



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

---

---

FACULTAD DE CIENCIAS

INFERENCIA ESTADÍSTICA PARA EL MODELO DE  
DIFUSIÓN MARKOV-MODULADO CON SALTOS Y  
APLICACIÓN A FINANZAS CUANTITATIVAS

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

MATEMÁTICO APLICADO

P R E S E N T A :

BOR GERARDO REYNOSO MANCERA

TUTOR

FERNANDO BALTAZAR LARIOS



CIUDAD UNIVERSITARIA, CD. MX. 2022



Universidad Nacional  
Autónoma de México



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.



*«Si quieres ir rápido, ve solo. Si quieres llegar lejos, ve acompañado.»*

Proverbio Africano



# Agradecimientos

A mis padres, a mi hermana y a Sandra por todo el apoyo, la paciencia y comprensión que hasta el día de hoy me siguen brindando.

A mis tutores Laura y Fernando por su guía y apoyo en la realización de esta tesis.

A mis sinodales por sus numerosas correcciones y la ayuda que me proporcionaron para la finalización del documento.

A la UNAM, la Facultad de Ciencias y a mis profesores por todas sus enseñanzas.

Investigación realizada gracias al Programa de Apoyo a Proyectos de Investigación e Innovación Tecnológica (PAPIIT) de la UNAM, con número TA100820. Agradezco a la DGAPA-UNAM la beca recibida.



# Índice general

<b>Agradecimientos</b>	<b>v</b>
<b>1. Introducción</b>	<b>1</b>
<b>2. Preliminares</b>	<b>5</b>
2.1. Definiciones Básicas . . . . .	5
2.2. Cadenas de Markov . . . . .	6
2.3. Procesos de Saltos de Markov . . . . .	8
2.3.1. Distribuciones Tipo Fase . . . . .	12
2.4. Movimiento Browniano . . . . .	13
2.5. Integración Estocástica . . . . .	15
2.6. Ecuaciones Diferenciales Estocásticas . . . . .	17
<b>3. Simulación e Inferencia para PSM y EDE</b>	<b>21</b>
3.1. Simulación de Procesos de Saltos de Markov . . . . .	21
3.1.1. PSM Estándar . . . . .	22
3.1.2. PSM con Estados Absorbentes . . . . .	23
3.1.3. Puentes Markovianos . . . . .	26
3.2. Estimación del Generador de un Proceso de Saltos de Markov . . . . .	35
3.2.1. Estimación con Información Completa . . . . .	35
3.2.2. Estimación con Información Incompleta . . . . .	38
3.3. Simulación de Ecuaciones Diferenciales Estocásticas . . . . .	45
3.3.1. Simulación del Movimiento Browniano . . . . .	45
3.3.2. Simulación de EDE Vía su Solución . . . . .	46
3.3.3. Simulación de EDE Vía su Discretización . . . . .	48
3.4. Estimación de Parámetros para EDE . . . . .	51
<b>4. Difusión Markov-Modulada con Saltos</b>	<b>55</b>
4.1. Difusiones de Itô en las Finanzas Cuantitativas . . . . .	55
4.1.1. Consideraciones acerca del MBG . . . . .	57
4.2. El Modelo de Merton . . . . .	58
4.2.1. Desventajas . . . . .	60

4.3. Modelo de Difusión Markov-Modulado con Saltos . . . . .	61
4.4. Estimación . . . . .	62
4.4.1. Identificación de Saltos . . . . .	63
4.4.2. Estimación de Coeficientes de la Difusión . . . . .	64
4.4.3. Ventajas del Modelo de Kou . . . . .	66
4.4.4. Estimación de los Saltos . . . . .	68
4.4.5. Estimación del Generador Infinitesimal . . . . .	69
4.5. Estudio de Simulación . . . . .	71
<b>5. Ajuste del Modelo a Datos Reales</b>	<b>79</b>
5.1. Dos Acciones del bloque FAANG . . . . .	79
5.2. El Problema de la Identificación de los Saltos . . . . .	81
5.3. Ajuste del Modelo a las series de Netflix y Amazon . . . . .	89
5.4. Comparación del Generador Infinitesimal . . . . .	93
<b>6. Conclusiones</b>	<b>97</b>
<b>A. Códigos</b>	<b>99</b>
A.1. Simulación de un PSM . . . . .	99
A.2. Simulación de un PSM con Estado Absorbente . . . . .	100
A.3. Simulación de una Distribución Tipo Fase . . . . .	101
A.4. Estimación del tiempo medio de absorción vía Monte Carlo . . . . .	102
A.5. Momentos de una distribución Tipo Fase . . . . .	102
A.6. Estimación del Generador Infinitesimal de un PSM con Información Completa . . . . .	103
A.7. Método Auxiliar de Puentes Markovianos . . . . .	105
A.8. Método de la Bisección . . . . .	106
A.9. Método de Aceptación-Rechazo . . . . .	115
A.10. Método de Uniformización . . . . .	116
A.11. Norma Matricial Modificada . . . . .	118
A.12. Algoritmo EM Estocástico para estimar el Generador Infinitesimal de un PSM con información incompleta . . . . .	119
A.13. Algoritmo EM Monte Carlo para estimar el Generador Infinitesimal de un PSM con información incompleta . . . . .	121
A.14. Simulación de un MB . . . . .	123
A.15. Simulación de un MB con Deriva vía su solución analítica . . . . .	124
A.16. Simulación de un MBG vía su solución analítica . . . . .	124
A.17. Estimación de Parámetros de un MBG . . . . .	125
A.18. Simulación del Modelo de Difusión Markov-Modulado con Saltos . . . . .	126
A.19. Identificación de los Saltos . . . . .	128
A.20. Estimación de Parámetros de la Difusión del Modelo de Difusión Markov- Modulado con Saltos . . . . .	129
A.21. Clasificación de Bloques . . . . .	132

# Capítulo 1

## Introducción

En la actualidad, el rápido crecimiento de los mercados financieros así como la gran incertidumbre que éstos implican, ha llevado tanto al desarrollo como a la utilización de modelos y teoría matemática cada vez más complejos. Esto con el fin de que el comportamiento de dichos modelos se asemeje cada vez más a los fenómenos financieros que se observan en la realidad.

Las finanzas cuantitativas, también conocidas como finanzas matemáticas son una rama de las matemáticas aplicadas enfocadas a la modelación matemática de los mercados financieros. Esta rama matemática es de gran relevancia en el ámbito financiero pues brinda herramientas y modelos matemáticos con el fin de explicar fenómenos como el cambio en las tasas de interés, el movimiento en los precios de las acciones, la quiebra de entidades financieras, etc. Entre las herramientas más comunes dentro del área, se encuentran: Probabilidad, Estadística, Procesos Estocásticos, Teoría de la Medida y dentro de estas sobresale el uso del Cálculo Estocástico, pues tiene un papel fundamental en la valuación de instrumentos financieros derivados, tales como: opciones, swaptions, caps, floors, etc.

El matemático francés Louis Bachelier es considerado uno de los padres de las finanzas cuantitativas, en su tesis doctoral [17] presentó por primera vez un modelo matemático del Movimiento Browniano usado para valorar opciones financieras sobre acciones. Sin embargo, una deficiencia del modelo de Bachelier es que los precios de las acciones podrían ser negativos, debido a que los incrementos de éste tienen una distribución Normal. La deficiencia en el modelo de Bachelier fue corregida por Samuelson al introducir el Movimiento Browniano Geométrico (MBG), que es el proceso estocástico  $(S_t)_{t \geq 0}$  solución de la ecuación diferencial estocástica

$$dS_t = \mu S_t dt + \sigma S_t dW_t, \quad S_0 > 0, \quad \mu \in \mathbb{R}, \quad \sigma > 0.$$

Dicho proceso estocástico fue utilizado en 1973 por Fischer Black y Myron Scholes para asemejar el movimiento de las acciones en un modelo nuevo de valuación de

opciones [18]. Después, el entendimiento matemático del modelo fue expandido por el economista Robert Merton y dicho modelo les valió a Merton y Scholes el Premio Nobel de Economía en 1997.

A pesar del éxito del modelo Black–Scholes, éste se vio envuelto en críticas y comentarios negativos debido a que no todos estaban de acuerdo con la dinámica que se suponía para el movimiento de las acciones, debido a que modelar el precio de las acciones con un MBG tiene ventajas y notables desventajas.

Entre las ventajas se encuentran:

1. Bajo este modelo, los retornos esperados del MBG son independientes del valor del proceso, lo cual es algo que se esperaría que pase en la realidad.
2. El MBG solamente toma valores positivos, como los precios de las acciones.
3. Gráficamente, las trayectorias del proceso muestran la misma “rugosidad” que las trayectorias de los precios de las acciones.
4. Los cálculos con este proceso son relativamente sencillos.

Sin embargo, el modelo no es perfecto, ya que cuenta con las siguientes desventajas:

1. La volatilidad en el modelo no cambia, sin embargo se ha observado que en la realidad, la volatilidad de las acciones no es constante.
2. Debido a que se trata de un proceso continuo, los cambios en el valor del proceso no son tan significativos, mientras que en la realidad algunas acciones presentan saltos abruptos en sus precios, debido a eventos fortuitos o noticias que puedan afectar la confianza en la empresa por parte de los poseedores de acciones.
3. La distribución de las colas de los log-retornos no se ajusta adecuadamente a la de una distribución Normal al observar datos reales.

Una posible solución al segundo problema del MBG sería incrementar la volatilidad del proceso para poder simular los cambios abruptos, sin embargo el proceso se vuelve caótico, y por lo tanto, no se apega a la realidad.

