathbb{R}^+$  y valores iniciales  $V(t_0) = V_0, R(t_0) = R_0$ .

# Modelos de EDO

Dado el conjunto de datos  $y_{ij}$  y las predicciones (soluciones) de las ecuaciones diferenciales, se considera el siguiente modelo estadístico:

$$y_{ij} = \hat{y}_j(t_i|\theta) + \epsilon_{ij}$$

Donde:

$\hat{y}_j(t_i|\theta)$  es la solución numérica del sistema de ecuaciones del estado  $j$  (en este casos  $j = 1, 2$ ) en el tiempo  $t_i$  con valor de los parámetros  $\theta$ .

$\epsilon_{ij}$  es el error de medición de la observación  $i$  del estado  $j$ ,  $\epsilon_{ij} \sim N(0, \sigma_j)$

La distribución final resultante es:

$$p(\theta|y) \propto p(y|\theta) \times p(\theta)$$

Donde  $p(y|\theta)$  es la función de máxima verosimilitud de una distribución normal.

Sea  $n_j$  el número de observaciones del estado  $j$ :

$$p(y|\theta) = \prod_{j=1}^2 \prod_{i=1}^{n_j} N(y_{ij}|\hat{y}_j(t_i|\theta), \sigma_j)$$



# Metropolis-Hastings

El algoritmo Metropolis-Hastings asociado con una densidad objetivo  $p(\theta|y)$  requiere la elección de una densidad multivariada propuesta  $q$ , lo cual implica que el cociente adicional en el cálculo de  $\alpha$ , corrige cualquier asimetría de la distribución propuesta  $q$ .

---

## Algorithm 1: Metropolis-Hastings

---

Dado que  $\theta(t) = \theta_t$ ,

Generamos  $\theta^* \sim q(\theta|\theta_t)$ ,

Sea:

$$\alpha = \min \left( 1, \frac{p(\theta^*|y) q(\theta_t|\theta^*)}{p(\theta_t|y) q(\theta^*|\theta_t)} \right),$$

Entonces:

$$\theta(t+1) = \begin{cases} \theta^* & \text{con probabilidad } \alpha \\ \theta_t & \text{con probabilidad } 1 - \alpha \end{cases} .$$

---

Se conoce que una **elección adecuada** de una distribución de propuesta  $q$  para el algoritmo Metropolis-Hastings, es un factor crucial para la convergencia del algoritmo !!!!!

# Adaptive Metropolis (AM)

Los algoritmos adaptativos de MCMC intentan mejorar esta complicación al automáticamente ir "aprendiendo" los mejores valores de parámetros de los algoritmos de Monte Carlo de la cadena de Markov mientras se ejecutan.

Contrariamente a la intuición, es posible que los algoritmos MCMC adaptativos no siempre preserven la estacionariedad de  $p(\theta|y)$ .

# Robust Adaptive Metropolis (RAM)

Matti Vihola (2012) introdujo un nuevo Metropolis adaptativo llamado Robust Adaptive Metropolis.

El objetivo es buscar un factor de matriz  $S$  que capture la forma de  $p(\theta|y)$  y al mismo tiempo, permite alcanzar una tasa de aceptación media determinada.

A diferencia de otros MCMC adaptativos, este enfoque se basa en una sola fórmula para actualizar la matriz que es computacionalmente equivalente a la actualización del factor de covarianza en AM.

# Robust Adaptive Metropolis (RAM)

---

**Algorithm 2:** Robust Adaptive Metropolis

---

Dado que  $\theta(t) = \theta_t$ ,

Generamos  $\theta^* = \theta_t + S_{n-1}u_n$  donde  $u_n \sim q$  es un vector independiente, normalmente  $q = N(0, I)$ ,

Sea:

$$\alpha = \min \left( 1, \frac{p(\theta^* | y)}{p(\theta_t | y)} \right),$$

Entonces:

$$\theta(t+1) = \begin{cases} \theta^* & \text{con probabilidad } \alpha \\ \theta_t & \text{con probabilidad } 1 - \alpha \end{cases}$$

Se actualiza  $S_n$  tal que:

$$S_n S_n^T = S_{n-1} \left( I + \eta_n (\alpha_n - \alpha^*) \frac{u_n u_n^T}{\|u_n\|^2} \right) S_{n-1}^T$$

