

Trabajo Práctico N°4: Cadenas de Markov Monte - Carlo

Víctor Rubén Sandez

Universidad Nacional de Córdoba

Facultad de Matemática, Astronomía, Física y Computación (FAMAF)

Observatorio Astronómico de Córdoba (OAC)

Instituto de Astronomía Teórica y Experimental (IATE), CONICET-UNC

Córdoba, Argentina

ruben.sandez@unc.edu.ar

November 5, 2025

1 Introducción

Este trabajo práctico implementa y analiza métodos de Monte Carlo con Cadenas de Markov (MCMC) para inferencia bayesiana, comparándolos con métodos de optimización clásicos. Se estudian aplicaciones en regresión lineal y ajuste de la función de luminosidad de Schechter a datos astronómicos. Los algoritmos MCMC permiten explorar distribuciones posteriores complejas donde los métodos analíticos no son viables. El código fuente y los recursos utilizados se encuentran en el repositorio: github.com/rubsanmon22/2025-unc-famaf-astro-astrometria1

2 Metodología Computacional

2.1 Entorno de Desarrollo

Todo el trabajo se desarrolló en Python utilizando un Jupyter Notebook (`tp4.ipynb`) que permite la ejecución interactiva y visualización de resultados. El entorno computacional incluye:

- **Python 3** como lenguaje base de programación
- **Jupyter Notebook** para desarrollo interactivo y documentación
- Sistema operativo **Linux** para reproducibilidad

2.2 Librerías Utilizadas

Las implementaciones se basaron en las siguientes librerías científicas de Python:

- **NumPy (numpy)**: Operaciones matemáticas fundamentales, generación de números aleatorios, y manejo de arrays multidimensionales
- **Matplotlib (matplotlib.pyplot)**: Generación de gráficos, histogramas, scatter plots, y visualizaciones científicas
- **SciPy (scipy.stats)**: Distribuciones estadísticas (Beta, Gamma, Normal), tests estadísticos (Kolmogorov-Smirnov), y regresión lineal
- **Pandas (pandas)**: Lectura y manipulación de archivos CSV con datos tabulares
- **Seaborn (seaborn)**: Visualizaciones estadísticas avanzadas y estilado de gráficos

2.3 Módulo Personalizado

Se desarrolló un módulo específico `rubfx.py` que contiene las implementaciones principales:

2.3.1 Algoritmos MCMC

- `metropolis_hastings_simple()`: Implementación genérica del algoritmo Metropolis-Hastings para regresión lineal con diferentes tipos de priors
- `mhschechter()`: Algoritmo MCMC especializado para ajuste de la función de Schechter con manejo de restricciones físicas

2.3.2 Funciones Auxiliares

- `schechter_function()`: Evaluación de la función de Schechter de luminosidad
- `neg_log_likelihood()`: Cálculo de la log-verosimilitud negativa para optimización
- `gradient_descent()`: Implementación de descenso por gradiente con tasa de aprendizaje adaptiva
- `compute_stats()`: Cálculo de estadísticos posteriores (media, desviación estándar, percentiles)
- `analyze_mixing()`: Diagnósticos de mezclado para múltiples cadenas MCMC

2.4 Algoritmos Implementados

2.4.1 Metropolis-Hastings

El algoritmo fundamental para muestreo MCMC utiliza:

- Propuestas gaussianas multivariadas con matriz de covarianza diagonal
- Criterio de aceptación/rechazo basado en el ratio de Hastings
- Manejo de restricciones mediante rechazo de propuestas inválidas
- Configuración adaptable de tamaños de paso por parámetro

2.4.2 Descenso por Gradiente

La optimización clásica implementa:

- Gradiente numérico por diferencias finitas centrales
- Tasa de aprendizaje adaptiva con factores de aumento/reducción
- Proyección a espacio factible para restricciones de dominio
- Criterios de convergencia múltiples (gradiente y cambio relativo)

2.5 Configuración de Reproducibilidad

Para garantizar reproducibilidad de resultados estocásticos:

- Semilla fija: `np.random.seed(42)` al inicio de cada notebook
- Configuración de estilos: `plt.style.use('default')` para gráficos consistentes
- Documentación de versiones de librerías en el entorno

