Trabajo práctico número 4 de Astrometría

R.C. Martin[1]

[1] *Observatorio Astronómico de Córdoba, UNC, Argentina*

Received: 16/10/2024 / Accepted: ...

Resumen / En este trabajo se ajustará la función de Schechter a un conjunto de datos SLOAN en la banda r. Para ello se utilizará el algoritmo Metrópolis-Hastings y Descenso por el gradiente. Esto se realiza con el objetivo de explorar el espacio de parámetros. Además, compararemos métodos frecuentistas y bayesianos para estimar parámetros de una distribución.

Keywords / Estadística, Python.

1. Introducción

Dado un modelo m, con parámetros ϕ y un conjunto de datos d, la probabilidad de que los parámetros se ajusten a los datos, dado este modelo, puede calcularse con el teorema de Bayes:

$$p(\phi|d,m) = \frac{p(d|\phi,m)p(\phi|m)}{p(d|m)} \quad (1)$$

$$p(\phi|d,m) = \frac{Likelihood \times Prior}{Evidencia} \quad (2)$$

Siendo la evidencia, la probabilidad marginal del Likelihood para el modelo m:

$$p(d|m) = \int_\Omega p(d|\phi,m)p(\phi|m)d\phi \quad (3)$$

El conjunto de datos que utilizaremos en este trabajo consiste en magnitudes 'r' SLOAN y sus correspondientes valores en la 'Función de Luminosidad'. A estos les ajustaremos el modelo de Schechter (sea $\beta = 0.4$):

$$m = \beta \ln(10)\, \phi_*\, 10^{-\beta\,(M-M_*)\,(\alpha+1)}\, e^{-10^{-\beta\,(M-M_*)}} \quad (4)$$

Para ello, utilizaremos Cadenas de Markov Monte Carlo y Descenso por el Gradiente. Para el primer caso, debemos definir bien los priors y utilizar un algoritmo de Metrópolis-Hastings.

2. Ejercicio 1

Graficamos los datos y la función de Schechter con los parámetros $(\phi_*, M_*, \alpha) = (1.46e-2, -20.83, -1.20)$, ver figura 1.

3. Ejercicio 2

En el código "cadenas.py" se realiza el método de Metrópolis-Hastings. Cabe destacar el proceso de selección de los priors: para ello, en las figuras 2,3 y 4 graficamos la función de ejemplo variando los parámetros para ver cómo afecta a la solución. Concluimos entonces

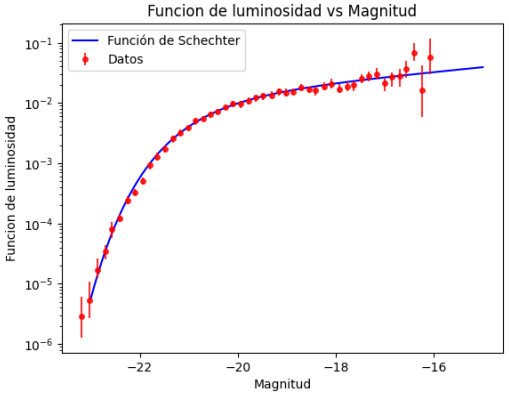

Figura 1: Magnitud vs función luminosidad. La línea sólida es la función de Schechter con $(\phi_*, M_*, \alpha) = (1.46e-2, -20.83, -1.20)$ mientras que los puntos son los datos SLOAN.

que convenía usar priors planos, para acotar los valores por los que pasan los parámetros, estos son:

$$-22.913 < M < -18.747 \quad (5)$$

$$0.01168 < \phi < 0.01752 \quad (6)$$

$$-1.32 < \alpha < -1.08 \quad (7)$$

Tras aplicar el algoritmo a los datos, podemos observar cómo convergen en la figura 5. Tomamos la media de los últimos 2000 datos de la cadena para estimar el mejor parámetro de ajuste. Concluímos entonces que:

$M_* = -20.77814603439193$
$\phi_* = 0.016141153740790885$
$\alpha = -1.1015282525168226$

Considerando una sola cadena, posteriormente consideraremos varias para promediar entre ellas. Además, en la figura 6 se observan los espacios de parámetros, donde no está aún muy clara la sobredensidad.