# Ejemplo RAM

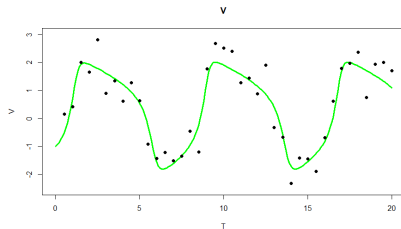


Figura 1: Verdadera solución para es estado  $V$  y los 40 datos simulados

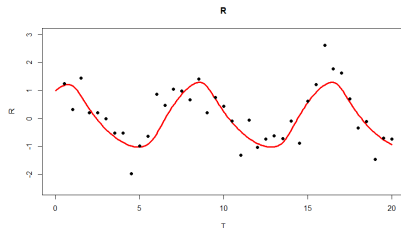


Figura 2: Verdadera solución para es estado  $R$  y los 40 datos simulados

# Ejemplo RAM

Por medio de programación en paralelo, con un periodo de calentamiento de 100,000 iteraciones y un lag de 60 para obtener una muestra final de tamaño 2,400. (118,000 por core vs 244,000 secuencial)

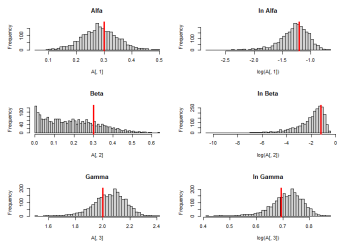
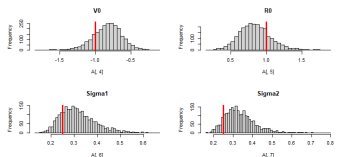


Figura 3: Histogramas de  $\alpha$ ,  $\beta$  y  $\gamma$  y sus respectivos logaritmos del tercer análisis



# Ejemplo RAM

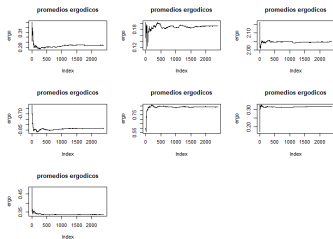


Figura 5: Promedios ergódicos de los parámetros  $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $V_0$ ,  $R_0$ ,  $\sigma_1^2$  y  $\sigma_2^2$

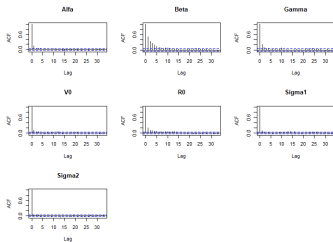


Figura 6: Gráficas de auto correlación de los parámetros  $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $V_0$ ,  $R_0$ ,  $\sigma_1^2$  y  $\sigma_2^2$

# Ejemplo RAM

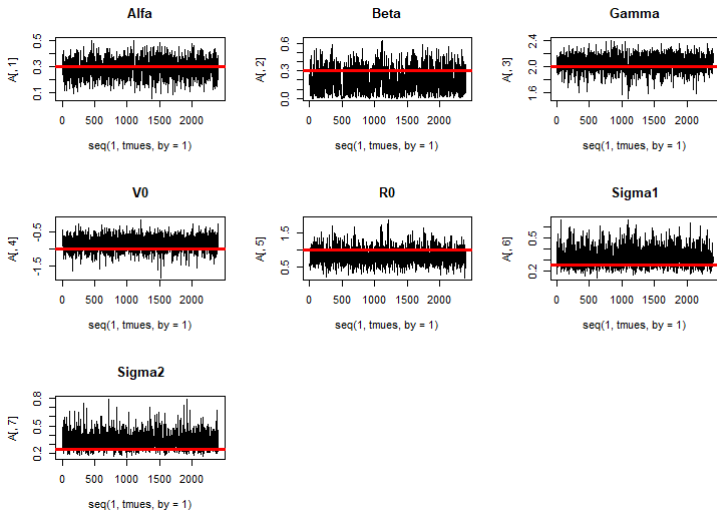


Figura 7: Trazas de la muestra final de los parámetros  $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $V_0$ ,  $R_0$ ,  $\sigma_1^2$  y  $\sigma_2^2$



# Ejemplo RAM

Parámetro	Verdadero valor	Media estimada	Mediana estimada	Intervalo de Prob.
$\alpha$	0.3	0.2835	0.2827	(0.1476 ; 0.4092)
$\beta$	0.3	0.1920	0.1770	( $2.543e^{-05}$ ; 0.4365)
$\gamma$	2	2.048	2.058	(1.802 ; 2.270)
$V_0$	-1	-0.8337	-0.8223	(-1.2128 ; -0.4369)
$R_0$	1	0.8380	0.8229	( 0.411 ; 1.268)
$\sigma_1^2$	0.25	0.3177	0.3060	(0.188 ; 0.480)
$\sigma_2^2$	0.25	0.3344	0.3227	(0.1968 ; 0.4882)