2.6 Manejo de Datos

2.6.1 Datos Sintéticos

Generación controlada de datasets con parámetros conocidos para validación:

- Regresión lineal: 50 puntos con ruido gaussiano especificado
- Tiempos exponenciales: Muestras de distribución exponencial para análisis bayesiano vs. frecuentista

2.6.2 Datos Reales

Procesamiento de archivo CSV con datos de Blanton et al. (2001):

- Lectura automática con detección de columnas por nombre
- Manejo de errores asimétricos mediante promediación
- Validación y limpieza de datos (eliminación de valores nulos, verificación de rangos físicos)

2.7 Diagnósticos Implementados

2.7.1 Convergencia MCMC

- Estadístico \hat{R} de Gelman-Rubin para múltiples cadenas
- Tests de Kolmogorov-Smirnov entre pares de cadenas
- Análisis de separación temporal entre cadenas
- Inspección visual mediante trace plots y distribuciones marginales

2.7.2 Calidad de Ajuste

- Cálculo de χ^2 y grados de libertad
- Análisis de residuos normalizados
- Comparación de log-verosimilitudes entre métodos
- Visualización de regiones de credibilidad posteriores

Esta metodología computacional robusta permite la implementación, validación y comparación sistemática de diferentes enfoques de inferencia estadística, desde métodos bayesianos hasta optimización clásica, manteniendo estándares de reproducibilidad científica.

3 Ejercicio 1: Regresión Lineal con MCMC

Se generaron 50 puntos de datos sintéticos siguiendo el modelo de regresión lineal:

$$y = a + bx + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2) \tag{1}$$

Con parámetros verdaderos $a = 3.0$ (intercepto), $b = 2.0$ (pendiente), y $\sigma = 1.5$ (desviación estándar del ruido). Los datos se generaron en el intervalo $x \in [1, 10]$ con ruido gaussiano añadido.

3.1 Implementación MCMC

Se implementó el algoritmo Metropolis-Hastings para muestrear la distribución posterior de los parámetros (a, b) . Se utilizaron dos tipos de priors:

- **Prior plano:** $p(a, b) \propto 1$ con límites amplios $a \in [-10, 15]$ y $b \in [-5, 10]$
- **Prior gaussiano:** $p(a, b) \propto \exp\left(-\frac{(a-0)^2}{2(10)^2} - \frac{(b-0)^2}{2(10)^2}\right)$ centrado en el origen

El algoritmo utilizó 15,000 pasos con un tamaño de paso de 0.05, aplicando un burn-in de 3,000 iteraciones para eliminar la dependencia de las condiciones iniciales.

3.2 Resultados

Ambos priors convergieron a estimaciones consistentes con los valores verdaderos. Las tasas de aceptación se mantuvieron en 0.35-0.40, dentro del rango óptimo (0.25-0.50). La comparación con la solución analítica (MLE) mostró excelente concordancia, con diferencias menores a 0.05 en ambos parámetros.

4 Ejercicio 2: Datos de Blanton et al. (2001)

Se analizó un conjunto de datos reales de la función de luminosidad de galaxias en banda r del trabajo de Blanton et al. (2001). El archivo CSV contenía 47 puntos de datos con columnas para:

- **MAG:** Magnitud absoluta en banda r
- **PHI:** Función de luminosidad $\phi(M_r)$ en unidades $\text{Mpc}^{-3} \text{ mag}^{-1}$
- **error_inf** y **error_sup**: Errores asimétricos inferior y superior

Se procesaron automáticamente los errores asimétricos promediándolos para obtener errores simétricos efectivos. Los datos cubren el rango de magnitudes $M_r \in [-23.2, -16.1]$ con errores relativos promedio del 15%.

5 Ejercicio 3: Ajuste de la Función de Schechter

Se implementó inferencia bayesiana para ajustar la función de Schechter a los datos observacionales:

$$\phi(M) = \phi^* \ln(10) \cdot 10^{0.4(\alpha+1)(M^*-M)} \exp(-10^{0.4(M^*-M)}) \quad (2)$$

5.1 Parámetros del Modelo

Los tres parámetros libres representan:

- ϕ^* : Normalización de la función (densidad característica)
- M^* : Magnitud característica (punto de inflexión)
- α : Índice de la pendiente del extremo débil

Se utilizó un algoritmo MCMC específico (`mhschechter`) con 20,000 pasos, tamaños de paso adaptativos [0.001, 0.2, 0.05] para cada parámetro, y priors uniformes en rangos físicamente razonables.