El problema de los saltos en los precios de las acciones ha sido investigado en varias ocasiones, incluso Robert Merton hizo una modificación al Movimiento Browniano Geométrico para obtener una ecuación diferencial estocástica de la forma

$$dS_t = \mu S_t dt + \sigma S_t dW_t + d \sum_{k=1}^{N_t} \Delta P_k$$

donde el término adicional añadido por Merton es un Proceso Poisson de tasa  $\lambda$ . La adición de este proceso buscaba representar los cambios abruptos eventuales en las acciones, sin embargo la volatilidad permanecía constante al igual que en el Movimiento Browniano Geométrico. Además la cobertura de portafolios usando este modelo es muy complicada, pues calibrar adecuadamente la parte de los saltos es difícil.

La meta de esta tesis es profundizar en la dinámica de los movimientos de las acciones para solucionar las desventajas mencionadas anteriormente y así lograr que la modelación de las acciones sea más apegada a la realidad. La idea es modificar el modelo de Merton para obtener una ecuación diferencial estocástica de la forma:

$$dS_t = \mu_{X_t} S_t dt + \sigma_{X_t} S_t dW_t + d \sum_{k=1}^{N_t} \Delta P_k$$

donde,  $W_t$  es un Movimiento Browniano,  $S_t$  es el precio de la acción,  $\Delta P_k$  representan los saltos o cambios abruptos del precio de la acción y el subíndice  $X_t$  de los coeficientes  $\mu$  y  $\sigma$  es el estado al tiempo  $t$  de un Proceso de Saltos de Markov (PSM), con espacio de estados  $\{1, \dots, m\}$ . Es decir, tenemos una colección de parámetros  $\{\mu_1, \dots, \mu_m\}$ ,  $\{\sigma_1, \dots, \sigma_m\}$ , porque la dinámica de una acción no siempre se mantiene constante durante periodos largos de tiempo, sino que cambia dependiendo el estado actual del mercado, la confianza de los inversionistas, etc. Además,  $N_t$  es el número de saltos totales del PSM hasta el tiempo  $t$ , éste representará la ocurrencia los eventos abruptos que provocan las discontinuidades en las trayectorias de los precios de los activos.

La idea principal es que el Proceso de Saltos de Markov represente el ambiente financiero al cual está sometida la acción y que por ende tendrá efecto en su dinámica de movimiento, tanto en el coeficiente de deriva  $\mu$  como en el coeficiente de volatilidad  $\sigma$ ; además cada que la acción tenga un cambio abrupto en su valor, la dinámica de movimiento cambia de parámetros. Modelar las acciones de esta forma soluciona el problema de la volatilidad constante que se supone por el uso del Movimiento Browniano Geométrico y también da la posibilidad de tener cambios abruptos en el precio de la acción sin necesidad de incrementar la volatilidad de la acción. Más aún, al introducir el Proceso de Saltos de Markov como un modulador del ambiente financiero al que está sometida la acción, al final de la tesis veremos que es posible comparar el Generador Infinitesimal de dos acciones distintas para identificar similitudes en los ambientes que les favorecen, más aún se incluye el código para hacer la estimación de los parámetros del modelo en el lenguaje de programación R.

En el Capítulo 2 se presentan conceptos básicos necesarios para el desarrollo de la tesis, daremos una breve introducción a los Procesos de Saltos de Markov (Cadenas de

Markov a Tiempo Continuo). Se enunciarán las propiedades de dichos procesos, sus diferencias y similitudes con procesos conocidos, así como también se presentarán algunos ejemplos de este tipo de procesos. Adicionalmente, en este Capítulo se revisarán los resultados más importantes del Cálculo Estocástico con respecto al Movimiento Browniano, así como detalles técnicos importantes.

En el Capítulo 3 se usará la teoría revisada en el Capítulo 2 para poder desarrollar métodos de simulación y de estimación tanto para Procesos de Saltos de Markov como para procesos solución a ecuaciones diferenciales estocásticas. Dichos métodos tomarán en cuenta la aplicación a Finanzas Cuantitativas al considerar problemas de datos incompletos (por la discretización en los reportes de los precios de las acciones) y también se incluyen ejemplos clásicos para que el lector se familiarice con éstos.

En el Capítulo 4 se darán interpretaciones financieras introductorias para la teoría vista en el Capítulo 2 y se presentará el Modelo de Merton como punto de partida para finalmente introducir el Modelo de Difusión Markov-Modulado con Saltos como una alternativa al MBG. Este nuevo modelo nos permite asemejar tanto los cambios en la dinámica de los precios de las acciones, así como también los cambios abruptos que ocasionalmente ocurren. Además, se utilizarán algunos métodos de estimación vistos en el Capítulo 3 y se modificarán algunos otros con el fin de corregir ciertas desventajas que tienen debido a la complejidad del Modelo de Difusión Markov-Modulado con Saltos.

En el Capítulo 5 se hará el ajuste del Modelo de Difusión Markov-Modulado con Saltos a dos series financieras del bloque FAANG con el objetivo de mostrar la consistencia del modelo con lo que se ve en el mundo real. Se verá además que debido a que se tratan de datos reales, se tendrán que hacer algunas consideraciones adicionales con respecto a la identificación de los saltos. Finalmente, se hará una comparación del generador infinitesimal obtenido para resaltar detalles y similitudes importantes.

Finalmente en el Capítulo 6 se presentan las conclusiones con respecto al Modelo de Difusión Markov-Modulado y la estimación de parámetros obtenida para los datos reales.

En el Apéndice A se incluyen todos los códigos de los algoritmos utilizados en la tesis en el lenguaje de programación R.

# Capítulo 2

## Preliminares

El objetivo de este capítulo es introducir la teoría matemática básica que se utilizará en esta tesis. Debido al carácter matemático del Modelo de Difusión Markov-Modulado con Saltos, es necesario conocer las bases matemáticas bajo las cuales se sustenta dicho modelo, para este fin la sección cuenta con las definiciones y resultados relevantes de los Procesos de Saltos de Markov, así como también una breve introducción al Cálculo Estocástico con respecto al Movimiento Browniano. El contenido proporcionado en este capítulo será de gran relevancia en secciones posteriores pues es necesario tanto para el desarrollo del modelo propuesto en esta tesis como para poder presentar los distintos métodos de simulación y estimación que se mencionarán en el Capítulo 3.

### 2.1. Definiciones Básicas

Debido a que los Procesos de Markov serán de gran importancia en esta tesis, es necesario presentar la definición de estos procesos en su forma más general.

**Definición 2.1.1** (Espacio Medible). *Un Espacio Medible es una pareja  $(X, \mathcal{F})$  donde  $X$  es un conjunto arbitrario distinto del vacío y  $\mathcal{F}$  es una  $\sigma$ -álgebra de subconjuntos de  $X$ .*

**Definición 2.1.2** (Espacio de Probabilidad). *Un Espacio de Probabilidad es una 3-tupla  $(\Omega, \mathcal{F}, \mathbb{P})$  donde  $(\Omega, \mathcal{F})$  es un espacio medible y  $\mathbb{P}$  es una medida de probabilidad definida sobre  $\mathcal{F}$ .*

**Definición 2.1.3** (Filtración). *Sea  $(\Omega, \mathcal{F}, \mathbb{P})$  un espacio de probabilidad. Supongamos que para cada  $t > 0$  se cumple que  $\mathcal{F}_t$  es una sub  $\sigma$ -álgebra de  $\mathcal{F}$ . Entonces, decimos que  $(\mathcal{F}_t)_{t>0}$  es una filtración si*

$$\mathcal{F}_s \subset \mathcal{F}_t \subset \mathcal{F}, \quad \text{para } 0 < s < t.$$

**Definición 2.1.4** (Medibilidad). Sean  $(X, \mathcal{F})$  y  $(Y, \mathcal{G})$  dos espacios medibles. Decimos que una función  $f : X \rightarrow Y$  es  $(\mathcal{F}, \mathcal{G})$ -medible si  $\forall A \in \mathcal{G}$  se cumple

$$f^{-1}(A) = \{x \in X : f(x) \in A\} \in \mathcal{F}.$$

Con respecto a la Definición 2.1.4, con frecuencia es necesario especificar quiénes son las  $\sigma$ -álgebras  $\mathcal{F}$  y  $\mathcal{G}$  debido a que podemos estar trabajando con varios espacios medibles que pueden o no compartir  $\sigma$ -álgebras. Sin embargo, cuando sea evidente cuáles son éstas, simplemente diremos que la función es medible.

**Definición 2.1.5** (Adaptabilidad). Sea  $(\Omega, \mathcal{F}, \mathbb{P}, \mathcal{F}_s)$  un espacio filtrado. Decimos que un proceso estocástico  $(X_t)_{t \geq 0}$  es adaptado a la filtración  $(\mathcal{F}_s)_{s \geq 0}$  si

$$X_t : \Omega \rightarrow \mathbb{R} \text{ es } (\mathcal{F}_t, B(\mathbb{R})) \text{ - medible } \quad \forall t \geq 0$$

donde  $B(\mathbb{R})$  denota a la  $\sigma$ -álgebra de Borel.

**Definición 2.1.6** (Proceso de Markov). Sea  $(\Omega, \mathcal{F}, \mathbb{P}, \mathcal{F}_s)$  un espacio filtrado. Se dice que un proceso estocástico  $(X_s)_{s \in I}$ , adaptado a la filtración  $(\mathcal{F}_s)_{s \in I}$  (con  $I \subset \mathbb{R}$ ), es un proceso de Markov si  $\forall A \in \mathcal{F}$ ,  $(X_s)_{s \in I}$  cumple la propiedad de Markov:

$$\mathbb{P}(X_t \in A \mid \mathcal{F}_s) = \mathbb{P}(X_t \in A \mid X_s) \quad \forall s < t, \quad s, t \in I.$$

La Definición 2.1.6 es la forma más general para definir un proceso de Markov, sin embargo en esta tesis no se usará la definición general, sino que usaremos procesos más específicos en los cuales la propiedad de Markov puede escribirse de manera más simple. Es necesario resaltar que cuando se tiene un espacio medible donde  $\Omega$  es un conjunto discreto, frecuentemente se utiliza  $\mathcal{F} = 2^\Omega$ . Los Procesos de Markov con este tipo de espacios medibles serán el objeto de estudio de la siguiente sección.