Cuadro 1: Resultados RAM

# Ejemplo RAM

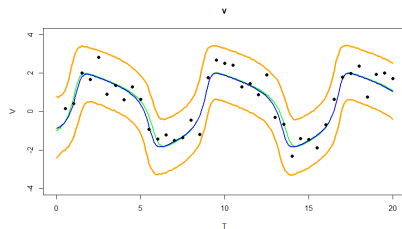


Figura 8: Bandas de probabilidad del estado  $V$

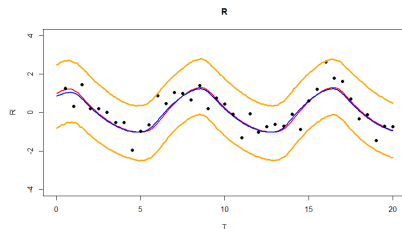


Figura 9: Bandas de probabilidad del estado  $R$

# Hamiltonian Monte-Carlo (HMC)

Las funciones hamiltonianas son funciones de la forma  $H(\theta, w)$  las cuales consisten de dos partes tal que:  $H(\theta, w) = U(\theta) + K(w)$  donde  $w, \theta \in \mathbb{R}^k$ . En general, las funciones  $U$  y  $K$  son funciones cuadráticas asociadas con las energías mecánicas potencial y cinética, respectivamente. Este hecho hace atractivo utilizar la estructura hamiltoniana para minimizar un funcional, como mostraremos (Steve Brooks et al., 2011).

Sea  $p(\theta|y)$  la función de distribución final de interés, entonces podemos definir  $U(\theta) := -\ln(p(\theta|y))$ . Ahora sea  $w \sim N_k(0, M)$  con  $M$  matriz de varianza-covarianza entonces:

$$H(\theta, w) = -\ln(p(\theta|y)) + w^T M^{-1} w.$$

# Hamiltonian Monte-Carlo (HMC)

Las trayectorias hamiltonianas están gobernadas por el siguiente sistema de ecuaciones diferenciales:

$$\begin{aligned}\frac{dw}{dt} &= \nabla_{\theta} \ln(p(\theta|y)) \\ \frac{d\theta}{dt} &= M^{-1}w.\end{aligned}$$

Donde  $\nabla_{\theta} \ln(p(\theta|y))$  es el gradiente del logaritmo de la distribución final. La solución a estas ecuaciones Hamiltonianas define el camino de donde se puede muestrear valores de  $\theta$ . La aleatoriedad del algoritmo viene de  $w \sim N_k(0, M)$ . El método de salto es una buena alternativa al método estándar de Euler para aproximar las soluciones numéricas a las ecuaciones hamiltonianas, y se resuelve con un salto de tamaño  $\epsilon$

# Ejemplo HMC

---

**Algorithm 3:** Hamiltonian Monte Carlo

---

Considerando que  $\theta(t) = \theta_t$ ,

Generamos  $w_t \sim N(0, M)$ ,

Resolvemos el mediante el método de salto,  $L$  veces con paso  $\epsilon$  el sistema de ecuaciones hamiltonianas, generando así  $\theta_{t+1}$  y  $w_{t+1}$ ,

Sea:

$$\alpha = \min \left( 1, \frac{\exp(\ln(p(\theta_{t+1}|y)) - (1/2)w_{t+1}^T M w_{t+1})}{\exp(\ln(p(\theta_t|y)) - (1/2)w_t^T M w_t)} \right),$$

Entonces:

$$\theta(t+1) = \begin{cases} \theta_{t+1} & \text{con probabilidad } \alpha \\ \theta_t & \text{con probabilidad } 1 - \alpha \end{cases} .$$

# Ejemplo HMC

Se realizó el método para 10,000 iteraciones, y con inferencia sobre las últimas 1,000.

Parámetro	Verdadero valor	Media estimada	Mediana estimada	Intervalo de Prob.
$\alpha$	0.3	0.2533	0.2517	(0.1379 ; 0.3641)
$\beta$	0.3	0.3952	0.3968	(0.2107 ; 0.5814)
$\gamma$	2	1.990	2.001	(1.728 ; 2.268)
$V_0$	-1	-1.0107	-0.987	(-1.398 ; -0.656)
$R_0$	1	1.176	1.154	(0.806 ; 1.6064)
$\sigma_1^2$	0.25	0.249	0.241	(0.1507 ; 0.3693)
$\sigma_2^2$	0.25	0.220	0.2125	(0.134 ; 0.3269)

