

Measurement of matter–antimatter differences in beauty baryon decays

Universidad de Guanajuato
División de Ciencias e Ingenierías
Laboratorio de Óptica
Loma del Bosque 103, Lomas del Campestre
C. P. 37150
León, Guanajuato
México

Moreno López, Noemí Lizbeth (e-mail: morenoln2016@licifug.ugto.mx)

Resumen

Se construyó el algoritmo de **Markov Chain Monte Carlo** con el propósito de determinar la función que corresponde a cualquier conjunto de datos, el algoritmo es el mismo pero debe ser adaptado a cada tipo de regresión.

Se usó el algoritmo para modelar un conjunto de datos, para esto se separaron los datos en tres regiones y el resultado final fue la unión de la regresión obtenida en cada región.

-A. Introducción

Algunas de las **diferencias entre la materia, antimateria** se observan en los decaimientos de mesones, tales como **K, B** pero no en decaimientos de **bariones**. [2]

¿Por qué con algunas partículas es posible distinguir las diferencias entre materia, antimateria y con otras no?

Esta diferencia se atribuye principalmente a la no invariancia de interacciones fundamentales sujetas a condiciones específicas puestas a prueba previamente con mesones.

La asimetría entre materia-antimateria se relaciona con la violación **CP**, donde **C, P** son operadores que representan **carga, paridad**. El mecanismo de **Cabibbo–Kobayashi–Maskawa (CKM)** describe las transiciones entre los quarks arriba, abajo (en general).

Este documento trata del decaimiento $\Lambda_b^0 \rightarrow p\pi^-\pi^+\pi^-$, mediado por interacción débil (importante a la hora de determinar amplitudes). El propósito es analizar la **distribución de masa invariante** en el decaimiento $\Lambda_b^0 \rightarrow p\pi^-\pi^+\pi^-$ para observar los rendimientos de la señal obtenida.

-B. MCMC

Comenzando con el **Teorema de Bayes**, para un espacio \mathcal{S} , con $\mathcal{B}_i \forall i \in \mathbb{Z}$, como particiones de \mathcal{S} , de tal forma que $\cup_i \mathcal{B}_i$ y $\mathcal{B}_i \cap \mathcal{B}_j = \emptyset \forall i \neq j$ entonces

Theorem 1. (Teorema de Bayes). Sea A cualquier evento. Entonces para cualquier $1 \leq k \leq K$ se tiene

$$P(\mathcal{B}_k|A) = \frac{P(A|\mathcal{B}_k)P(\mathcal{B}_k)}{P(A)} = \frac{P(A|\mathcal{B}_k)P(\mathcal{B}_k)}{\sum_{j=1}^K P(A|\mathcal{B}_j)P(\mathcal{B}_j)}, \quad (1)$$

-B1. Distribuciones del prior, posterior: Sea θ un vector de parámetros desconocido de interés. Se asume que θ tiene una distribución específica $\pi(\theta)$. Esta es la **distribución prior**, además sea \mathbf{X} con distribución de probabilidad $p(\mathbf{x}|\theta)$, este es el **likelihood**. La distribución conjunta de estos dos parámetros está dada por:

$$p(\theta, \mathbf{x}) = \pi(\theta)p(\mathbf{x}|\theta), \quad (2)$$

para obtener la distribución marginal de \mathbf{X} se integra de la siguiente manera:

$$p(\mathbf{x}) = \int_{\theta} \pi(\theta)p(\mathbf{x}|\theta)d\theta,$$

y haciendo uso del **Teorema de Bayes** se obtiene la siguiente relación, que representa la **distribución del posterior**:

$$\pi(\theta|\mathbf{x}) = \frac{\pi(\theta)p(\mathbf{x}|\theta)}{\int_{\theta} \pi(\theta)p(\mathbf{x}|\theta)d\theta}, \quad (3)$$

la idea central es pensar en que se cumple al siguiente relación

$$\pi(\theta|\mathbf{x}) \propto \pi(\theta)p(\mathbf{x}|\theta),$$

donde se obtiene información acerca de la distribución del posterior a partir del vector de parámetros θ y de \mathbf{X} .