## 2.2. Cadenas de Markov

A partir de este momento cuando se diga Cadena de Markov nos estaremos refiriendo al proceso descrito por la Definición 2.2.1 y si no se hace algún comentario adicional acerca de su espacio de estados, se dará por hecho que éste es finito ( $E = \{1, 2, \dots, m\}$ ).

**Definición 2.2.1** (Cadena de Markov). Una Cadena de Markov  $(Y_n)_{n \in \mathbb{N}}$  con espacio de estados  $E$  (finito o numerable) es un proceso estocástico a tiempo discreto que cumple la propiedad de Markov:

$$\mathbb{P}(Y_{n+1} = y_{n+1} \mid Y_n = y_n, Y_{n-1} = y_{n-1}, \dots, Y_0 = y_0) = \mathbb{P}(Y_{n+1} = y_{n+1} \mid Y_n = y_n)$$

donde  $y_k \in E \quad \forall k \in \{0, 1, \dots, n, n+1\}$ .

Se dice que una Cadena de Markov es homogénea si la probabilidad de ir de un estado a otro en un paso, no se ve afectada por el tiempo en el cual se encuentre la cadena, es decir:

$$\mathbb{P}(Y_n = j | Y_{n-1} = i) = \mathbb{P}(Y_1 = j | Y_0 = i), \quad \forall i, j \in E, \quad \forall n \in \mathbb{N}.$$

Así, podemos usar la siguiente notación para denotar dichas probabilidades:

$$p_{ij} = \mathbb{P}(Y_1 = j | Y_0 = i),$$

$$p_{ij}^{(n)} = \mathbb{P}(Y_n = j | Y_0 = i).$$

**Definición 2.2.2** (Matriz de Transición). *A la matriz formada por las probabilidades  $p_{ij}$  se le conoce como Matriz de Transición y se denota como  $P = (p_{ij})_{i,j \in E}$ .*

**Definición 2.2.3** (Distribución Inicial). *Se dice que un vector  $\pi = (\pi_1, \dots, \pi_m)$  es una Distribución Inicial de la Cadena de Markov  $(Y_n)_{n \in \mathbb{N}}$  si se cumplen:*

$$\pi_i = \mathbb{P}(Y_0 = i), \quad \sum_i^m \pi_i = 1 \quad y \quad \pi_i \geq 0.$$

La matriz de transición junto con el vector de distribución inicial determinan de manera única a la Cadena de Markov.

**Definición 2.2.4** (Distribución Estacionaria). *Si  $\pi$  es una distribución inicial de la Cadena de Markov, con matriz de transición  $P = (p_{ij})_{i,j \in E}$ , que satisface*

$$\pi_j = \sum_{i \in E} \pi_i p_{ij}, \quad \forall j \in E \tag{2.1}$$

*entonces  $\pi$  es una distribución estacionaria.*

Es importante destacar, que las Cadenas de Markov bajo ciertas condiciones, cumplen una propiedad de convergencia en el sentido de que, después de un tiempo suficientemente largo, se estabilizan. Esta propiedad será de gran importancia en secciones posteriores, razón por la cual es necesario definir conceptos para poder saber cuáles son las condiciones necesarias y qué implicaciones tienen éstas.

**Definición 2.2.5** (Cadena Irreducible). *Sea una Cadena de Markov con espacio de estados  $E$  y matriz de transición  $P = (p_{ij})_{i,j \in E}$ . Decimos que la Cadena es Irreducible si para cualquier pareja de estados  $i$  y  $j$  en  $E$ , existe  $n \in \mathbb{N}$  tal que  $p_{ij}^{(n)} > 0$ .*

**Definición 2.2.6** (Periodicidad). *Para una Cadena de Markov con espacio de estados  $E$ , se define el periodo de un estado  $i \in E$  como*

$$d(i) = \text{mcd}\{n \in \mathbb{N} \mid p_{ij}^{(n)} > 0\}$$

*donde mcd se refiere al máximo común divisor. Si  $d(i) = 1$ , se dice que el estado  $i$  es aperiódico.*

**Definición 2.2.7** (Cadena aperiódica). *Decimos que una Cadena de Markov es aperiódica si todos sus estados son aperiódicos.*

**Definición 2.2.8** (Recurrencia). *Sea  $i \in E$  un estado de una Cadena de Markov  $(X_n)_{n \in \mathbb{N}}$ . Entonces  $i$  es recurrente si*

$$\mathbb{P}(Y_n = i \text{ p.a. } n \geq 1 \mid Y_0 = i) = 1.$$

*En caso contrario, se dice que  $i$  es transitorio.*

**Definición 2.2.9** (Recurrencia Positiva). *Si  $(Y_n)_{n \in \mathbb{N}}$  es una Cadena de Markov con espacio de estados  $E$ . Decimos que un estado recurrente  $i \in E$  es recurrente positivo (o positivo-recurrente) si*

$$\mathbb{E}(\min\{n \geq 1 : Y_n = i\} \mid Y_0 = i) < \infty.$$

*Más aún, si todos los estados de la cadena son positivo-recurrentes, entonces diremos que la cadena es recurrente positiva (o positiva-recurrente).*

**Teorema 2.2.10** (Convergencia de Cadenas de Markov). *Supongamos que una Cadena de Markov es irreducible, aperiódica y positivo-recurrente, entonces las probabilidades límite*

$$\pi_j = \lim_{n \rightarrow \infty} p_{ij}^{(n)} \quad (2.2)$$

*existen y además están constituidas por la única solución del sistema de la ecuación (2.1).*

El Teorema 2.2.10 (cuya prueba puede encontrarse en [8]) nos dice que a largo plazo, la probabilidad de estar en alguno de sus estados converge a la distribución estacionaria. Más aún, dicha distribución es única, por lo que la proporción de tiempo que pasa la Cadena en cada estado es proporcional a la distribución estacionaria.

## 2.3. Procesos de Saltos de Markov

Si en vez de considerar un proceso estocástico a tiempo discreto  $(Y_n)_{n \in \mathbb{N}}$ , se considera uno a tiempo continuo  $(X_t)_{t \geq 0}$  que cumpla la propiedad de Markov descrita en la Definición 2.3.1 tendremos un Proceso de Saltos de Markov.

**Definición 2.3.1** (Proceso de Saltos de Markov). *Un Proceso de Saltos de Markov (o Cadena de Markov a Tiempo Continuo)  $(X_t)_{t \geq 0}$  con espacio de estados  $E$  (finito o numerable) es un proceso estocástico a tiempo continuo que cumple la propiedad de Markov:*

$$\mathbb{P}(X_{t_{n+1}} = x_{n+1} \mid X_{t_n} = x_n, X_{t_{n-1}} = x_{n-1}, \dots, X_{t_0} = x_0) = \mathbb{P}(X_{t_{n+1}} = x_{n+1} \mid X_{t_n} = x_n)$$

*donde  $0 \leq t_0 < t_1 < \dots < t_{n-1} < t_n$  y  $x_k \in E \quad \forall k \in \{0, 1, \dots, n, n+1\}$ .*

En este texto se refiere a los Procesos de Saltos de Markov cuando se escriba PSM y, al igual que en el caso discreto, si no se hacen comentarios adicionales acerca del espacio de estados  $E$ , se estará suponiendo que éste es finito.

La homogeneidad para los PSM se puede definir de forma análoga a como se hizo en el caso discreto. Es decir, el PSM es homogéneo si la probabilidad de ir de un estado a otro, no se ve afectada por el tiempo en el cual se encuentre el proceso, o matemáticamente

$$\mathbb{P}(X_{t+h} = j | X_h = i) = \mathbb{P}(X_t = j | X_0 = i), \quad \forall i, j \in E, \quad \forall t, h > 0.$$

De manera que, cuando el PSM es homogéneo, podemos adoptar una notación para las probabilidades de transición, para cada  $t \geq 0$  como:

$$p_{ij}(t) = \mathbb{P}(X_t = j | X_0 = i).$$

Y de manera equivalente  $P(t) = (p_{ij}(t))_{i,j \in E}$ .

**Definición 2.3.2** (Estado Absorbente). *Se dice que un estado  $i \in E$  es absorbente si y sólo si:*

$$p_{ij}(t) = 0, \quad \forall j \in E - \{i\}, \quad \forall t \geq 0.$$

*Y en consecuencia:*

$$p_{ii}(t) = 1, \quad \forall t \geq 0.$$

Cuando se hable de los PSM de vez en cuando será necesario especificar si se deja abierta o no la posibilidad de que el PSM en cuestión contenga estados absorbentes en su espacio de estados. Es por esto que, cuando se diga que el PSM es estándar, nos estamos refiriendo a que el proceso no contiene estados absorbentes en su espacio de estados.

**Definición 2.3.3** (Semigrupo de Transición). *A la familia de matrices  $\{P(t)\}_{t \geq 0}$  se le llama semigrupo de transición.*

El comportamiento de este tipo de procesos estocásticos es similar al del caso discreto. Sin embargo, recordemos que para una Cadena de Markov  $(Y_n)_{n \in \mathbb{N}}$ , en cada tiempo, la Cadena salta de un estado a otro y en particular es posible que, en el tiempo siguiente, la Cadena de Markov salte al estado en el que se encuentra actualmente (si  $\mathbb{P}(Y_1 = i | Y_0 = i) > 0$ ).