Cuadro 2: Resultados HMC con  $L=20$  y  $\epsilon=0.01$

# Ejemplo HMC

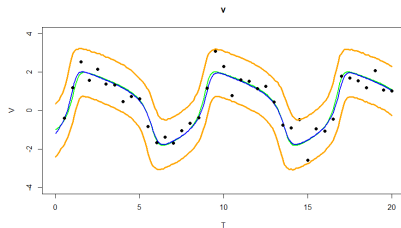


Figura 10: Bandas de probabilidad del estado  $V$  usando HMC

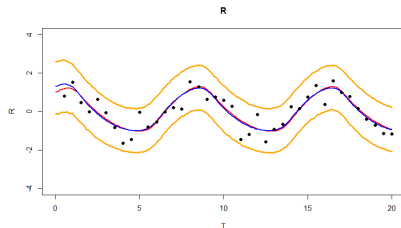


Figura 11: Bandas de probabilidad del estado  $R$  usando HMC

# Aproximate Bayesian Computation (ABC)

¿ Qué sucede cuando la función de verosimilitud  $L(y|\theta)$  no está disponible analíticamente o computacionalmente sea muy costosa de evaluar?

---

**Algorithm 4:** Muestreo de rechazo sin verosimilitud

---

Generamos  $\theta' \sim p(\theta)$ ,

Generar  $z$  datos del modelo  $p(z|\theta')$ ,

Aceptar  $\theta'$  si  $z \approx y$ .

---

Una función de ponderación utilizada con frecuencia es el kernel de densidad uniforme, donde  $T(y)$  se distribuye uniformemente en la esfera centrada en  $T(z)$  con radio  $\epsilon$ . Esto se escribe comúnmente como:

$$\pi_{\epsilon}(y|z, \theta) \begin{cases} 1 & \text{si } \rho(T(z), T(y)) \leq \epsilon \\ 0 & \text{en otro caso} \end{cases}$$



# Aproximate Bayesian Computation (ABC)

La efectividad de los enfoques ABC depende de elegir una función de discrepancia adecuada  $\rho(z, y)$  y una tolerancia adecuada  $\epsilon$ . Utilizamos la siguiente medida para ambos estados  $V$  y  $R$  tal que  $\rho(z, y) < \epsilon$ .

$$\rho(z, y) = \sum_{j=1}^n (z_j - y_j)^2$$

---

## Algorithm 5: MCMC-ABC

---

Obtener  $\theta_0$  y  $z_0$  del muestreo de rechazo ABC.

Para  $t = 1, \dots, T$

Genera  $\theta^* \sim q(\theta^* | \theta^{t-1})$ ,

Calcula

$$r = \frac{p(\theta^*)q(\theta_{t-1} | \theta^*)}{p(\theta_{t-1})q(\theta^* | \theta_{t-1})},$$

Si  $U(0, 1) < r$  entonces:

Simula  $z^*$  del modelo dado  $\theta^*$

Si  $\rho(z^*, y) < \epsilon$  entonces:

$\theta_t := \theta^*$  si no  $\theta_t := \theta_{t-1}$ ,

Si no entonces  $\theta_t := \theta_{t-1}$ .

# Ejemplo ABC

Con 150,000 iteraciones de calentamiento más un lag de 500 para tamaños de muestra 300 por core (300,000 por core vs 1,350,000 secuencial)

Parámetro	Verdadero valor	Media estimada	Mediana	Intervalo de Prob.
$\alpha$	0.3	0.3793	0.3961	(0.1583 ; 0.5045)
$\beta$	0.3	0.3355	0.3799	(0.151 ; 0.455)
$\gamma$	2	1.899	1.934	(1.55 ; 2.195)
$V_0$	-1	-0.5025	-0.6659	(-1.0423 ; 0.4169)
$R_0$	1	0.9775	1.1886	(-0.1717 ; 2.0899)
$\sigma_1$	0.5	0.6938	0.749	(0.102 ; 1.4)
$\sigma_2$	0.5	0.7405	0.7974	(0.218 ; 1.415)

Cuadro 3: Resultados ABC con  $\epsilon=0.5$

# Ejemplo ABC

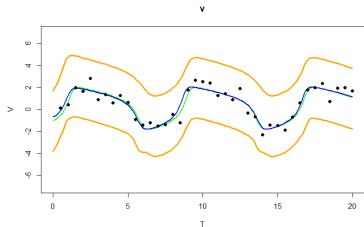


Figura 12: Bandas de probabilidad del estado  $V$  del escenario  $\epsilon=0.5$  con estimación de la variabilidad

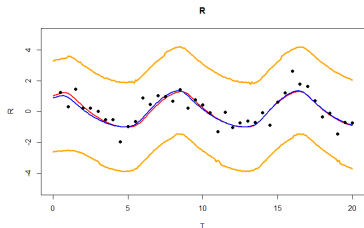


Figura 13: Bandas de probabilidad del estado  $R$  del escenario  $\epsilon=0.5$  con estimación de la variabilidad