Los algoritmos de **MCMC** aparecen en la década de **1940** en Los Alamos. estaban interesados en modelar el estado probabilístico de colecciones de partículas atómicas. No se contaba con solución analítica, pero sí numérica. Ulam y Metrópolis construyeron una cadena de Markov para la cual la distribución deseada era la distribución estacionaria de la cadena de Markov. Entonces solo necesitaban simular la cadena de Markov hasta que se lograba la estacionariedad. Con este fin, introdujeron el algoritmo **Metropolis** y su impacto fue enorme. Posteriormente, **MCMC** se introdujo en las estadísticas y se generalizó con el algoritmo **Metrópolis-Hastings (1970)**.

Una **cadena de Markov** es una secuencia de variables random $\{X_1, X_2, \dots, X_N\}$ es un espacio discreto Ω , es de **primer orden** si

$$\begin{aligned} p(X_N = x_N | X_{N-1} = x_{N-1}, \dots, X_1 = x_1) \\ = p(X_N = x_N | X_{N-1} = x_{N-1}), \end{aligned}$$

La idea principal del algoritmo de Metrópolis es construir una cadena de Markov reversible en el tiempo con distribuciones límite.

La idea general de la construcción del algoritmo se muestra a continuación:

Algoritmo Metrópolis-Hastings:

```

Inicializa  $x^{(0)} \sim q(x)$ ,
for iteración  $i$  ( $i \in \mathbb{Z}$ ) do
  Proponer  $x^{(can)} \sim q(x^{(i)}|x^{(i-1)})$ ,
  Probabilidad de aceptación:
   $\alpha(x^{(can)}|x^{(i-1)}) = \min \left\{ 1, \frac{q(x^{(i-1)}|x^{(can)})\pi(x^{(can)})}{q(x^{(can)}|x^{(i-1)})\pi(x^{(i-1)})} \right\}$ 
   $u \sim \text{Uniforme}(u; 0, 1)$ 
  if  $u < \alpha$  entonces:
    Aceptación de propuesta:  $x^{(can)} \rightarrow x^{(i)}$ 
  else
    Rechazo de la propuesta:  $x^{(i-1)} \rightarrow x^{(i)}$ 
  fin for

```

Para este análisis se consideró una **distribución gaussiana** en los datos, de tal forma que haciendo uso de **Python**, se construyó el algoritmo de **Metrópolis-Hastings** [1]. Es importante que antes de cualquier cosa, se observen los datos con los que se trabajará, a continuación se muestra una figura con los datos exportados:

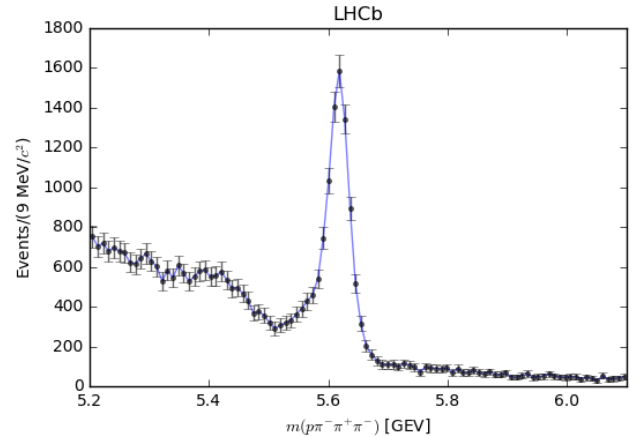


Figura 1. Muestra gráfica de los datos exportados.