A diferencia de las Cadenas de Markov, los Procesos de Saltos de Markov permanecen en un estado  $i \in E$  un tiempo aleatorio  $\tau_i$  y después saltan a un estado  $j \in E - \{i\}$ , donde se repite el proceso anterior siempre y cuando el estado  $i$  no sea un estado absorbente.

Denotaremos por  $p_{ij}$  (tasa de transición) a la probabilidad de que el Proceso  $(X_t)_{t \geq 0}$

pase del estado  $i$  al estado  $j$  al efectuar un salto.

Supongamos que el proceso se encuentra en el estado  $i$  al tiempo  $t$ , es decir  $X_t = i$ , entonces podemos preguntarnos por el comportamiento del proceso en el intervalo  $(t, t + h)$ . En ese caso, hay dos posibilidades:

1. El proceso permanece en  $i$  de  $t$  a  $t + h$ , esto ocurre con probabilidad  $p_{ii}(h) + o(h)$ , observemos que dicha probabilidad es mayor que  $p_{ii}(h)$  debido a que  $p_{ii}(h)$  deja abierta la posibilidad de visitar un estado distinto y regresar a  $i$  antes de  $t + h$ .
2. El proceso permanece cierto tiempo en  $i$  y después puede saltar a un estado distinto  $j$  antes de  $t + h$  donde permanece el resto del tiempo, esto ocurre con probabilidad  $p_{ij}(h) + o(h)$ . Nuevamente tenemos que  $p_{ij}(h) + o(h)$  es mayor a  $p_{ij}(h)$ , debido a que ésta última contempla saltos adicionales a otros estados antes de llegar a  $j$ .

Así, es interesante preguntarnos por el comportamiento de  $p_{ij}(h)$  para  $h$  suficientemente pequeña. Utilizaremos la siguiente representación de las transiciones infinitesimales cuya prueba se puede encontrar en [1]. A la matriz resultante  $Q = (q_{ij})_{i,j \in E}$  de la Proposición 2.3.4 se le conoce como el Generador Infinitesimal, Matriz de Intensidad (o simplemente Generador) del Proceso.

**Proposición 2.3.4** (Generador Infinitesimal). *Sea un Proceso de Saltos de Markov con semigrupo de transición  $(P(t))_{t \geq 0}$ . Entonces, el límite:*

$$\lim_{h \rightarrow 0} \frac{P(h) - I}{h} = Q$$

*existe. Es decir, los siguientes límites, existen:*

$$\lim_{h \rightarrow 0} \frac{p_{ii}(h) - 1}{h} = q_{ii} := -q_i,$$

$$\lim_{h \rightarrow 0} \frac{p_{ij}(h)}{h} = q_{ij}.$$

Así como la Matriz de Transición en el caso discreto, el Generador Infinitesimal cumple ciertas propiedades, éstas se enlistan en la Proposición 2.3.5 y su demostración puede encontrarse en [1].

**Proposición 2.3.5** (Propiedades del Generador Infinitesimal). *El Generador Infinitesimal  $Q = (q_{ij})_{i,j \in E}$  cumple las siguientes propiedades:*

$$q_i = \sum_{j \neq i} q_{ij} \tag{2.3}$$

$$p_{ij} = \frac{q_{ij}}{q_i}, \quad (2.4)$$

$$P(t) = \exp(Qt) = \sum_{n=0}^{\infty} \frac{(Qt)^n}{n!}. \quad (2.5)$$

Más aún, suponiendo que el proceso se encuentra en  $i \in E$  al tiempo  $t$  ( $X_t = i$ , donde  $i$  no es un estado absorbente), recordemos que el proceso permanece en dicho estado un tiempo aleatorio  $\tau_i$  antes de saltar a otro, entonces:

$$\tau_i \sim \text{Exponencial}(q_i). \quad (2.6)$$

Así, por (2.3) y la Definición 2.3.2 podemos saber que  $i$  es un estado absorbente si  $q_{ij} = 0 \quad \forall j \in E$ . Es decir, si el  $i$ -ésimo renglón del Generador  $Q$  es un renglón de ceros, entonces  $i \in E$  es un estado absorbente. Lo anterior es debido a que, si  $i \in E$  es un estado absorbente, entonces:

$$\lim_{h \rightarrow 0} \frac{p_{ij}(h)}{h} = \lim_{h \rightarrow 0} \frac{0}{h} = 0 = q_{ij}, \quad \forall j \in E - \{i\}$$

y

$$\lim_{h \rightarrow 0} \frac{p_{ii}(h) - 1}{h} = \lim_{h \rightarrow 0} \frac{1 - 1}{h} = 0 = -q_i.$$

En este caso particular donde  $i \in E$  es un estado absorbente,  $\tau_i$  no está bien definido ( $\tau_i = \infty$ ) y (2.6) no aplica.

Es importante entender (2.4) y (2.6) pues serán de gran utilidad en el Capítulo 3.

A continuación se presentan dos ejemplos de Procesos de Saltos de Markov muy utilizados con el motivo de presentar aplicaciones de esta teoría, así como para comentar sus propiedades y usos.

**Ejemplo 2.3.1** (Proceso Poisson). *El Proceso de Saltos de Markov más famoso es el Proceso Poisson( $\lambda$ ), en este caso su espacio de estados es  $E = \mathbb{N} \cup \{0\}$ . Si el Proceso Poisson se encuentra en el estado  $i \in E$  en un tiempo determinado, entonces el siguiente estado que visitará es el  $i + 1 \in E$  casi seguramente y  $\tau_i \sim \text{Exponencial}(\lambda) \quad \forall i \in E$ . Por lo tanto, de las propiedades (2.3), (2.4) y (2.6) en la Proposición 2.3.5 podemos inferir que su Generador  $Q$  es:*

$$Q = \begin{pmatrix} -\lambda & \lambda & 0 & 0 & \cdots \\ 0 & -\lambda & \lambda & 0 & \cdots \\ 0 & 0 & -\lambda & \lambda & \cdots \\ \vdots & \vdots & \vdots & \ddots & \ddots \end{pmatrix}$$

Este proceso estocástico es usado para modelar la llegada de personas a una sucursal bancaria, la llegada de llamadas a un centro de atención o de camiones a una parada. Una de las propiedades más importantes de este proceso es que, si el proceso tuvo  $k$  saltos en el intervalo  $[0, T]$ , entonces los tiempos de salto tienen la misma distribución que una muestra de  $k$  Estadísticos de Orden  $(U_{(1)}, \dots, U_{(k)})$  donde  $U_i \sim \text{Unif}(0, T)$ ,  $\forall i \in \{1, \dots, k\}$ . La demostración de dicha propiedad (que será de gran utilidad más adelante) se puede encontrar en [7].

**Ejemplo 2.3.2** (Proceso de Nacimiento y Muerte). Este proceso tiene espacio de estados  $E = \mathbb{N} \cup \{0\}$ . Su Generador  $Q$  es:

$$Q = \begin{pmatrix} -\lambda_0 & \lambda_0 & 0 & 0 & \cdots \\ \mu_1 & -(\mu_1 + \lambda_1) & \lambda_1 & 0 & \cdots \\ 0 & \mu_2 & -(\mu_2 + \lambda_2) & \lambda_2 & \cdots \\ \vdots & \vdots & \vdots & \ddots & \ddots \end{pmatrix}$$

Donde  $\lambda_0, \lambda_1, \mu_1, \lambda_2, \mu_2, \dots$  son constantes positivas conocidas como las tasas de nacimiento ( $\lambda_i$ ) y muerte ( $\mu_i$ ). Dicho proceso estocástico se utiliza para modelar el comportamiento del número de individuos de una especie. Los saltos de este proceso se interpretan como el nacimiento de un individuo, si el proceso pasa del estado  $i$  al  $i+1$  para alguna  $i \in E$ , o como muerte, si el proceso pasa del estado  $i+1$  al  $i$  para alguna  $i \in E$ .

### 2.3.1. Distribuciones Tipo Fase

En matemáticas, nos llama la atención cuando algo puede ser visto o representado en distintas maneras. Las Distribuciones Tipo Fase son una forma de representar el tiempo (aleatorio) que tarda un Proceso de Saltos de Markov en llegar a un estado absorbente. Además, es importante estudiar las distribuciones tipo fase, debido a que éstas son densas en las variables aleatorias continuas con soporte en  $(0, \infty)$  y por su representación como tiempos de llegada dentro un Proceso de Saltos de Markov.

Sea  $(X_t)_{t \geq 0}$  un Proceso de Saltos de Markov con espacio de estados finito  $S = \{1, 2, \dots, m+1\}$  donde los estados  $1, 2, \dots, m$  son transitorios y el estado  $m+1$  es absorbente. Entonces el Generador  $Q$  del proceso se puede escribir de la forma

$$Q = \begin{pmatrix} \mathbf{\Lambda} & \lambda \\ \mathbf{0} & 0 \end{pmatrix}, \quad (2.7)$$

donde  $\mathbf{\Lambda}$  es una matriz de  $m \times m$ ,  $\lambda$  es un vector columna de  $m$  entradas y  $\mathbf{0}$  es un vector renglón de  $m$  entradas iguales a cero. A  $\mathbf{\Lambda}$  se le llama **Matriz de Subintensidades** y a los elementos de  $\lambda$  se les llama **Tasas de Salida**. Además, de (2.3) podemos deducir que  $\lambda = -\mathbf{\Lambda}\mathbf{1}$  donde  $\mathbf{1}$  es un vector columna de unos.

Para el estudio de las Distribuciones Tipo Fase, consideraremos distribuciones iniciales  $\pi$  de la forma  $\pi = (\pi_1, \dots, \pi_m, \pi_{m+1})$  donde  $\pi_{m+1} = \mathbb{P}(X_0 = m + 1) = 0$ . Más aún, debido a esto último, podemos reducir  $\pi$  a la forma:  $\pi = (\pi_1, \pi_2, \dots, \pi_m)$  donde solamente nos enfocaremos en la probabilidad de iniciar sobre los estados transitorios.