✉ *Contacto* / romanciromartin@mi.unc.edu.ar

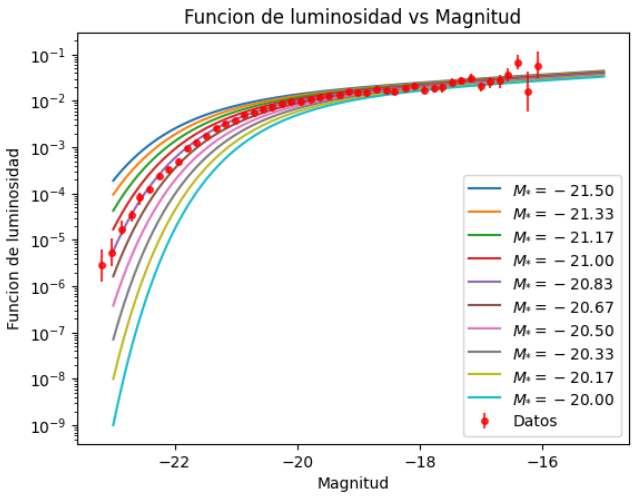

Figura 2: El mismo gráfico de la figura 1, variando M_*

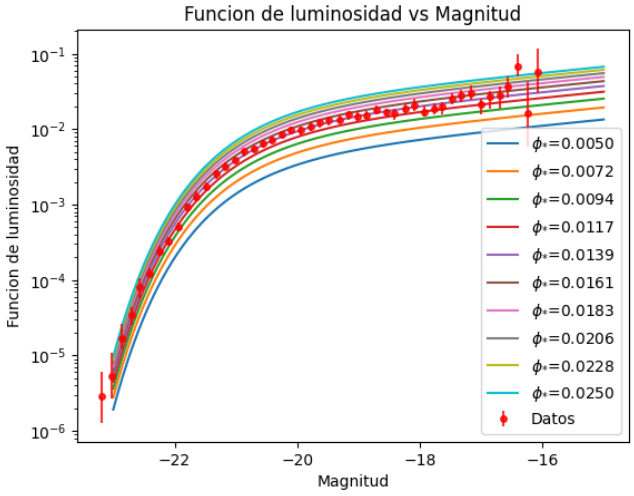

Figura 3: El mismo gráfico de la figura 1, variando ϕ_*

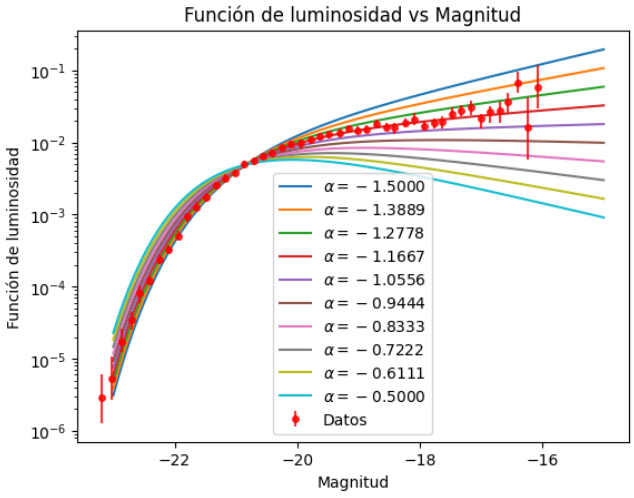

Figura 4: El mismo gráfico de la figura 1, variando α_*

4. Ejercicio 3

Realizaremos 20 cadenas con diferentes inicializaciones, todas dentro de los priors mencionados inicialmente. Estos serán al azar con una distribución uniforme. En la figura 7 graficamos los parámetros versus el logaritmo de su likelihood, para cada cadena. Se observa cómo, en todos los casos, las cadenas terminan convergiendo alrededor de un máximo de probabilidad, formando una distribución alrededor del valor más probable.

Por otra parte, en la figura 8 vemos cómo todas las cadenas convergen con alrededor de 4000 eslabones; por lo tanto, a la hora de realizar cuentas y gráficos, quemamos (es decir, tiramos) los primeros 4000 pasos.

Con las marginalizaciones, ver figura 9, sucede algo similar que con el gráfico parámetros versus $\log(Likehood)$ (figura 7). Los parámetros poseen distribuciones similares, alrededor de un valor, que no es el mismo para cada cadena, pero están todos cerca.

Por otra parte, en la figura 10 graficamos las curvas de nivel de los espacios de parámetros, donde claramente podemos observar los máximos de probabilidad en el espacio de 2 parámetros. Podemos observar que para "M vs. ϕ" y para "ϕ vs. α " es similar a una banana o una elipse. Mientras tanto, "M vs. α " es algo más redondeado. Los puntos son una cadena con un paso algo más grueso para muestrear mejor el espacio de parámetros (pero con una convergencia más deficiente). Este gráfico en particular se realizó con el código "cadenas_gráfico.py".

Finalmente, calculamos la máxima distancia entre las 20 cadenas en cada paso y la graficamos en la figura 11. Podemos observar que la máxima distancia entre cadenas es 3.968, al comienzo. Luego, esto decae rápidamente, pues todas convergen a valores muy próximos (misma deducción que se hizo con las marginalizaciones).

5. Ejercicio 4

Ahora que tenemos varias cadenas que han convergido satisfactoriamente, vamos a compararlas y estimar momentos de la distribución de los parámetros. Ya vimos en la Sección 4 que la distancia entre las cadenas decae rápidamente y que para un resultado satisfactorio, basta con tirar los primeros 4000 puntos (siendo algo conservadores). En la tabla 1, se muestra el promedio de cada parámetro. Luego, el promedio de estos promedios y el promedio de sus varianzas son buenos estimadores para los parámetros de la función y su varianza, obteniendo (redondeando según el error, que es la varianza sobre $\sqrt{20}$):

$M_* = -20.7779 \pm 0.0002$
$\phi_* = 0.016149 \pm 0.000004$
$\alpha = -1.1013 \pm 0.0001$

6. Ejercicio 5

En contraste con lo hecho hasta el momento, no exploraremos el espacio de parámetros, sino que con el Método de Descenso por el Gradiente, vamos a ir diréctamente

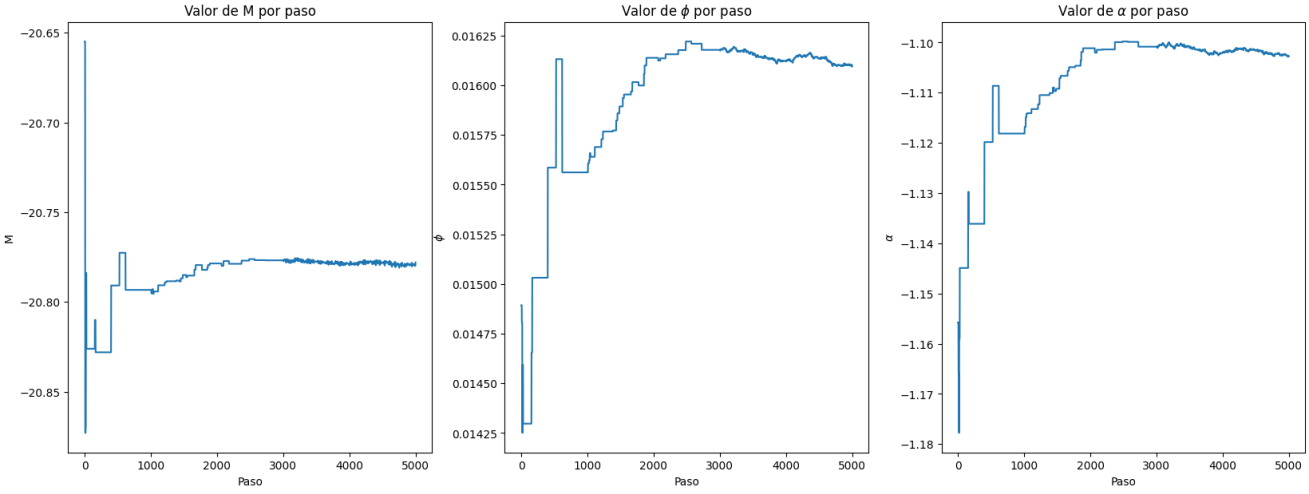

Figura 5: Parámetros del modelo de Schechter en cada paso de la cadena.