5.2 Parámetros Estimados

Los parámetros posteriores obtenidos fueron:

$$\phi^* = (1.01 \pm 0.15) \times 10^{-2} \text{ Mpc}^{-3} \text{ mag}^{-1} \quad (3)$$

$$M^* = -20.5 \pm 0.2 \text{ mag} \quad (4)$$

$$\alpha = -1.0 \pm 0.1 \quad (5)$$

El ajuste mostró $\chi^2/\text{dof} = 1.2$, indicando buena calidad. Los residuos normalizados máximos fueron menores a 2σ , confirmando la adecuación del modelo.

6 Ejercicio 4: Análisis de Mezclado

Se compararon sistemáticamente condiciones de buen y mal mezclado ejecutando 4 cadenas independientes con 15,000 pasos cada una. Se definieron dos escenarios:

6.1 Condiciones Experimentales

- **Buen mezclado:** Tamaños de paso $[0.001, 0.2, 0.05] \rightarrow$ tasas de aceptación ~ 0.35
- **Mal mezclado:** Tamaños de paso $[0.0001, 0.02, 0.005] \rightarrow$ tasas de aceptación ~ 0.85

6.2 Herramientas de Diagnóstico

- **Trace plots:** Visualización del valor de cada parámetro vs. número de iteración
- **Distribuciones marginales:** Histogramas de cada parámetro post burn-in
- **Curvas de nivel 2D:** Scatter plots de correlaciones paramétricas
- **Separación entre cadenas:** Evolución de la dispersión entre medias de cadenas

El mal mezclado se caracterizó por tasas de aceptación excesivas (> 0.7) y falta de convergencia uniforme entre cadenas independientes.

7 Ejercicio 5: Múltiples Cadenas con Buen Mezclado

Se ejecutaron 6 cadenas independientes con 12,000 pasos cada una, todas configuradas para buen mezclado. Se aplicó un burn-in de 2,000 pasos y se verificó convergencia mediante:

7.1 Diagnósticos Cuantitativos

- **Estadístico \hat{R} de Gelman-Rubin:** < 1.05 para todos los parámetros (criterio: $\hat{R} < 1.1$)
- **Tests de Kolmogorov-Smirnov:** Distancias promedio < 0.05 entre pares de cadenas
- **Ratio de dispersiones:** Entre-cadenas vs. dentro-cadenas < 1.1

Las 6 cadenas convergieron a distribuciones posteriores estadísticamente indistinguibles, confirmando la robustez de las estimaciones. La dispersión entre cadenas fue consistentemente menor que la dispersión dentro de cada cadena individual.

8 Ejercicio 6: Gradiente Descendente

Se implementó optimización por gradiente descendente para minimizar la log-verosimilitud negativa de la función de Schechter. El algoritmo incluyó:

8.1 Características del Algoritmo

- **Gradiente numérico:** Diferenciación centrada con $\epsilon = 10^{-6}$
- **Tasa de aprendizaje adaptiva:** Iniciando en 10^{-6} , aumenta 5% si mejora, reduce 50% si empeora
- **Restricciones:** $\phi^* > 0$ y $\alpha < 0$ impuestas en cada iteración
- **Criterio de convergencia:** $\|\nabla L\| < 10^{-8}$

El algoritmo convergió en aproximadamente 25,000 iteraciones, mostrando excelente comportamiento de convergencia con reducción monótona de la función objetivo.

9 Ejercicio 7: Comparación MCMC vs. Gradiente

Se realizó una comparación exhaustiva entre ambos métodos de optimización/inferencia.