Así, podemos definir el tiempo de absorción  $\tau$  de la siguiente forma:

$$\tau = \inf\{t > 0 \mid X_t = m + 1\}.$$

Se dice que  $\tau$  tiene una distribución Tipo Fase, con parámetros  $\mathbf{\Lambda}$  y  $\pi$ , o de forma abreviada

$$\tau \sim PH(\pi, \mathbf{\Lambda}).$$

Algunas de las propiedades de las Distribuciones Tipo Fase se exponen en la siguiente proposición cuya demostración puede encontrarse en [2].

**Proposición 2.3.6** (Propiedades de las Distribuciones Tipo Fase). *Si  $\tau \sim PH(\pi, \mathbf{\Lambda})$ , entonces  $\tau$  satisface las siguientes propiedades:*

1. *La densidad de  $\tau$  está dada por*

$$f_\tau(t) = \pi \exp(\mathbf{\Lambda}t)\lambda.$$

2. *La función de distribución de  $\tau$  está dada por*

$$F_\tau(t) = 1 - \pi \exp(\mathbf{\Lambda}t)\mathbf{1}.$$

3. *El  $n$ -ésimo momento de  $\tau$  se puede obtener de la siguiente forma*

$$\mathbb{E}(\tau^n) = (-1)^n n! \pi \mathbf{\Lambda}^{-n} \mathbf{1}. \quad (2.8)$$

4. *La función generadora de momentos de  $\tau$  tiene la siguiente forma*

$$\mathbb{E}(\exp(\tau t)) = \pi(-tI - \mathbf{\Lambda})^{-1}\lambda.$$

## 2.4. Movimiento Browniano

En las finanzas, uno de los procesos estocásticos más importantes es el Movimiento Browniano, por lo cual se vuelve muy relevante para esta tesis. Incluimos esta sección con las propiedades y definiciones necesarias para entender el modelo matemático que se propondrá, así como para desarrollar un algoritmo de simulación de este proceso que se presentará en el Capítulo 3 y de aquellos procesos relacionados con el Movimiento Browniano.

**Definición 2.4.1** (Movimiento Browniano). Sea  $(W_t)_{t \geq 0}$  un proceso estocástico a tiempo continuo. Decimos que  $(W_t)_{t \geq 0}$  es un Movimiento Browniano (o un Proceso de Wiener) si satisface las siguientes propiedades.

1.  $W_0 = 0$  casi seguramente.
2. Las trayectorias  $t \rightarrow W_t$  son continuas casi seguramente.
3. Para cualquier conjunto de tiempos  $0 \leq t_1 < t_2 < \dots < t_n$ , las variables aleatorias  $W_{t_1}, W_{t_2} - W_{t_1}, \dots, W_{t_n} - W_{t_{n-1}}$  son independientes.
4. Para cualesquiera  $0 \leq s < t$ ,  $W_t - W_s \sim \text{Normal}(0, t - s)$ .

Las propiedades 3 y 4 de la Definición 2.4.1 son muy importantes debido a que serán útiles para la simulación.

En ocasiones, para abreviar nos vamos a referir al Movimiento Browniano simplemente como MB. A continuación se enunciarán dos resultados que serán de gran importancia en esta tesis, la prueba de estas afirmaciones se puede encontrar en [8].

**Proposición 2.4.2.** El Movimiento Browniano  $(W_t)_{t \geq 0}$  es un proceso de Markov, es decir, satisface la propiedad enunciada en la Definición 2.1.6.

**Proposición 2.4.3** (No Diferenciabilidad y Variación). Sea  $t \in \mathbb{R}^+ \cup \{0\}$ . El Movimiento Browniano no es diferenciable en  $t$  casi seguramente. Más aún, la variación del MB, en un intervalo  $[a, b]$  de cualquier longitud, es infinita casi seguramente.

La Proposición 2.4.3 es de suma importancia, pues nos dice que no podemos definir una integral con respecto al MB

$$\int_0^t X_s dW_s, \quad (2.9)$$

como se hace con una integral de Riemann-Stieltjes pues recordemos que una integral de la forma

$$\int_a^b f(s) dg(s),$$

puede definirse en el sentido de Riemann-Stieltjes cuando la función integradora  $g : [a, b] \rightarrow \mathbb{R}$  tiene variación finita en el intervalo  $[a, b]$  y debido a que las trayectorias del MB no tienen variación acotada en ningún intervalo sin importar su longitud, esto ocasiona que sea necesario desarrollar una teoría coherente de integración si es que queremos tratar con integrales como la que se muestra en (2.9).

## 2.5. Integración Estocástica con respecto al Movimiento Browniano

El modelo que se presentará utiliza Ecuaciones Diferenciales Estocásticas las cuales están basadas en la teoría del Cálculo Estocástico con respecto al MB. Esta teoría puede consultarse en [9]. Debido a que nos enfocaremos en las aplicaciones, la construcción de la Integral Estocástica no formará parte de este documento.

Puesto que trabajaremos con integrales estocásticas con respecto al MB cada vez que se hable de procesos o funciones adaptadas, nos estaremos refiriendo a que son adaptados a la filtración natural del MB  $\mathcal{F}_t := \sigma(W_s : 0 \leq s \leq t)$  y donde la adaptabilidad se entiende en el sentido de la Definición 2.1.5.

En general, cuando hablamos de integración estocástica el objetivo es comprender y definir adecuadamente los objetos matemáticos de la integral

$$Z_t(\omega) = \int_0^t f(\omega, s) dW_s, \quad \text{para } 0 \leq t \leq T, \quad (2.10)$$

lo anterior es posible cuando la función  $f : \Omega \times [0, T]$ , en la expresión (2.10), es medible, adaptada y además satisface

$$\mathbb{P}\left(\int_0^T f^2(\omega, t) dt < \infty\right) = 1, \quad (2.11)$$

más aún, para cualquier función  $g : \mathbb{R} \rightarrow \mathbb{R}$  continua se tiene que  $f(\omega, t) = g(W_t(\omega))$  satisface (2.11). La prueba de dichas afirmaciones puede encontrarse en [9].

El Teorema Fundamental del Cálculo (TFC), que puede consultarse en [16], nos señala un camino para poder calcular integrales de las funciones de variable real que cumplan con las condiciones de integrabilidad. Por otro lado, el Cálculo Estocástico sería poco viable y significativamente más complicado si no contara con un resultado análogo al TFC que nos relacione la integral estocástica de una función con otro proceso estocástico.

Los dos siguientes resultados, cuya demostración puede encontrarse en [9], nos proporcionan una solución a nuestro problema.

**Lema 2.5.1** (Lema de Itô-caso simple). *Sea  $(W_t)_{t \geq 0}$  un MB y  $f : \mathbb{R} \rightarrow \mathbb{R}$  una función de clase  $\mathcal{C}^2$ , entonces*

$$f(W_t) - f(0) = \int_0^t f'(W_s) dW_s + \frac{1}{2} \int_0^t f''(W_s) ds, \quad \forall t \geq 0. \quad (2.12)$$

El resultado anterior es un caso específico del que se enunciará a continuación, éste último nos permite involucrar funciones  $f(t, x)$  que contengan no sólo un argumento espacial  $x = W_t$ , sino también un argumento  $t$  que representa el tiempo en el cual se evalúa el MB.

**Lema 2.5.2** (Lema de Itô-caso general). *Sea  $(W_t)_{t \geq 0}$  un MB y  $f : \mathbb{R}^+ \times \mathbb{R} \rightarrow \mathbb{R}$  una función de clase  $C^1$  en el primer argumento y de clase  $C^2$  en el segundo argumento, entonces*

$$f(t, W_t) - f(0, 0) = \int_0^t \frac{\partial f}{\partial t}(s, W_s) ds + \int_0^t \frac{\partial f}{\partial x}(s, W_s) dW_s + \frac{1}{2} \int_0^t \frac{\partial^2 f}{\partial x^2}(s, W_s) ds, \quad (2.13)$$

para toda  $t \geq 0$ .

De (2.13) podemos ver que si reordenamos el lado derecho juntando las integrales con respecto a  $s$  y haciendo  $Z_t = f(t, W_t)$  obtenemos una ecuación de la forma

$$Z_t - Z_0 = \int_0^t \alpha(s, Z_s) ds + \int_0^t \beta(s, Z_s) dW_s. \quad (2.14)$$

En algunos casos, para simplificar la notación, en lugar de usar la representación (2.14) se escribe simplemente

$$dZ_t = \alpha(t, Z_t) dt + \beta(t, Z_t) dW_t. \quad (2.15)$$

Para que podamos escribir ya sea (2.14) o (2.15) es necesario que tanto  $\alpha$  como  $\beta$  cumplan las condiciones especificadas en la Definición 2.5.3.

**Definición 2.5.3** (Difusión de Itô). *Sea  $(W_t)_{t \geq 0}$  un MB. Decimos que un proceso estocástico  $(Z_t)_{0 \leq t \leq T}$  es una Difusión de Itô si tiene la representación*

$$Z_t = Z_0 + \int_0^t \alpha(s, Z_s) ds + \int_0^t \beta(s, Z_s) dW_s \quad \text{para } 0 \leq t \leq T, \quad (2.16)$$

y  $\alpha(t, Z_t), \beta(t, Z_t)$  son procesos adaptados y medibles que satisfacen las siguientes condiciones de integrabilidad:

$$\mathbb{P}\left(\int_0^T |\alpha(s, Z_s)| ds < \infty\right) = 1 \quad \text{y} \quad \mathbb{P}\left(\int_0^T \beta(s, Z_s)^2 ds < \infty\right) = 1. \quad (2.17)$$

Una pregunta que podríamos hacernos es si en el Cálculo Estocástico existe algún resultado análogo de la regla de la cadena, o en otras palabras, supongamos que  $Z_t$  es un proceso como el de la Definición 2.5.3 y hacemos  $V_t = f(t, Z_t)$ , podemos preguntarnos si es posible escribir  $dV_t$  en términos de  $dZ_t$ . El siguiente resultado nos proporciona una manera de lograr lo anterior.