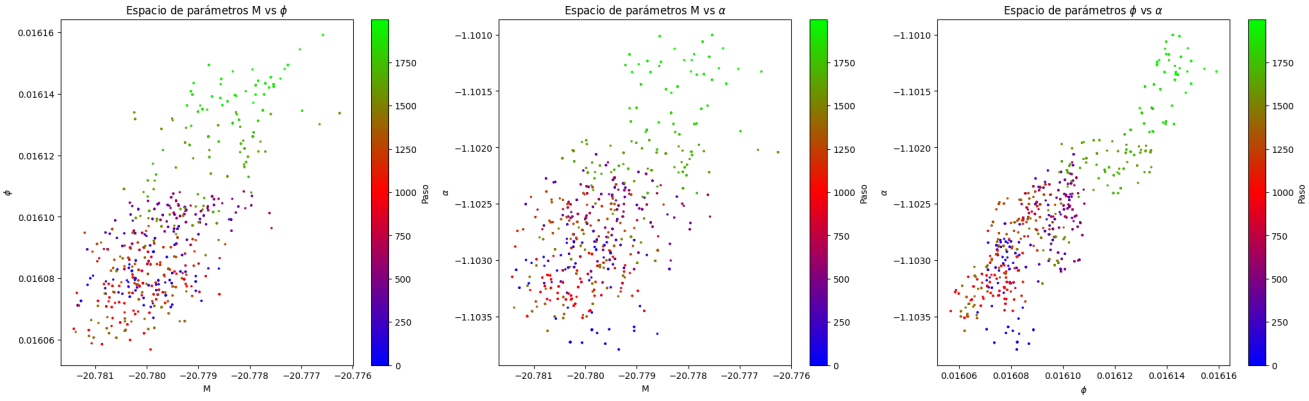

Figura 6: Espacio de parámetros muestreado con una sola cadena.

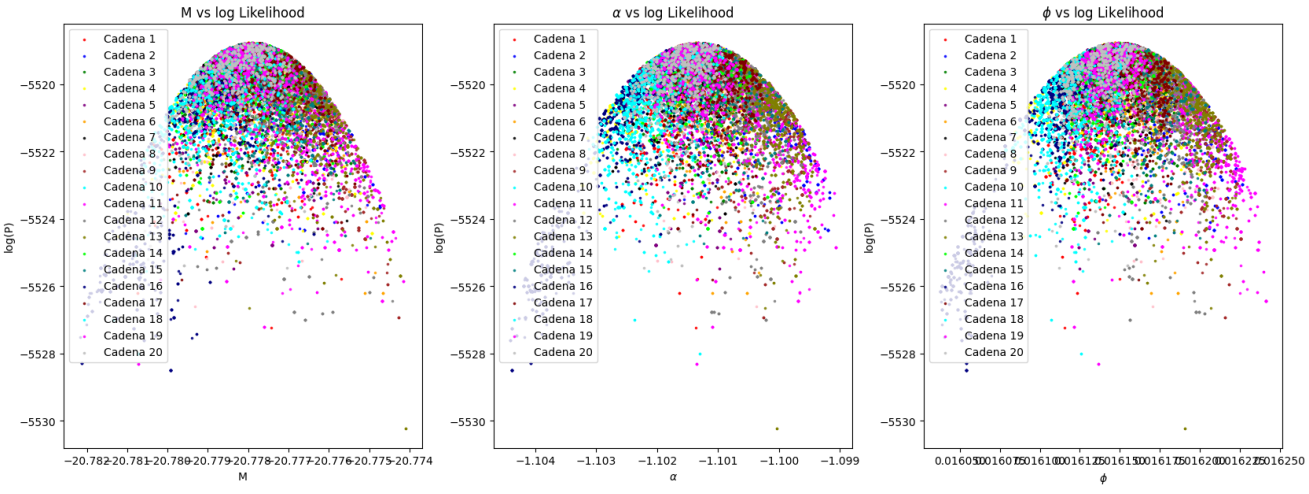

Figura 7: Parámetros vs Log Likehood para cada una de las cadenas.

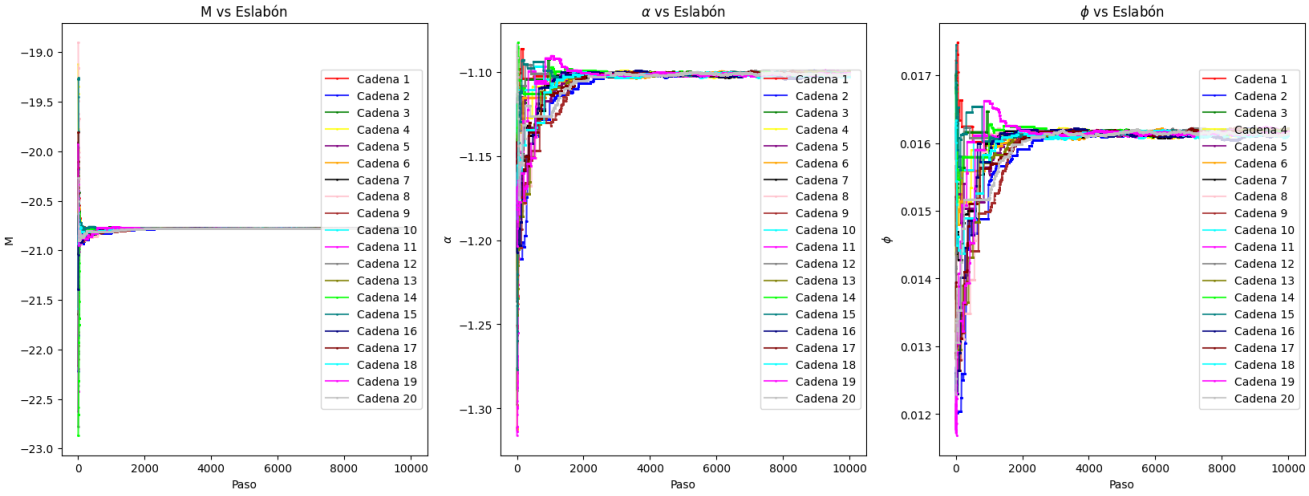

Figura 8: Parámetros vs Paso para cada cadena.