Para este modelo se identificaron tres áreas principales, las cuales se muestran en las figuras (2), (3), (4):

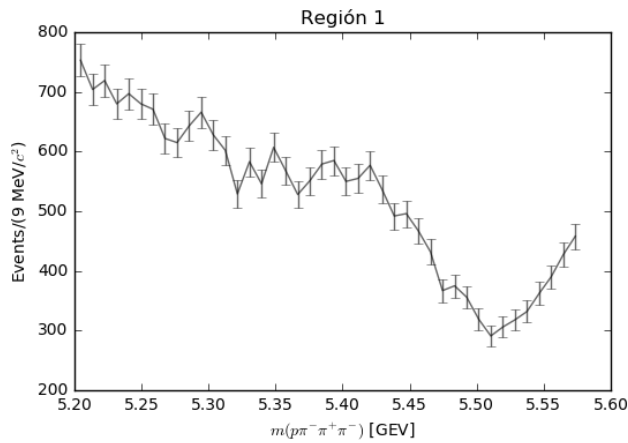


Figura 2. Figura que representa la **región 1**.

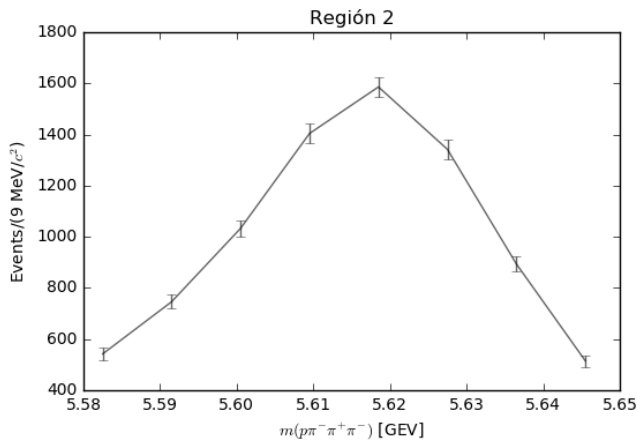


Figura 3. Figura que representa la **región 2**.

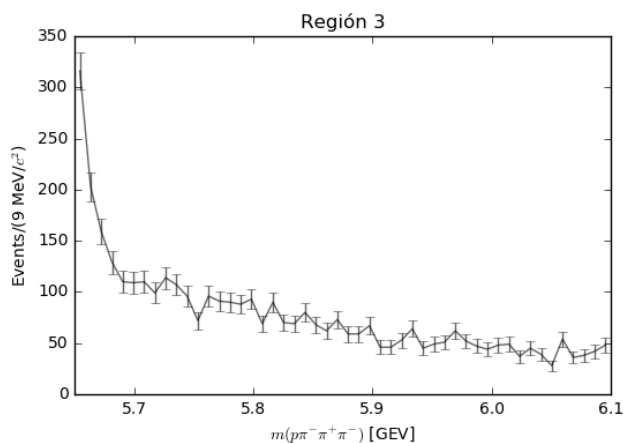


Figura 4. Figura que representa la **región 3**.

Se trabajó con cada región individualmente, i. e., una regresión por región que al final se unificó

para construir una función general. Se consideró que las **regiones 1, 3** correspondían a una **regresión lineal**, mientras que la **región 2** correspondía a una **regresión gaussiana**.

El algoritmo se dividió en métodos o partes que componen a **Metrópolis-Hastings**:

1. Obtención del valor inicial de los parámetros, propuesto por el usuario, se colocan estos valores en la entrada de la función.
2. Algoritmo de **Metrópolis-Hastings**.
3. Análisis de convergencia, identificando la región en la que convergen los parámetros e individualmente para observar si el valor **converge o diverge**.
4. Se desecha la mitad del arreglo donde se guardan los valores obtenidos de los parámetros, la primera mitad se desecha dado que el parámetro comienza a converger y los valores antes de esta mitad afectarían valores de interés tales como la **media, mediana**.
5. Una vez obtenida la **media** de los parámetros, se procede a graficar los datos con el ajuste obtenido para observar qué tan aceptable es.
6. Los valores obtenidos se comparan con la librería **scipy** para observar la calidad de cada uno de los ajustes (scipy, Metrópolis-Hastings).