**Lema 2.5.4** (Lema de Itô). *Sea  $(W_t)_{t \geq 0}$  un MB,  $f : \mathbb{R}^+ \times \mathbb{R} \rightarrow \mathbb{R}$  una función de clase  $\mathcal{C}^{1,2}$  y  $Z_t$  una Difusión de Itô con representación*

$$dZ_t = \alpha(t, Z_t)dt + \beta(t, Z_t)dW_t. \quad (2.18)$$

Entonces,  $V_t = f(t, Z_t)$  tiene la representación

$$dV_t = \frac{\partial f}{\partial t}(t, Z_t)dt + \alpha(t, Z_t) \frac{\partial f}{\partial x}(t, Z_t)dt \quad (2.19)$$

$$+ \frac{1}{2}\beta(t, Z_t)^2 \frac{\partial^2 f}{\partial x^2}(t, Z_t)dt + \beta(t, Z_t) \frac{\partial f}{\partial x}(t, Z_t)dW_t \quad (2.20)$$

La demostración del Lema 2.5.4 puede encontrarse en [9]. De acuerdo al Lema 2.5.4, podemos notar que lo que hace especial a las Difusiones de Itô, es que continúan siendo difusiones bajo funciones  $\mathcal{C}^{1,2}$ .

## 2.6. Ecuaciones Diferenciales Estocásticas

Ya que tenemos la representación de las integrales con respecto al MB, es natural que el siguiente paso sea modelar fenómenos que hagan uso de esta herramienta. En el Cálculo Diferencial e Integral, planteamos ecuaciones diferenciales que nos ayudan a modelar ciertos fenómenos bajo la suposición de que éstos sean deterministas. La idea de introducir el concepto de Integral Estocástica en este trabajo, es debido a que se precisa modelar fenómenos (en nuestro caso financieros) que están sujetos a condiciones que los hagan parecer estocásticos. Es por eso que nos interesarán modelos de la forma

$$dZ_t = \alpha(t, Z_t)dt + \beta(t, Z_t)dW_t, \quad Z_0 = z_o. \quad (2.21)$$

A las ecuaciones como (2.21) se les conoce como Ecuaciones Diferenciales Estocásticas (se les denotará EDE por sus siglas). Éstas, al igual que las ecuaciones diferenciales ordinarias o parciales, tienen su propia teoría de existencia y unicidad. El siguiente teorema (cuya demostración puede consultarse en [9]) impone condiciones sobre los coeficientes  $\alpha$  y  $\beta$  para que la solución a la ecuación (2.21) exista.

**Teorema 2.6.1** (Teorema de Existencia y Unicidad para EDE). *Sea la ecuación diferencial estocástica*

$$dZ_t = \alpha(t, Z_t)dt + \beta(t, Z_t)dW_t, \quad Z_0 = z_o,$$

para  $0 \leq t \leq T$ . Si los coeficientes  $\alpha$  y  $\beta$  satisfacen

$$|\alpha(t, x) - \alpha(t, y)|^2 + |\beta(t, x) - \beta(t, y)|^2 \leq K|x - y|^2 \quad (2.22)$$

y

$$|\alpha(t, x)|^2 + |\beta(t, x)|^2 \leq K(1 + |x|^2), \quad (2.23)$$

para alguna  $K > 0$ , entonces existe  $Z_t$  una solución a la EDE continua y adaptada, que satisface

$$\sup_{0 \leq t \leq T} \mathbb{E}(Z_t^2) < \infty. \quad (2.24)$$

Más aún, si dos procesos  $Z_t$  y  $V_t$  son soluciones de la EDE continuas que satisfacen (2.24), entonces

$$\mathbb{P}(Z_t = V_t \quad \forall t \in [0, T]) = 1.$$

Cuando podamos encontrar una solución  $Z_t$  a una EDE que cumpla las condiciones establecidas en el Teorema 2.6.1, le llamaremos **Solución Fuerte** a la EDE.

Notemos que el Teorema 2.6.1, a pesar de que nos ayuda a identificar cuando existe una solución a la EDE, no nos dice cómo encontrar dicha solución, sin embargo en ocasiones, podemos encontrarla usando alguna de las versiones del Lema de Itô o inspeccionando y haciendo observaciones.

**Ejemplo 2.6.1** (Movimiento Browniano con Deriva). Sea  $(W_t)_{t \geq 0}$  un MB. Al proceso  $S_t = \mu t + \sigma W_t$  (con  $\mu \in \mathbb{R}, \sigma > 0$ ) se le llama *Movimiento Browniano con Deriva*. Podemos usar el Lema de Ito 2.5.2 para obtener su EDE. En este caso  $f(t, x) = \mu t + \sigma x$ . Por lo tanto

$$S_t = f(t, W_t) = \int_0^t \mu ds + \int_0^t \sigma dW_s,$$

o de forma equivalente

$$dS_t = \mu dt + \sigma dW_t. \quad (2.25)$$

Haciendo la identificación de coeficientes con los del Teorema 2.6.1 obtenemos que  $\alpha(t, x) = \mu$  y  $\beta(t, x) = \sigma$ . Así, tomando  $K = \mu^2 + \sigma^2$  tenemos que  $\alpha$  y  $\beta$  satisface la condición (2.22) pues

$$|\mu - \mu|^2 + |\sigma - \sigma|^2 = 0 \leq (\mu^2 + \sigma^2)|x - y|^2.$$

Y también satisface la condición (2.23) pues

$$|\mu|^2 + |\sigma|^2 = \mu^2 + \sigma^2 \leq (\mu^2 + \sigma^2)(1 + |x|^2).$$

Por lo tanto, la solución  $S_t = \mu t + \sigma W_t$  es la única solución fuerte que nos garantiza el Teorema 2.6.1.

Algo que es importante resaltar de (2.25) es que  $\mu$ , que es el coeficiente asociado a  $dt$ , aparece en  $\mathbb{E}(S_t) = \mu t$  y de manera análoga,  $\sigma$  está asociado con  $\text{Var}(S_t) = \sigma^2 t$ . La anterior es una observación muy importante ya que nos permite darnos cuenta que podemos observar la EDE para obtener información acerca de la dinámica que sigue el proceso solución pues el coeficiente asociado a  $dt$  está directamente relacionado con la esperanza en un intervalo infinitesimal de tiempo y análogamente, el coeficiente asociado a  $dW_t$  impacta en la varianza infinitesimal del proceso solución.

**Ejemplo 2.6.2** (Movimiento Browniano Geométrico). Sea  $(W_t)_{\geq 0}$  un MB. Al proceso solución de la EDE

$$dS_t = \mu S_t dt + \sigma S_t dW_t, \quad \mu \in \mathbb{R}, \quad \sigma > 0, \quad (2.26)$$

se le conoce como *Movimiento Browniano Geométrico* (o MBG) y es ampliamente usada en el ámbito financiero debido a que su media de crecimiento y variabilidad, en intervalos infinitesimalmente pequeños, son proporcionales al valor del proceso y es el comportamiento que se esperaría que tengan ciertos valores, por ejemplo, las acciones. Para asegurarnos que dicha solución existe, hacemos la identificación de coeficientes para verificar el Teorema 2.6.1. Tenemos que  $\alpha(t, x) = \mu x$  y  $\beta(t, x) = \sigma x$ . Entonces, tomando  $K = \mu^2 + \sigma^2$  tenemos que se satisface (2.22)

$$|\mu x - \mu y|^2 + |\sigma x - \sigma y|^2 = \mu^2 |x - y|^2 + \sigma^2 |x - y|^2 \leq (\mu^2 + \sigma^2) |x - y|^2,$$

y también se cumple (2.23)

$$|\mu x|^2 + |\sigma x|^2 = \mu^2 |x|^2 + \sigma^2 |x|^2 = (\mu^2 + \sigma^2) |x|^2 \leq (\mu^2 + \sigma^2) (1 + |x|^2).$$

Por lo tanto, se tiene que la solución fuerte existe. Para encontrar el proceso solución, aplicamos el Lema de Itô 2.5.4 a la EDE (2.26) con  $f(t, x) = \ln(x)$ . Tenemos que

$$\frac{\partial f}{\partial t}(t, x) = 0, \quad \frac{\partial f}{\partial x}(t, x) = \frac{1}{x} \quad \text{y} \quad \frac{\partial^2 f}{\partial x^2}(t, x) = -\frac{1}{x^2},$$

así, por (2.20)  $\ln(S_t)$  tiene la representación

$$d \ln(S_t) = \frac{1}{S_t} \mu S_t dt - \frac{1}{2} \frac{1}{S_t^2} \sigma^2 S_t^2 dt + \frac{1}{S_t} \sigma S_t dW_t = \left( \mu - \frac{1}{2} \sigma^2 \right) dt + \sigma dW_t.$$

Por lo tanto,  $\ln(S_t) - \ln(S_0)$  es un MB con Deriva, entonces

$$\ln(S_t) - \ln(S_0) = \ln\left(\frac{S_t}{S_0}\right) = \int_0^t \left( \mu - \frac{1}{2} \sigma^2 \right) du + \int_0^t \sigma dW_u = \left( \mu - \frac{1}{2} \sigma^2 \right) t + \sigma W_t.$$

Así, el proceso solución es

$$S_t = S_0 \exp \left( \left( \mu - \frac{1}{2} \sigma^2 \right) t + \sigma W_t \right). \quad (2.27)$$

Una observación detallada de los procesos solución de los Ejemplos 2.6.1 y 2.6.2 revelará que ambos pueden escribirse en la forma  $S_t = g(t, W_t)$  para una función  $g : \mathbb{R}^+ \times \mathbb{R} \rightarrow \mathbb{R}$  adecuada. Lo anterior será de gran ayuda para simular el proceso solución cuando sea posible hallar una expresión analítica de éste.