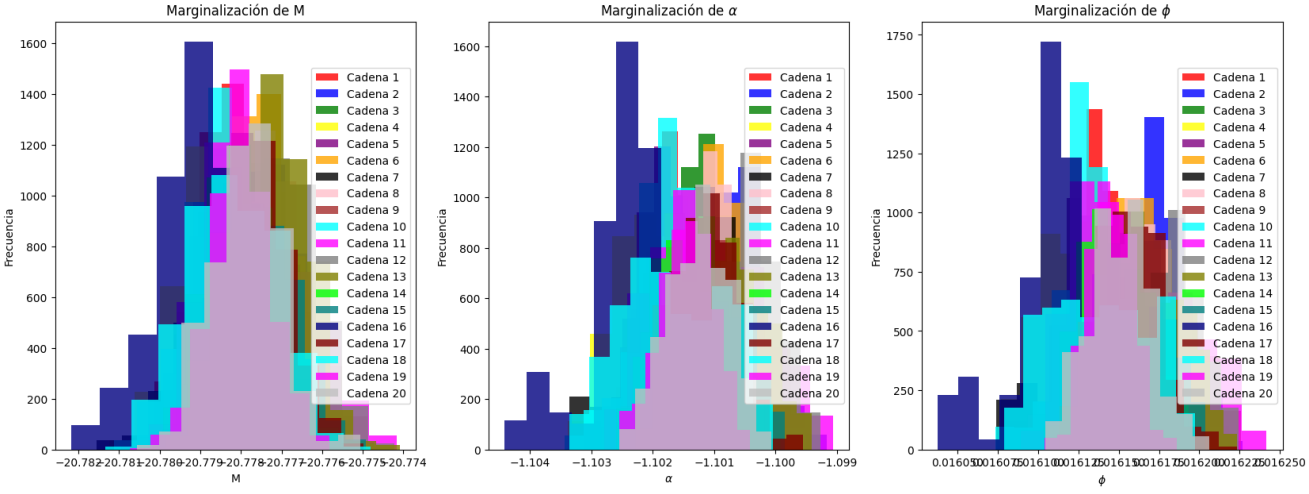

Figura 9: Marginalización de los parámetros para cada cadena.

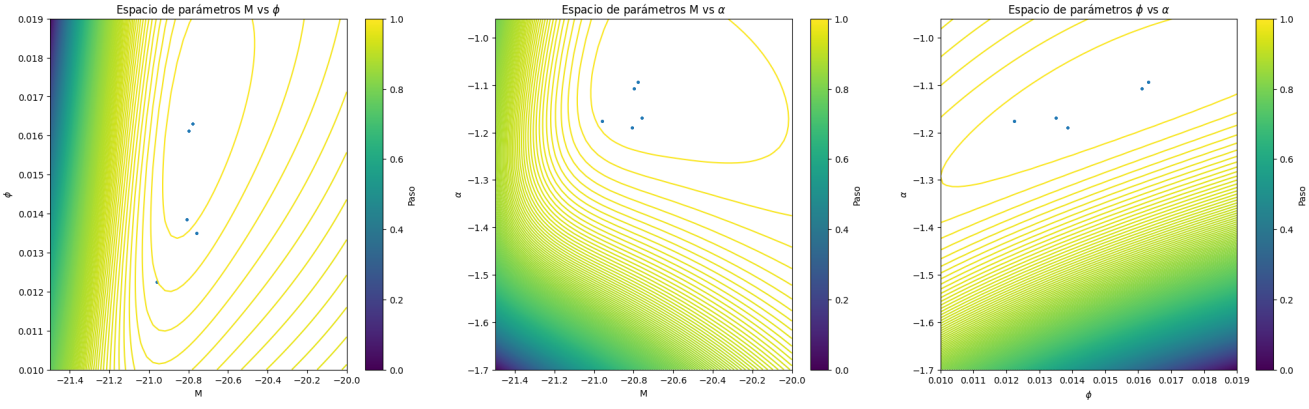

Figura 10: Curvas de nivel de los parámetros.

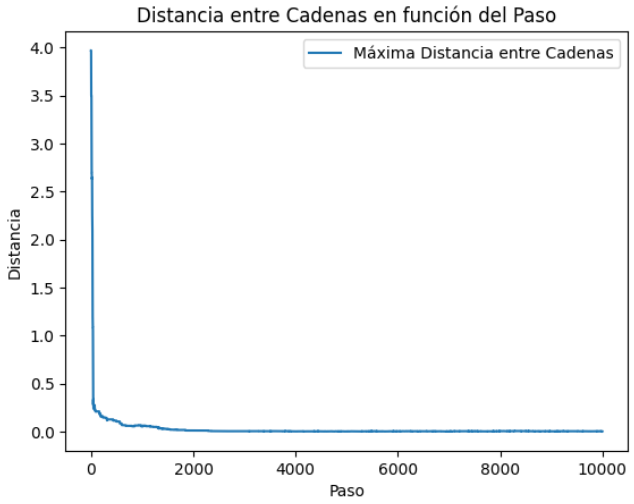

Figura 11: Máxima distancia entre las 20 cadenas para cada paso.

Tabla 1: Valores promedio de M, ϕ, y α para cada cadena

$\overline{M}$	$\overline{\phi}$	$\overline{\alpha}$
-20.778306	0.016135	-1.101715
-20.777760	0.016154	-1.101218
-20.777627	0.016159	-1.101074
-20.778349	0.016134	-1.101731
-20.778272	0.016137	-1.101649
-20.777660	0.016157	-1.101150
-20.778067	0.016143	-1.101492
-20.777521	0.016163	-1.100979
-20.777794	0.016152	-1.101266
-20.778343	0.016134	-1.101706
-20.777729	0.016153	-1.101228
-20.777452	0.016164	-1.100929
-20.777253	0.016170	-1.100817
-20.777824	0.016151	-1.101267
-20.777634	0.016158	-1.101108
-20.779023	0.016110	-1.102326
-20.777568	0.016159	-1.101053
-20.778071	0.016142	-1.101539
-20.777902	0.016149	-1.101355
-20.777975	0.016147	-1.101371

al valor que minimice χ^2. Para ello, la función dará saltos en la dirección del gradiente,que viene dado por la ecuación 8.

$$(\nabla\chi^2)_k = -2\sum_{n=1}^{N} \frac{y_n - y(\vec{x_n}, \vec{a})}{\sigma_n^2} \frac{\partial y(\vec{x_n}, \vec{a})}{\partial a_k} \tag{8}$$

En el código "gradiente.py", realizamos esto considerando y como la función de Schechter. En la figura 12 vemos cómo los parámetros convergen rápidamente, en tan sólo 2000 pasos. No obstante, en la figura 13 donde están las marginalizaciones de los parámetros para los últimos 8000 pasos, podemos observar que cuando nos acercamos al valor máximo, este método comienza a dar vueltas sobre el mismo lugar.