# Comparación

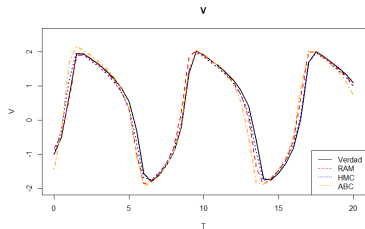


Figura 14: Comparación soluciones medias de los distintos métodos MCMC en el estado  $V$

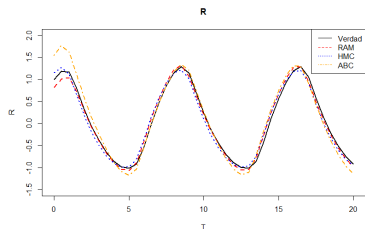


Figura 15: Comparación soluciones medias de los distintos métodos MCMC en el estado  $R$

# Comparación

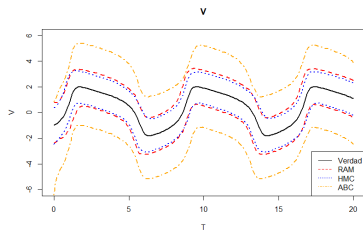


Figura 16: Comparación bandas de probabilidad de los distintos métodos MCMC en el estado  $V$

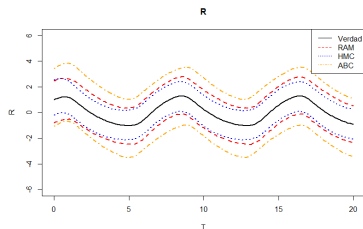


Figura 17: Comparación bandas de probabilidad de los distintos métodos MCMC en el estado  $R$

# Comparación

Método MCMC	Iteraciones	Tiempo de CPU.
HMC	10,000	10 horas
RAM	118,000 (Paralelo)	40 minutos
ABC	300,000 (Paralelo)	33 minutos

Cuadro 4: Tiempo de Ejecución

# Ventajas y Desventajas

Cuadro 5: Ventajas y Desventajas

<i>Métodos MCMC</i>	<i>Ventajas</i>	<i>Desventajas</i>
RAM	<p>Es fácil de implementar.</p> <p>Es un método adaptativo.</p> <p>Es un precio justo entre tiempo de cómputo y calidad de estimaciones.</p>	<p>Requiere de varias iteraciones y éstas aumentan según la dificultad del modelo a analizar.</p>
HMC	<p>El gradiente nos ayuda a mover el vector de parámetros de forma más eficiente.</p> <p>No requiere tantas iteraciones</p> <p>Rápido si el gradiente se puede calcular analíticamente.</p>	<p>Requiere el cálculo del gradiente y si éste es numérico puede ser muy costoso computacionalmente.</p>
ABC	<p>No requiere verosimilitud.</p> <p>Es bastante rápido.</p>	<p>Depende de las distribuciones iniciales.</p> <p>Depende del valor de <math>\epsilon</math>.</p> <p>Necesita que la medida de comparación de los datos sea suficiente.</p> <p>Es complicado estimar la varianza.</p>
MH	<p>Es fácil de implementar.</p> <p>Es un método clásico para la inferencia bayesiana.</p>	<p>Depende fuertemente de la elección de la distribución propuesta.</p> <p>Puede tener tasas de aceptación muy baja.</p>

# Modelo predador-presa

Otra ventaja de usar sistemas de ecuaciones diferenciales, es que sino llegáramos a tener información de un estado, la propia naturaleza de los sistemas de ecuaciones diferenciales (que explican la relación entre los estados) permitiría que los datos de otros estados nos pueden explicar el comportamiento de todo el sistema. El modelo explica la relación entre un depredador metazoario, el rotífero *Brachionus calyciflorus s.s.*, y su presa *Monoraphidium minutum*, una alga verde unicelular, y las algas crecen en un medio con nitrógeno (Rosenbaum, 2019).

$$\begin{aligned}\frac{dS}{dt} &= (S^* - S)\delta - \frac{1}{c_A} \frac{f_A S}{h_A + S} A \\ \frac{dA}{dt} &= \frac{f_A S}{h_A + S} A - \frac{1}{c_R} \frac{f_R A}{h_R + A} R - \delta A \\ \frac{dR}{dt} &= \frac{f_R A}{h_R + A} R - \delta R\end{aligned}$$

Donde  $S^*, \delta, h_A, h_R$  se pueden suponer como parámetros constantes.