# Capítulo 3

## Simulación e Inferencia para PSM y EDE

El objetivo de este capítulo es utilizar parte de la teoría y conceptos proporcionados en el Capítulo 2 para desarrollar algoritmos tanto de simulación como de estimación, ya sea para Procesos de Saltos de Markov o para ecuaciones diferenciales estocásticas. Dichos algoritmos serán de vital importancia en la tesis, pues debido a la naturaleza del tema, será necesario aplicar los métodos usando poder computacional pues la complejidad de los cálculos no permite desarrollar éstos de forma eficiente si no es a través de un lenguaje de programación.

Todos los Algoritmos utilizados pueden encontrarse en el lenguaje de programación R dentro del Apéndice A.

### 3.1. Simulación de Procesos de Saltos de Markov

En la primera parte de esta sección se presentarán dos algoritmos base para simular trayectorias provenientes de un Proceso de Saltos de Markov dado un Generador Infinitesimal  $Q$  y una distribución inicial  $\pi$ . Cabe destacar que el primero de estos dos algoritmos será de gran importancia pues se utilizará posteriormente en la Sección 3.2.2.

En la segunda parte, se abordará la simulación de distribuciones Tipo Fase y se verá que, debido a su conexión con los Procesos de Saltos de Markov con estados absorbentes, éstas pueden ser simuladas al modificar uno de los algoritmos presentados en la primera parte. También se reafirmará la coherencia de la representación de las distribuciones Tipo Fase vía los PSM mediante la comparación de la esperanza calculada de forma analítica con la obtenida mediante una estimación MonteCarlo.

### 3.1.1. PSM Estándar

Para poder simular un PSM es necesario recordar que este tipo de procesos pueden contener estados absorbentes y que no importa si el proceso es estándar o no, éste queda caracterizado por su Generador  $Q$  y su distribución inicial. Además, recordemos que el Generador  $Q$  satisface las propiedades enunciadas en la Proposición 2.3.5. Es necesario poner énfasis en las propiedades (2.4) y (2.6) pues se utilizarán para que la ley aleatoria del proceso simulado sea la de un PSM con las características deseadas.

Supongamos que queremos simular una trayectoria del PSM estándar con Generador  $Q$  y distribución inicial  $\pi$ , desde  $t = 0$  hasta  $t = T$ , entonces se usa el Algoritmo 1, el cual se desarrolló utilizando las propiedades del Generador expuestas en la Proposición 2.3.5. El Código A.1 muestra el algoritmo en el lenguaje de programación R.

---

**Algoritmo 1** Simulación de un PSM estándar hasta el horizonte de tiempo  $T$

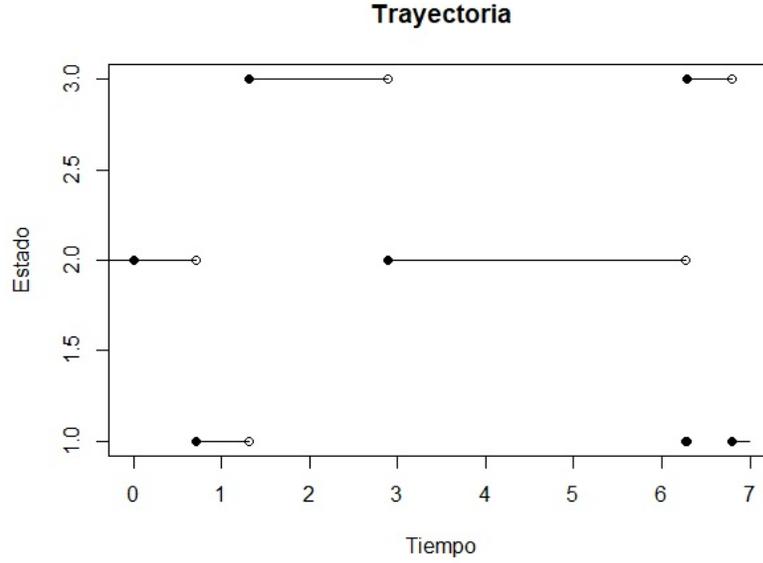
---

1. Elegir el estado inicial  $i$  con distribución  $\pi$ .
  2. Simular un tiempo de estancia  $\tau_i$  usando (2.6).
  3. Si ya se rebasó el horizonte de tiempo  $T$ , terminar. En caso contrario ir al paso 4.
  4. Obtener el estado  $j$  al que saltará el proceso cuando el tiempo  $\tau_i$  termine, usando (2.4).
  5. Hacer  $i = j$  y regresar al paso 2.
- 

La Figura 3.1, muestra una trayectoria simulada con el Algoritmo 1, horizonte de tiempo  $T = 7$ , tomando

$$\pi = \left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right), Q = \begin{pmatrix} -1 & 0.5 & 0.5 \\ 0.5 & -1 & 0.5 \\ 0.5 & 0.5 & -1 \end{pmatrix}. \quad (3.1)$$

El hincapié que se hace en el Algoritmo 1 acerca de que funciona exclusivamente para PSM estándar, es debido a que si tenemos uno o más estados absorbentes, puede ocurrir que durante la simulación, el PSM caiga en algún estado absorbente y una vez que eso pase, no tiene sentido seguir aplicando dicho algoritmo, pues al ser un estado absorbente, el PSM pierde su aleatoriedad desde ese momento, ya que de acuerdo a la Definición 2.3.2, el proceso se queda en dicho estado permanentemente.



**Figura 3.1:** Simulación de la Trayectoria de un PSM con parámetros como en (3.1) hasta el tiempo  $T = 7$ .

### 3.1.2. PSM con Estados Absorbentes

Si nuestro proceso tiene al menos un estado absorbente entonces, como se explicó anteriormente, la simulación de dicho proceso no puede hacerse usando el Algoritmo 1 pues existe la posibilidad de que el proceso salte al estado absorbente antes de que termine el horizonte de tiempo. Por lo tanto, será de nuestro interés poder simular una trayectoria del proceso hasta el momento en el cual éste salta a un estado absorbente, para lo cual es necesario hacer modificaciones al Algoritmo 1.

Si queremos simular la trayectoria del proceso hasta el momento de la absorción, podemos usar el Algoritmo 2. Éste, al igual que el Algoritmo 1, hace uso de las propiedades del Generador que se enuncian en la Proposición 2.3.5. El Código A.2 muestra dicho algoritmo en el lenguaje de programación R.

La Figura 3.2, muestra la simulación de la trayectoria de un proceso con un estado absorbente, hasta el tiempo de absorción usando el Algoritmo 2, con

$$\pi = \left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}, 0\right), Q = \begin{pmatrix} -7/3 & 1 & 1 & 1/3 \\ 1 & -7/3 & 1 & 1/3 \\ 1 & 1 & -7/3 & 1/3 \\ 0 & 0 & 0 & 0 \end{pmatrix}. \quad (3.2)$$

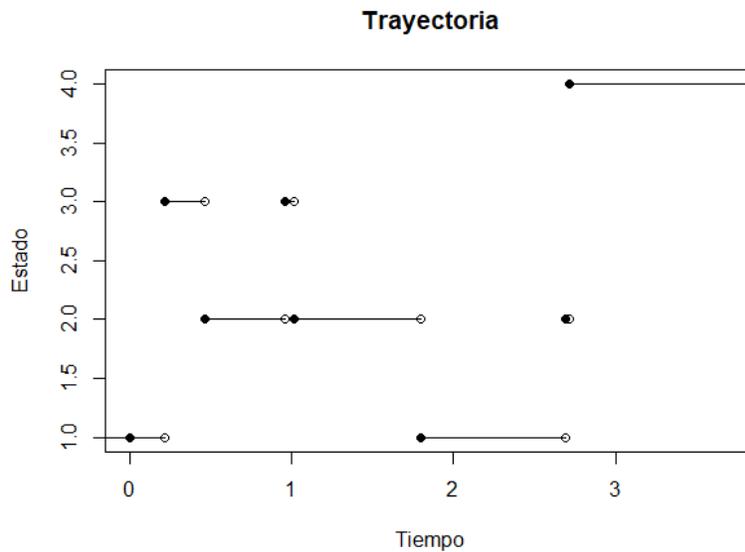
De (3.2) podemos observar que  $Q$  puede escribirse como en (2.7). Pues al tratarse de

---

**Algoritmo 2** Simulación de un PSM hasta el tiempo de absorción
 

---

1. Elegir el estado inicial  $i$  con distribución  $\pi$ .
  2. Si  $i$  es un estado absorbente, terminar. En caso contrario continuar con el siguiente paso.
  3. Simular un tiempo de estancia  $\tau_i$  usando (2.6).
  4. Obtener el estado  $j$  al que saltará el proceso cuando el tiempo  $\tau_i$  termine, usando (2.4).
  5. Si  $j$  es un estado absorbente, terminar. En caso contrario, hacemos  $i = j$  y regresamos al paso 3.
- 



**Figura 3.2:** Simulación de la Trayectoria de un PSM con parámetros como en (3.2) hasta el tiempo de absorción.

un PSM con un estado absorbente, recordemos que existe una forma de representar al tiempo de absorción como el valor de una distribución Tipo Fase.