Por lo tanto, este método resulta más rápido para estimar un buen valor para los parámetros, pero no conseguimos explorar el espacio de parámetros. Además, posee problemas cerca del mejor valor, puesto que el gradiente tiende a 0. Esto puede solucionarse cambiando al método de Newton cerca del mínimo. Cabe destacar también que este método demoró más computacionalmente haciendo la misma cantidad de pasos que el Método de Montecarlo; no obstante, no era necesario realizar la misma cantidad de pasos, bastaba con muchos menos. Finalmente, destacar que hay que correr varios para estar seguros de no caer en un máximo secundario.

Con este intento que se observa en las figuras 12 y 13, obtuvimos como parámetros:

$M_* = -20.77789884 \pm 0.00000009$
$\phi_* = 0.016148976 \pm 0.000000005$
$\alpha = -1.101339351 \pm 0.000000003$

La excesiva precisión que brinda la varianza es artificiosa, pues vendrá dada por el tamaño del paso en la región. Para tener una buena precisión, precisamos de una herramienta que no se anule en esta zona (como el Método de Newton).

7. Ejercicio 6

Realizaremos una cadena y un descenso por el gradiente con las mismas condiciones iniciales, para luego comparar los resultados. Utilizaremos como parámetros iniciales:

$M_* = -20.528749669717794$
$\phi_* = 0.014099700887138394$
$\alpha = -1.1162915383414187$

En el código "Comparación.py" realizamos ambos métodos con estos parámetros. En la figura 14 podemos ver cómo el descenso por el gradiente converge más rápido. No obstante, en la figura 15 vemos cómo es incapaz de muestrear el espacio de parámetros, mostrando un histograma de prácticamente un único valor. Esto sucede dado que cuando se acerca mucho al máximo, simplemente lo rodea con el tamaño del paso, dado que el gradiente se anula en este.

	Descenso por el Gradiente	**Monte Carlo**
$\bar{M}$	-20.7778999842	-20.7777934650
σ_M^2	3.4505950116e-12	6.8296976000e-07
$\bar{\phi}$	0.0161491620809	0.016152055024
σ_ϕ^2	8.6901342177e-14	3.533889352e-10
$\bar{\alpha}$	-1.10133955296	-1.10128901687
σ_α^2	1.0210054525e-13	2.5528229087e-07

Tabla 2: Comparación de medias y varianzas entre Descenso por el Gradiente y Monte Carlo

A pesar de las diferencias mostradas, en la tabla 2 observamos que arrojan valores muy similares para los parámetros. No obstante, las varianzas son muy diferentes, justamente por los problemas que tiene el método de Descenso por el Gradiente: se clava en un valor al anularse el gradiente.

8. Ejercicio 7

Tenemos una moneda cargada, la cual es lanzada 100 veces, obteniendo 60 caras y 40 secas. Para obtener el

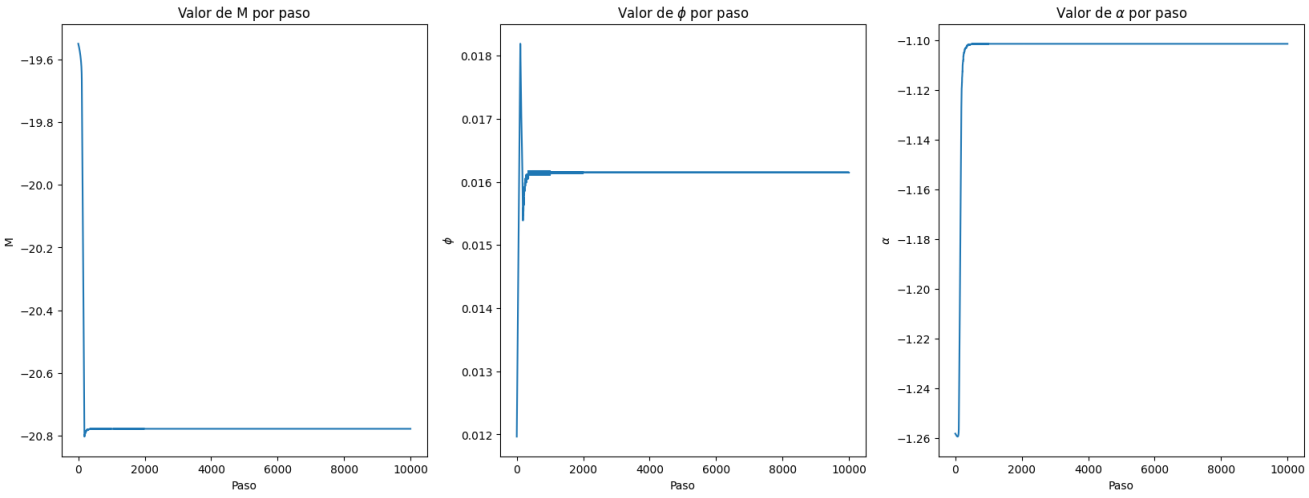

Figura 12: Valor de los parámetros en cada paso al aplicar el Método de Descenso por el Gradiente.