A continuación se muestran las regresiones hechas para las rectas y la unión de estas funciones:

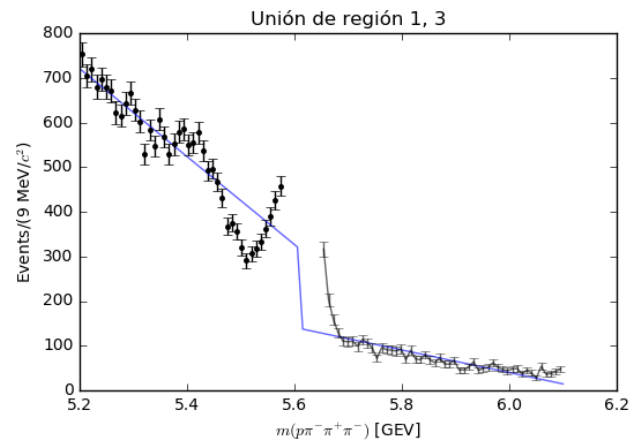


Figura 5. Unión de las funciones para la región 1, 3.

Posteriormente se realizó la regresión para la región 2, se graficó el resultado, se muestra en la siguiente figura una de las pruebas realizadas al modificar el valor de los parámetros de entrada:

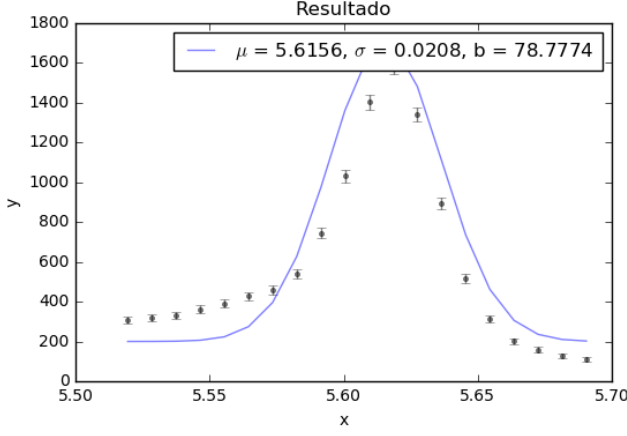


Figura 6. Unión de los resultados obtenidos, las funciones se aplican a su respectiva región.

Cabe destacar que para esta región sí fue muy necesario el uso de un **prior**, dado que se observaba que el valor de los parámetros divergía rápidamente, se acotó una zona en la que se requería encontrar el valor real, pero sin embargo no se puede constar que se trata del mejor valor, puesto que la convergencia se observa que se apega a los límites superiores, i. e., si los parámetros pudieran crecer más, lo harían.

Para la región 2, se hizo uso de una traslación que no estuvo sujeta a ajuste, esto se debió a que al momento de tratar de implementarla al modelo, los valores divergían rápidamente y no se encontraba una zona de convergencia. Una de las modificaciones a este trabajo sería tratar de añadir dicha traslación al modelo, en este caso se usó sólo un valor propuesto por el usuario.

Al final, se construyó una función de la forma:

$$f(x) = \begin{cases} a_1x + b_1 & \text{si } x < 5.5195 \\ \frac{1}{\sqrt{2\pi\sigma^2}} e^{(x-\mu)^2/2\sigma^2} & \text{si } 5.5195 \leq x \leq 5.6995 \\ a_3x + b_3 & \text{si } x > 5.6995 \end{cases} \quad (4)$$

donde los índices indican la región a la cual trata la regresión, a continuación se muestra el valor de los parámetros obtenido:

Región	Valor de los parámetros		
Región 1	$a_1 = -1082.54$	$b_1 = 6351.30$	
Región 2	$\mu = 5.615$	$\sigma = 0.0208$	$amp = 78.73$
Región 3	$a_3 = -278.00$	$b_3 = 1709.03$	

donde se obtuvo gráficamente la siguiente solución a la regresión, colocando máscaras en regiones

y posteriormente uniendo los resultados:

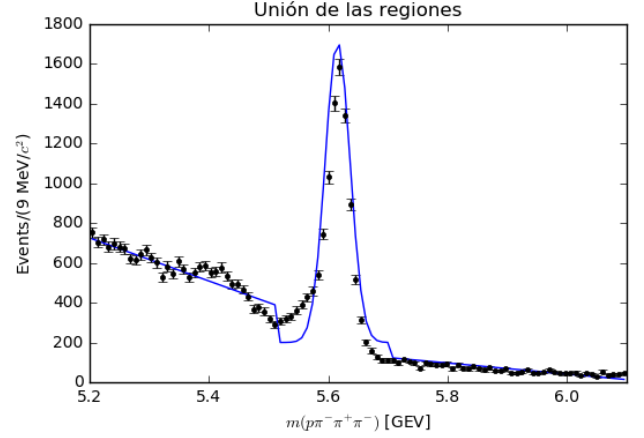


Figura 7. Resultado obtenido con las regresiones cada una de las regiones.

Se observa que las regresiones se traslapan con las barras de error de los datos, lo cual es una buena señal dado que resulta ser consistente el análisis con las observaciones en algunas regiones, pero el mayor problema consta de la traslación hecha no sujeta a la regresión.

También se probó unir las regiones 1, 3 para ajustar a una exponencial pero se observó que los parámetros divergían de una forma más evidente que de lo que trataba con la gaussiana, los detalles se muestran en [1].

□

I. CONCLUSIONES

Se realizó el algoritmo de **Metrópolis-Hastings** con el fin de ajustar un modelo que no parecía a alguna función particular, si no más bien, se componía de regiones a las que podía hacerseles regresión individualmente y posteriormente unificar para producir una sola.

Se tuvieron algunos problemas como ajustar una traslación al modelo, además se observó que el algoritmo puede hacerse mucho más eficiente si se incrementa el número de regiones o si se incrementa el grado del polinomio que corresponde a las regiones 1, 3. Este proceso no se realizó, implica un mayor trabajo dado que son más parámetros de entrada, esto implica un análisis de convergencia de cada parámetro y una variación de más parámetros de entrada para poder ajustar el modelo correctamente, o por lo menos aceptablemente. Aún así sería

un buen ejercicio intentar con más posibilidades y sobre todo en otra forma de unir las funciones que corresponden a las regiones.

REFERENCIAS

- [1] MORENO LÓPEZ, NOEMÍ LIZBETH. **Measurement of matter–antimatter differences in beauty baryon decays.** https://github.com/DCIDA2019/da2019-MIMIMUSTAINED/blob/master/Proyectos/Proyecto_2/Proyecto.ipynb
- [2] THE LHCb COLLABORATION. *Measurement of matter–antimatter differences in beauty baryon decays.* Nature Physics. <https://www.nature.com/articles/nphys4021.pdf>
- [3] THE LHCb COLLABORATION. HEPData : <https://www.hepdata.net/record/76921?version=2>
- [4] JAVIER DUARTE. *CMS OPEN DATAFORMACHINE LEARNING - JET DATASET.* DATA SCIENCE @ HIGH ENERGY PHYSICS 2017. Fermilab. BATAVIA, IL, USA. <https://indico.fnal.gov/event/13497/session/6/contribution/54/material/slides/0.pdf>
- [5] INTRODUCTION TO SCIPY: FITTING DATA. https://astrofrog.github.io/py4sci/_static/15.%20Fitting%20models%20to%20data.html
- [6] MARTIN HAUGH. *MCMC and Bayesian Modeling.* Columbia University. 2017. http://www.columbia.edu/~mh2078/MachineLearningORFE/MCMC_Bayes.pdf
- [7] DAVID A. VAN DYK. *Markov Chain Monte Carlo.* Statistics Section, Imperial College London Smithsonian Astrophysical Observatory, March 2014. https://hea-www.harvard.edu/AstroStat/Stat310_1314/dvd_20140328.pdf