# Modelo predador-presa

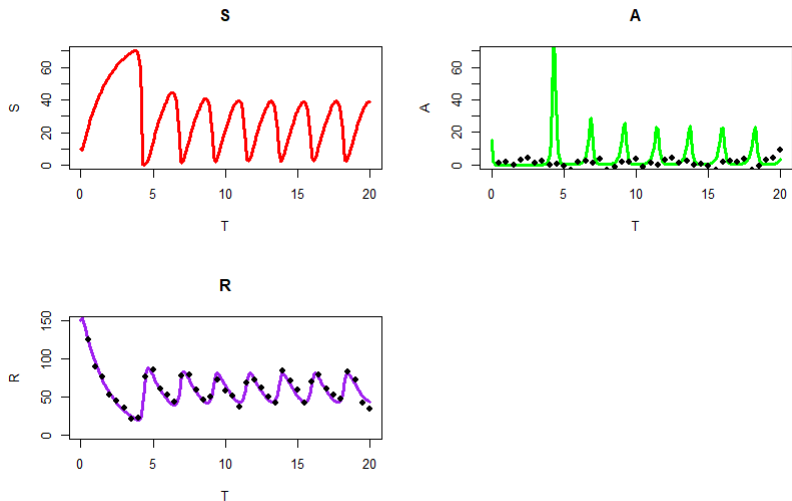


Figura 18: Solución del sistema predador-presa con los datos simulados del estado  $A$  y  $R$

# Modelo predador-presa

Se realizaron simulaciones con 40 datos, con un periodo de calentamiento e iteraciones intermedias de 200,000 y lag 100 para poder obtener una muestra final de tamaño 1,600 utilizando programación en paralelo (220,000 por core) con un tiempo de 6 horas.

Parámetro	Verdadero valor	Media	Mediana	Intervalo de Prob.
$f_A$	15	15.29	15.30	(11.626 ; 16.6134)
$f_R$	10	9.937	9.952	(9.0533 ; 10.7972)
$1/c_A$	0.6	0.5811	0.5821	(0.5396 ; 0.6260)
$1/c_R$	1.5	1.535	1.53	(1.4052 ; 1.6701)
$S_0$	10	6.5287	1.3192	$(5.63e^{-04}; 3710802)$
$A_0$	15	54.895	24.882	(2.21 ; 145.93)
$R_0$	150	120.87	146.58	(24.893 ; 161.868)
$\sigma_1^2$	4	5.265	4.036	(2.35 ; 14.339)
$\sigma_2^2$	25	28.58	24.12	(14.756 ; 55.545)

Cuadro 6: Modelo predador-presa

# Modelo predador-presa

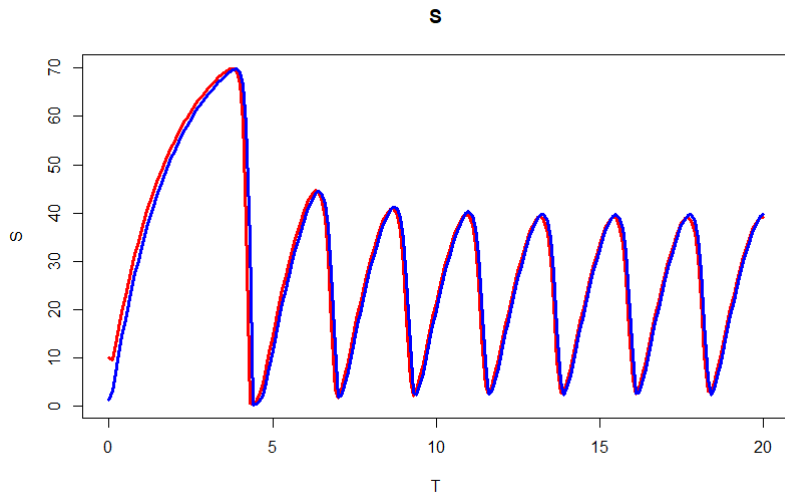


Figura 19: Solución media del estado  $S$  y su verdadera solución usando el método RAM