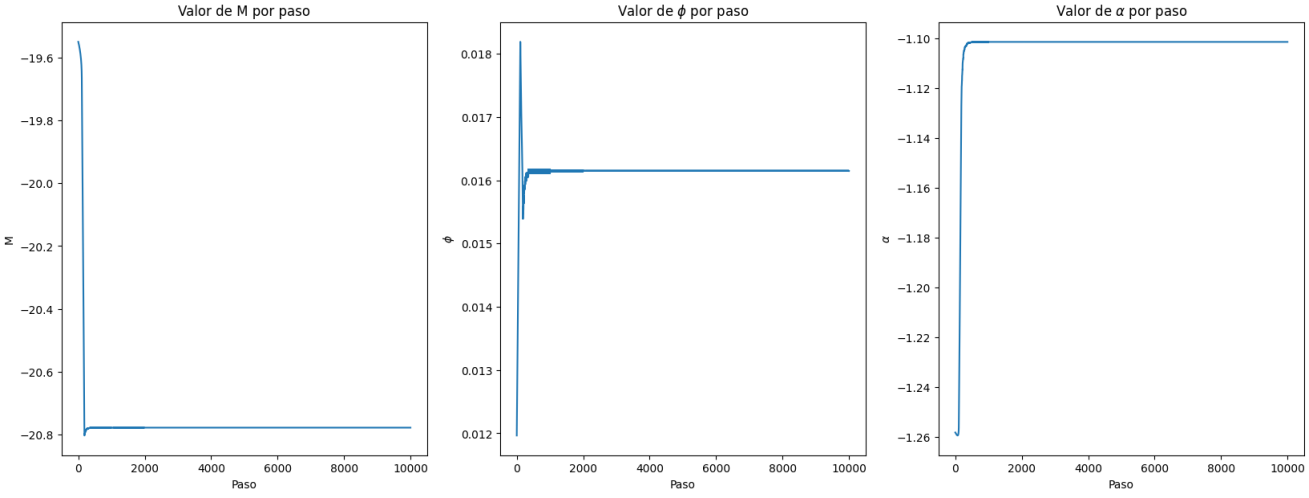

Figura 13: Distribución de los parámetros al aplicar el Método de Descenso por el Gradiente.

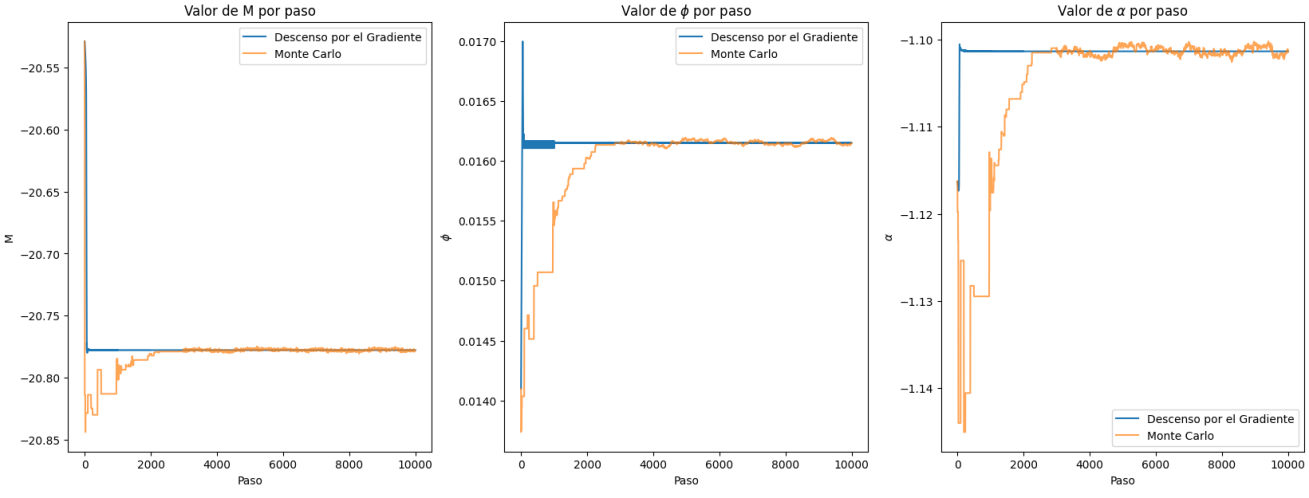

Figura 14: Parámetros según el paso en cada método presentado en este trabajo.

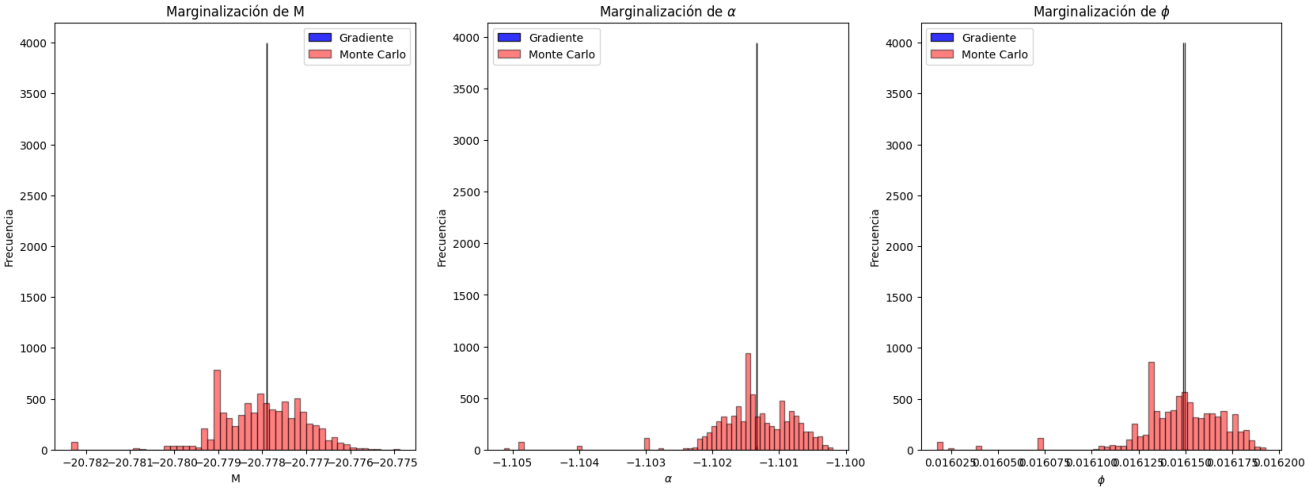

Figura 15: Marginalización de los parámetros mediante los 2 métodos presentados en este trabajo.