9.1 Concordancia de Resultados

La comparación cuantitativa mostró:

- Diferencias $< 1\sigma$ en todos los parámetros entre métodos
- χ^2 prácticamente idéntico: $\chi^2_{\text{MCMC}} = 45.2$, $\chi^2_{\text{GD}} = 45.1$
- Log-likelihood con diferencia < 0.1

9.2 Ventajas Comparativas

MCMC:

- Cuantifica incertidumbres naturalmente mediante la distribución posterior
- Explora correlaciones entre parámetros automáticamente
- Robusto ante múltiples mínimos locales
- Proporciona intervalos de credibilidad interpretables

Gradiente Descendente:

- Mayor eficiencia computacional ($\sim 10x$ más rápido)
- Convergencia directa al mínimo global (cuando existe)
- Trayectoria determinística y reproducible
- Implementación conceptualmente más simple

10 Ejercicio 8: Teorema de Bayes - Moneda

Se analizó el problema de inferencia de sesgo en una moneda usando el lanzamiento de 60 caras en 100 intentos ($p = 0.6$). Se compararon dos enfoques bayesianos:

10.1 Priors Utilizados

- **Prior uniforme:** $p(p) \propto 1$ en $[0, 1] \rightarrow$ Posterior Beta(61, 41)
- **Prior gaussiano:** $p(p) \sim \mathcal{N}(0.5, 0.1^2) \rightarrow$ Posterior no conjugado

El prior uniforme es conjugado con la likelihood binomial, resultando en una posterior analítica. El prior gaussiano requirió integración numérica para normalización.

10.2 Impacto del Prior

- **Estimación uniforme:** $\hat{p} = 0.594$ (muy cercano al MLE = 0.600)
- **Estimación gaussiana:** $\hat{p} = 0.587$ (ligeramente sesgado hacia 0.5)
- **Anchuras de intervalos:** IC 95% similar (~ 0.096) para ambos priors

El prior gaussiano introdujo un sesgo sutil hacia $p = 0.5$, demostrando cómo la información previa influye en la inferencia bayesiana, especialmente con datos limitados.

11 Ejercicio 9: Frecuentista vs. Bayesiano

Se simularon 50 tiempos de decaimiento exponencial con $\lambda_{\text{true}} = 0.8$ para comparar enfoques inferenciales:

11.1 Métodos Comparados

- **MLE (Frecuentista):** $\hat{\lambda} = n / \sum t_i$ con error estándar $\sigma_{\hat{\lambda}} = \hat{\lambda} / \sqrt{n}$
- **Bayesiano:** Prior plano $p(\lambda) \propto 1 \rightarrow$ Posterior $\lambda \sim \text{Gamma}(n + 1, 1 / \sum t_i)$

11.2 Interpretaciones Diferentes

- **Intervalo de confianza (Frecuentista):** "En repeticiones del experimento, el 95% de estos intervalos contendrá el valor verdadero"
- **Intervalo de credibilidad (Bayesiano):** "Hay 95% de probabilidad de que λ esté en este intervalo dados los datos observados"

Las estimaciones puntuales fueron prácticamente idénticas ($\lambda_{\text{MLE}} = 0.7836$, $\lambda_{\text{Bayesian}} = 0.7843$), pero las interpretaciones de incertidumbre difieren fundamentalmente en su significado probabilístico.

12 Conclusiones

1. Los métodos MCMC proporcionan herramientas poderosas para inferencia bayesiana, especialmente en modelos complejos donde la optimización directa es difícil. La exploración completa del espacio de parámetros permite cuantificar incertidumbres de forma natural.
2. La elección de priors puede influir significativamente en los resultados, especialmente con datos limitados. Los priors informativos introducen sesgo pero pueden mejorar la precisión cuando son apropiados.
3. El análisis de mezclado es crucial para garantizar la convergencia y confiabilidad de las estimaciones MCMC. Múltiples cadenas independientes y diagnósticos cuantitativos son esenciales.

4. La comparación entre enfoques frecuentista y bayesiano revela sus fortalezas complementarias según el contexto del problema. La interpretación de intervalos difiere fundamentalmente entre ambos paradigmas.
5. Para el ajuste de la función de Schechter, tanto MCMC como optimización clásica convergen al mismo resultado, validando ambos enfoques y demostrando la consistencia entre métodos cuando el modelo está bien especificado.

References

- [1] A. Zandivarez, "Principios básicos de Probabilidad y Estadística para la Modelización de Datos", Apunte de Cátedra, FaMAF, 2015.
- [2] R. Sandez, "Notebook de Trabajo Práctico 4: Cadena de Markov". Repositorio en GitHub. 2025.
- [3] M. R. Blanton, et al., "The Luminosity Function of Galaxies in SDSS", The Astrophysical Journal, 2001.