# Modelo predador-presa

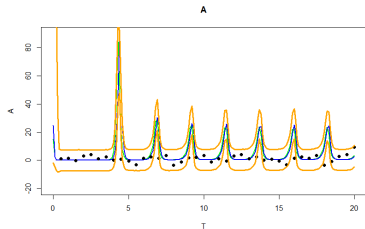


Figura 20: Bandas de probabilidad del estado  $A$  usando el método RAM

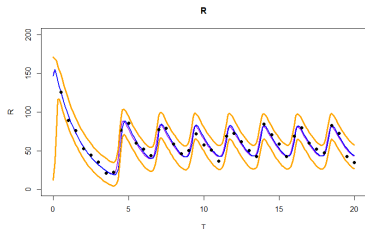


Figura 21: Bandas de probabilidad del estado  $R$  usando el método RAM

# Modelo predador-presa

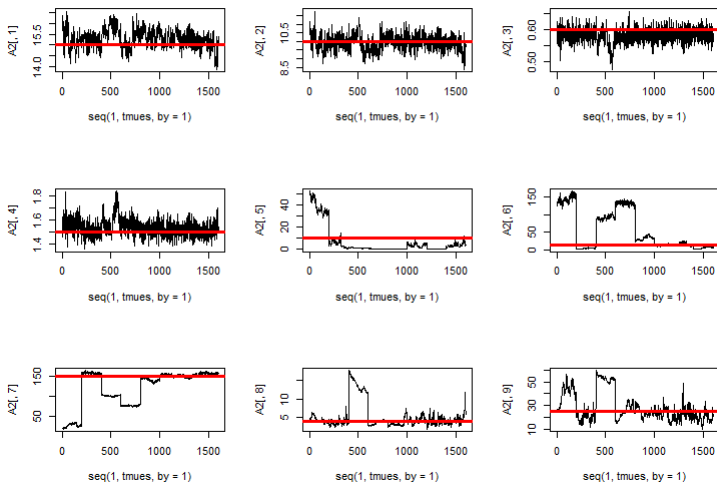


Figura 22: Trazas de la muestra final de los parámetros  $f_A, f_R, \frac{1}{c_A}, \frac{1}{c_R}, S_0, A_0, R_0, \sigma_1^2, \sigma_2^2$



# Datos Circulares

Thetas Reales



Figura 24: Datos del vector  $\mu=(1,1)$

Thetas Reales



Figura 25: Datos del vector  $\mu=(0.5,0.5)$

# Datos Circulares

Parámetro	Verdadero valor	Media	Mediana	Intervalo de Prob.
<b>HMC</b>				
$\mu_1$	5	4.816	4.809	(4.13 ; 5.48)
$\mu_2$	5	4.905	4.896	(4.18 ; 5.61)
<b>RAM</b>				
$\mu_1$	5	4.797	4.786	(4.15 ; 5.49)
$\mu_2$	5	4.885	4.868	(4.15 ; 5.53)
<b>MH</b>				
$\mu_1$	5	4.876	4.871	(4.17 ; 5.51)
$\mu_2$	5	4.961	4.944	(4.32 ; 5.66)
<b>ABC</b>				
$\mu_1$	5	6.479	6.274	(4.2482 ; 8.7071)
$\mu_2$	5	6.595	6.365	(4.6391 ; 9.3031)

Cuadro 7: Resultados escenario  $\mu=(5,5)$

Método MCMC	Iteraciones	Tiempo de CPU
HMC	30,000	18 minutos
RAM	160,000	4 minutos
MH	40,000	30 minutos
ABC	—	1 minuto



# Datos Circulares

Parámetro	Verdadero valor	Media	Mediana	Intervalo de Prob.
<b>HMC</b>				
$\mu_1$	1	1.170	1.174	(0.947 ; 1.423)
$\mu_2$	1	0.976	0.976	(0.735 ; 1.188)
<b>RAM</b>				
$\mu_1$	1	1.182	1.183	(0.919 ; 1.416)
$\mu_2$	1	0.978	0.979	(0.744 ; 1.225)
<b>MH</b>				
$\mu_1$	1	1.216	1.215	(0.966 ; 1.468)
$\mu_2$	1	1.010	1.006	(0.766 ; 1.251)
<b>ABC</b>				
$\mu_1$	1	1.1952	1.1610	(0.7054 ; 1.771)
$\mu_2$	1	1.0393	1.0407	(0.6969 ; 1.3404)

Cuadro 9: Resultados escenario  $\mu=(1,1)$

Método MCMC	Iteraciones	Tiempo de CPU
HMC	30,000	13 minutos
RAM	160,000	4 minutos
MH	40,000	30 minutos
ABC	—	1 minuto

# Datos Circulares

Parámetro	Verdadero valor	Media	Mediana	Intervalo de Prob.
<b>HMC</b>				
$\mu_1$	0.5	0.5001	0.5004	(0.290 ; 0.726)
$\mu_2$	0.5	0.283	0.361	(0.137 ; 0.582)
<b>RAM</b>				
$\mu_1$	0.5	0.497	0.499	(0.263 ; 0.738)
$\mu_2$	0.5	0.362	0.359	(0.138 ; 0.592)
<b>MH</b>				
$\mu_1$	0.5	0.510	0.509	(0.262 ; 0.712)
$\mu_2$	0.5	0.377	0.378	(0.147 ; 0.584)
<b>ABC</b>				
$\mu_1$	0.5	0.4943	0.4882	(0.1061 ; 0.9053)
$\mu_2$	0.5	0.3863	0.3877	(0.0979 ; 0.6617)