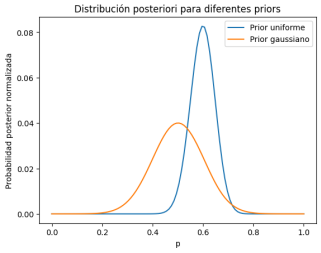

Figura 16: Distribución posteriori de la moneda cargada con distintos priors.

sesgo p, en el código "Ej7.py" computamos la probabilidad a posteriori de dos formas: Con un prior uniforme entre 0 y 1, y con un prior gaussiano de media 0.5 y desviación estándar 1. En la figura 16 comparamos ambas distribuciones. En cada caso, el valor que maximiza la distribución es diferente: Para el prior uniforme, tenemos $p = 0.595959595959596$ y para el prior gaussiano, $p = 0.5050505050505051$. Es decir, con el prior gaussiano no se nota que la moneda está cargada, mientras que con el plano se aprecia muy bien.

9. Ejercicio 8

Vamos a estimar la constante λ de una distribución exponencial mediante un método frecuentista y mediante inferencia bayesiana. Simularemos 50 mediciones de una distribución con $\lambda = 5$, por lo que los estimadores deberían aproximar este valor.

- Frecuentista: Maximizaremos el likelihood, en este caso tenemos una distribución exponencial, por lo tanto:

$$L = \prod_{i=1}^{N} \lambda e^{-\lambda x_i} \tag{9}$$

$$\log L = N \log \lambda - \lambda \sum_{i=1}^{N} x_i \tag{10}$$

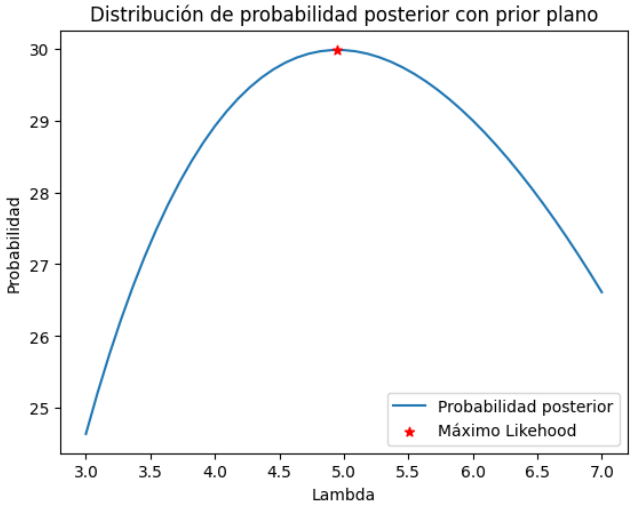

Figura 17: Distribución posteriori con un prior plano, la estrella es la estimación de máximo Likelihood.

$$\frac{\partial \log L}{\partial \lambda} = \frac{N}{\lambda} - \sum_{i=1}^{N} x_i = 0 \tag{11}$$

$$\lambda = \frac{1}{\overline{x}} \tag{12}$$

Por lo tanto, este método será sencillamente calcular la media de los datos generados e invertir.

- Inferencia bayesiana: Tomaremos un prior plano entre 3 y 7 y generaremos una lista de 100 valores entre 1 y 10 para lambda, luego, calcularemos su posteriori y tomaremos como estimador el valor que la maximize.

Tras realizar esto, obtenemos valores próximos (ver figura 17), pero cada vez que repetimos el experimento (simulamos datos diferentes) obtenemos que el mejor estimador es diferente. Es decir, no basta hacerlo una vez como para saber cual método es más preciso. Por lo tanto, repetimos el proceso 10000 veces y consideramos las medias. Los resultados finales son los presentados en

la tabla 3.

Dato	Frecuentista	Bayesiano
Primer Estimación	4.926	4.909
Primer $E_{\%}$	1.470 %	1.818 %
Estimación media	5.098	5.093
$E_{\%}$ de la media	1.970 %	1.872 %

Tabla 3: Comparación entre métodos frecuentistas y bayesianos. $E_{\%}$ es el error porcentual respecto al valor real de $\lambda = 5$.

Podemos observar que el método de inferencia bayesiana es más preciso, aunque por muy poco para este caso.

10. Hessiano

Para comprobar lo dicho durante el trabajo, realizamos una modificación al descenso por el gradiente: tras 5000 pasos, pasamos a utilizar el Hessiano para movernos en el espacio de parámetros. Esto se conoce cómo Método de Newton. Es bueno cerca del máximo, por ello cambiamos de método cuando ya estamos cerca de converger. En las figuras 18 y 19 podemos observar cómo el cambio al Método de Newton nos permite alcanzar mejores precisiones. Por ejemplo, en ϕ y α se observa cómo antes del cambio el gradiente estaba saltando entre 2 puntos alrededor del máximo y gracias al Hessiano pudo converger con mayor precisión.

Además, en la figura 20 podemos observar cómo ya no se clava en un único valor, sino que logra formar una distribución. Esto se desarrolla en el código "Hessiano.py".

11. Conclusiones

En este trabajo comparamos métodos bayesianos y no bayesianos. Encontramos que con el Método de Montecarlo podemos explorar el espacio de parámetros, lo cual nos proporciona información sobre la distribución completa. En cambio, con el método de descenso por el gradiente, vamos rápidamente al valor deseado pero sin explorar el espacio de parámetros. Además, este se anula justo cerca del valor deseado, por lo que tiene una precisión limitada. Esta precisión mejora si cuando estamos muy cerca del máximo (tan cerca como orden menos cinco en las variables normalizadas), cambiamos al Método de Newton. Este consiste en utilizar el Hessiano, el cual no se anula cerca del máximo. Pudimos observar cómo al realizar el cambio el algoritmo deja de rebotar alrededor del máximo y comienza a converger mejor. Destacar que el Método de Newton resultó computacionalmente mucho más caro.

Por otra parte, comparamos resultados frecuentistas y bayesianos para un mismo experimento, encontrando que los métodos bayesianos son más precisos, lo cual es coherente teniendo en cuenta que consideran más factores.

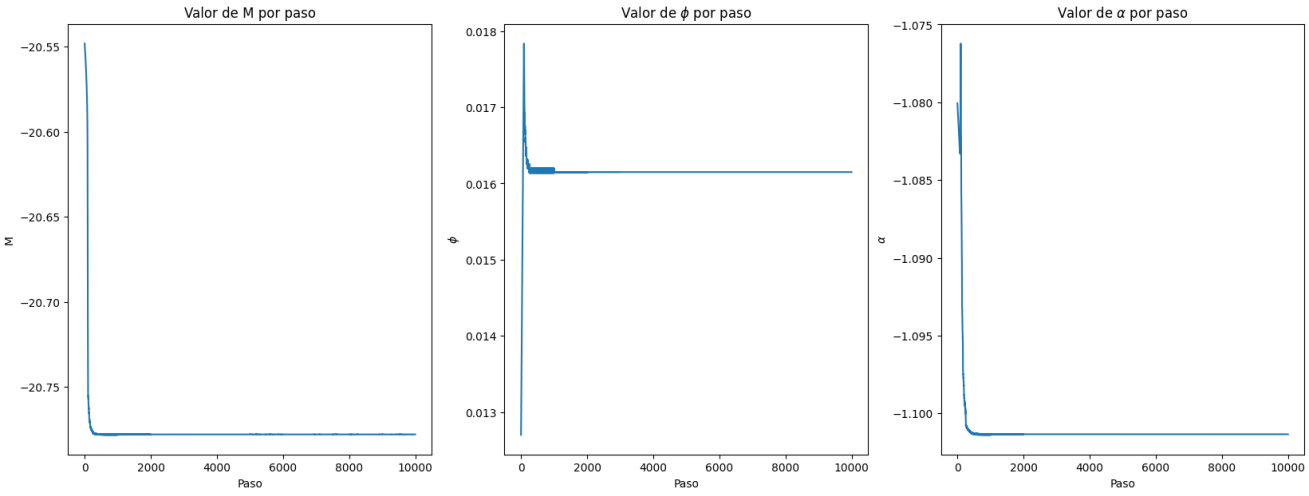

Figura 18: Valor de los parámetros para cada paso, utilizando el Hessiano y el gradiente combinados.

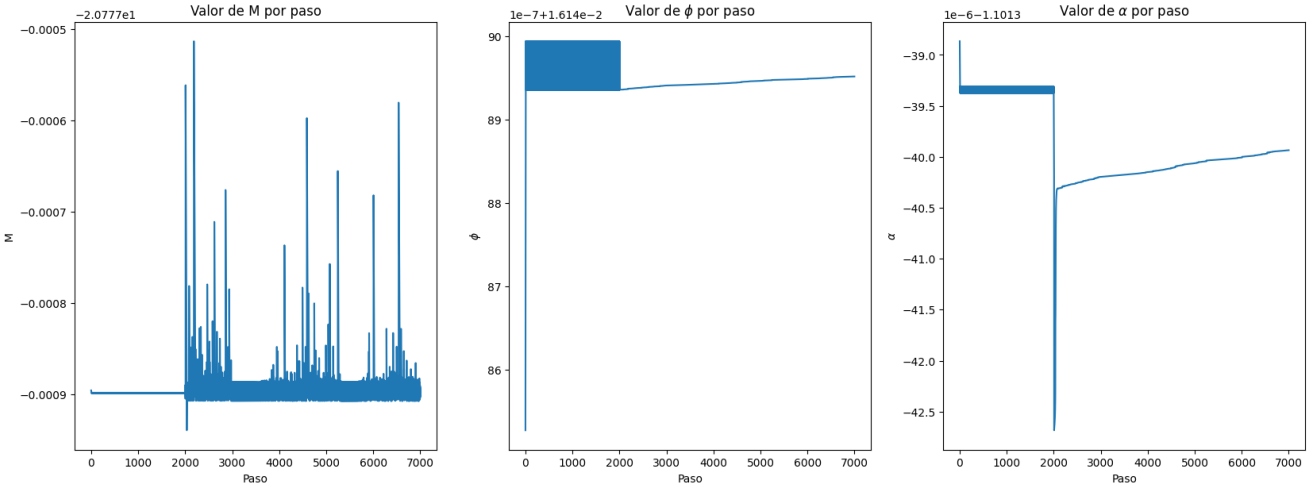

Figura 19: Valor de los parámetros para los últimos 7000 pasos, utilizando el Hessiano y el gradiente combinados.

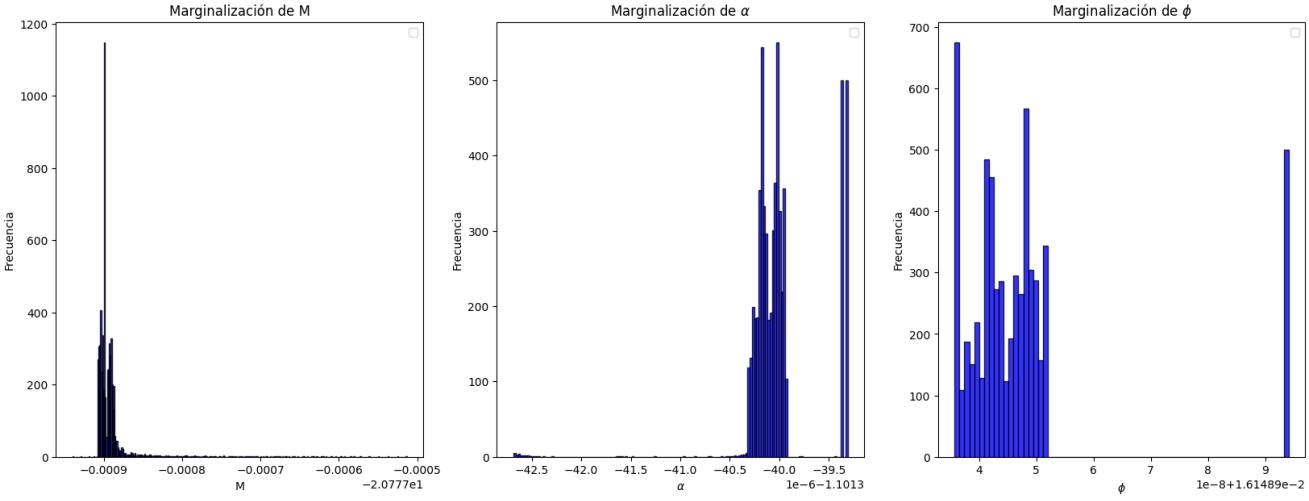

Figura 20: Marginalización de los parámetros calculados en los últimos 6000 pasos.