Cuadro 11: Resultados escenario  $\mu=(0.5,0.5)$

Método MCMC	Iteraciones	Tiempo de CPU
HMC	30,000	13 minutos
RAM	160,000	4 minutos
MH	40,000	30 minutos
ABC	—	1 minuto

# Datos Circulares

La medida de distancia utilizada para el algoritmo de muestreo de rechazo ABC, para el primer escenario, que se empleó fue el siguiente (Mardia & Jupp, 2009)

$$\rho(\theta, z) = \frac{1}{n} \sum_{j=1}^n (1 - \text{Cos}(z_j - \bar{\theta}_D)),$$

donde  $\bar{\theta}_D$  es la media muestral de los datos circulares  $\theta$ . El valor de  $\epsilon$  considerado fue de  $\epsilon = 0.01$ .

La medida de distancia utilizada para el algoritmo de muestreo de rechazo ABC, para el segundo y tercer escenario, que se empleó fue el siguiente (Mardia & Jupp, 2009)

$$\rho(\theta, z) = \frac{1}{n^2} \sum_{i=1}^n (1 - \text{Cos}(z_i - \theta_i)),$$

con el valor de  $\epsilon = 0.0005$

# Temas Abarcados

Estadística Bayesiana

Sistemas de Ecuaciones Diferenciales

Datos Circulares

Simulación Estocástica

Programación

Modelación Matemática

# Gracias!!

# Bibliografía I



Amani A. Alahmadi, Jennifer A. Flegg, Davis G. Cochrane, Christopher C. Drovandi, and Jonathan M. Keith.

A comparison of approximate versus exact techniques for bayesian parameter inference in nonlinear ordinary differential equation models.

*Royal Society Open Science*, 7(3):Article number: 191315, March 2020.



J.M. Bernardo and A.F.M. Smith.

*Bayesian Theory*.

Wiley Series in Probability and Statistics. Wiley, 2007.



G.E.P. Box and G.C. Tiao.

*Bayesian inference in statistical analysis*.

Addison-Wesley series in behavioral science: quantitative methods. Addison-Wesley Pub. Co., 1973.



Steve Brooks, Andrew Gelman, Galin Jones, and Xiao-Li Meng.

*Handbook of Markov Chain Monte Carlo*.

CRC press, 2011.









Andrew Gelman, John B. Carlin, Hal S. Stern, and Donald B. Rubin.

*Bayesian Data Analysis*.

Chapman and Hall/CRC, 2nd ed. edition, 2004.

# Bibliografía II

-  Heikki Haario, Eero Saksman, and Johanna Tamminen.  
An adaptive Metropolis algorithm.  
*Bernoulli*, 7(2):223 – 242, 2001.
-  Peter D. Hoff.  
*A First Course in Bayesian Statistical Methods*.  
Springer Publishing Company, Incorporated, 1st edition, 2009.
-  K.V. Mardia and P.E. Jupp.  
*Directional Statistics*.  
Wiley Series in Probability and Statistics. Wiley, 2009.
-  Cody McRae.  
Bayesian inference in nonlinear differential equation models.  
*Monash University*, 2019.
-  Christian P. Robert.  
*The Bayesian choice: a decision-theoretic motivation*.  
Springer-Verlag, 1994.
-  G.O. Roberts and J.S. Rosenthal.  
Coupling and ergodicity of adaptive mcmc.  
*Journal of Applied Probability*, 2007.

# Bibliografía III



G.O. Roberts and J.S. Rosenthal.

Examples of adaptive mcmc.

*Journal of Computational and Graphical Statistics*, (18(2)):349–367, 2009.



Benjamin Rosenbaum, Michael Ratz, Guntram Weithoff, Gregor F. Fussmann, and Ursula Gaedke.

Estimating parameters from multiple time series of population dynamics using bayesian inference.

*Frontiers in Ecology and Evolution*, 6:234, 2019.



Samuel Thomas.

*hmclearn: Fit Statistical Models Using Hamiltonian Monte Carlo*, 2020.

R package version 0.0.4.



M. Vihola.

Robust adaptive metropolis algorithm with coerced acceptance rate.

*Stat Comput* 22, page 997–1008, 2012.