### 3.1.2.1. Distribuciones Tipo Fase

La simulación de valores provenientes de una distribución Tipo Fase puede hacerse fácilmente adaptando el Algoritmo 2. Así, suponiendo que queremos simular un valor

$\tau$  de una distribución  $PH(\pi, \Lambda)$ , se usará el Algoritmo 3, la efectividad de este se encuentra respaldada por la Proposición 2.3.5. El Código A.3 muestra el algoritmo en el lenguaje de programación R.

---

**Algoritmo 3** Simulación de un valor proveniente de una distribución Tipo Fase

---

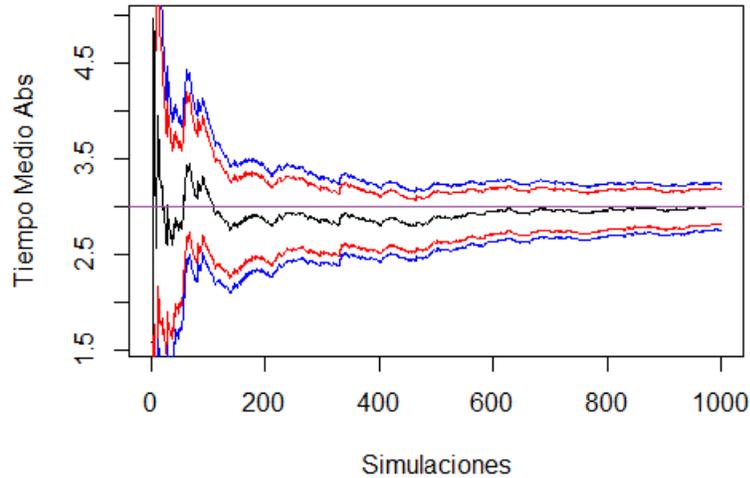
1. Usar (2.7) junto con  $\Lambda$  para obtener  $Q$ .
  2. Hacer  $\tau = 0$ .
  3. Elegir el estado inicial  $i$  con distribución  $\pi$ .
  4. Si  $i$  es un estado absorbente, regresar  $\tau$  y terminar. En caso contrario continuar con el siguiente paso.
  5. Simular un tiempo de estancia  $\tau_i$  usando (2.6).
  6. Actualizar  $\tau = \tau + \tau_i$ .
  7. Obtener el estado  $j$  al que saltará el proceso cuando el tiempo  $\tau_i$  termine, usando (2.4).
  8. Si  $j$  es un estado absorbente, regresar  $\tau$  y terminar. En caso contrario, hacemos  $i = j$  y regresamos al paso 3.
- 

Debido a que simular valores provenientes de una distribución Tipo Fase, no supone más que una modificación del Algoritmo 2, otro valor interesante que podemos calcular tanto de forma analítica como usando simulación, es el tiempo medio de absorción para una  $\Lambda$  y  $\pi$  específicos. Para el cálculo mediante simulación, podemos hacerlo mediante el método MonteCarlo y el Algoritmo 3.

Como ejemplo, para comparar el tiempo medio de absorción estimado mediante Monte Carlo con el valor real, usaremos

$$\pi = \left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}, 0\right), Q = \begin{pmatrix} -7/3 & 1 & 1 & 1/3 \\ 1 & -7/3 & 1 & 1/3 \\ 1 & 1 & -7/3 & 1/3 \\ 0 & 0 & 0 & 0 \end{pmatrix}. \quad (3.3)$$

La Figura 3.3 muestra el resultado de la estimación usando MonteCarlo con 1000 simulaciones. Las líneas rojas son intervalos al 95% de confianza, las azules al 99% de confianza y la línea lila es el valor analítico calculado usando (2.8) con  $n = 1$ .



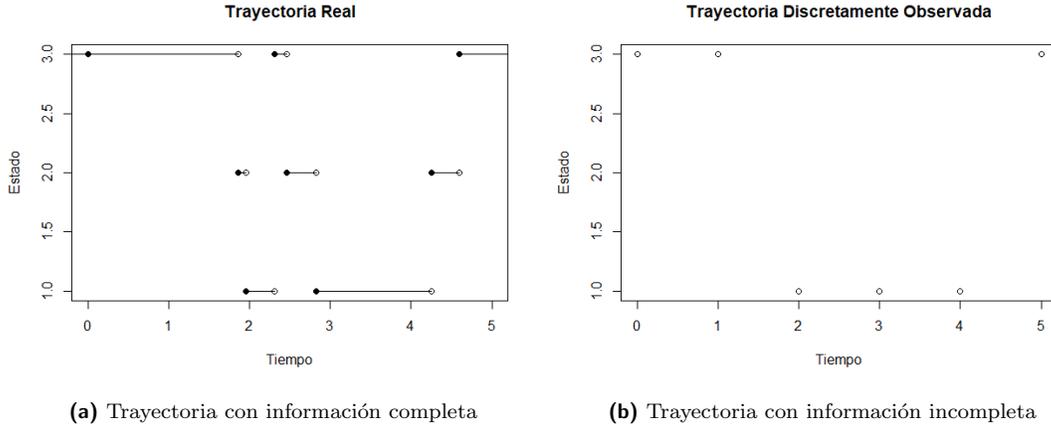
**Figura 3.3:** Comparación de la estimación del tiempo de absorción del ejemplo con parámetros en (3.3) con la esperanza  $\mathbb{E}(\tau)$  en su forma analítica.

### 3.1.3. Puentes Markovianos

Supongamos que se cuenta con una realización de un PSM. Sin embargo, en vez de contar con una trayectoria como la que se muestra en la Figura 3.1 o en la Figura 3.2, se cuenta con un subconjunto discreto de ésta como la que se muestra en la Figura 3.4. Tener información incompleta acerca de la trayectoria de un proceso, limita lo que se puede hacer con dicha trayectoria. En esta sección introduciremos métodos para simular la información faltante, con el fin de poder completar la trayectoria cuando se tiene una observación de la trayectoria a tiempo discreto.

Para comparar los resultados proporcionados por los algoritmos que presentaremos en esta sección, usaremos la misma información de una trayectoria continua, recopilada a tiempo discreto. La Figura 3.4 muestra los datos que utilizaremos. Los métodos que usaremos para completar la información faltante son:

1. Método de la Bisección: Destaca debido a que sirve para cualquier PSM. Sin embargo, su implementación es muy difícil.
2. Método de Aceptación-Rechazo: Este es el más fácil de implementar de todos. Pero en ciertos casos, su tiempo de término puede ser muy superior al de los demás algoritmos.
3. Método de Uniformización: En la mayoría de los casos es el algoritmo más veloz



**Figura 3.4:** Trayectoria Real e información a tiempo discreto que usarán los algoritmos para la reconstrucción.

de todos. No obstante, depende de la velocidad con la cual se simula el valor de una variable aleatoria programada por el usuario.

A las trayectorias reconstruidas por cualquier método les llamaremos **Puentes Markovianos**.

El objetivo de esta sección es introducir los métodos anteriores para construir Puentes Markovianos cuando tenemos información incompleta. Esto, como se mencionó al inicio de esta sección, nos servirá para no estar limitados por la información faltante cuando esta se presente.

### 3.1.3.1. Método de la Bisección

Este algoritmo se llama así porque bisecta los intervalos donde no conocemos la trayectoria del proceso, es decir, divide a la mitad los intervalos formados por cada observación discreta. Es un algoritmo recursivo, por lo que necesitamos casos base. Supongamos que tenemos dos observaciones discretas sucesivas en los tiempos  $t_1$  y  $t_2$  ( $t_1 < t_2$ ) y queremos completar la trayectoria entre esas observaciones. Los casos base se obtienen de las siguientes dos observaciones fundamentales:

1. Si  $X_{t_1} = a = X_{t_2}$  y el proceso no tiene saltos en el intervalo  $(t_1, t_2)$ .  
Entonces tenemos que  $X_t = a \forall t \in (t_1, t_2)$ .
2. Si  $X_{t_1} = a \neq b = X_{t_2}$  y el proceso tiene un solo salto en el intervalo  $(t_1, t_2)$ .  
Entonces tenemos que, para algún  $\tau \in (t_1, t_2)$ ,  $X_t = a$  para  $t_1 \leq t < \tau$  y  $X_t = b$  para  $\tau \leq t \leq t_2$ .

Recordemos que nuestros PSM son homogéneos. Así que, para simplificar la notación,

condideraremos sólo el intervalo  $(0, T)$  en lugar de  $(t_1, t_2)$ . Donde, obviamente  $(0, T)$  tiene la misma longitud que el intervalo original. Es decir  $T = t_2 - t_1$ . Así, la densidad de  $\tau$  se puede obtener mediante el siguiente lema cuya prueba puede encontrarse en [4].

**Lema 3.1.1** (Densidad del Tiempo de Salto). *Supongamos que tenemos un intervalo de longitud  $T$  con  $X_0 = a \neq b = X_T$ . La probabilidad de que sólo haya un salto en el intervalo (necesariamente de  $a$  a  $b$ ) está dada por*

$$R_{ab}(T) = q_{ab} \begin{cases} \frac{e^{-q_a T} - e^{-q_b T}}{q_b - q_a} & \text{si } q_a \neq q_b, \\ T e^{-q_a T} & \text{si } q_a = q_b. \end{cases} \quad (3.4)$$

Y la densidad del tiempo de salto (de  $a$  a  $b$ ) es

$$f_{ab}(t; T) = \frac{q_{ab} e^{-q_b T}}{R_{ab}(T)} e^{-(q_a - q_b)t}, \quad 0 \leq t \leq T. \quad (3.5)$$

Más aún, la probabilidad de que  $X_T = b$  y haya al menos dos saltos en el intervalo  $(0, T)$  es  $p_{ab}(T) - R_{ab}(T)$ .

Es importante resaltar, que si  $q_a = q_b$  entonces la densidad de cambio de estado es  $Unif \sim (0, T)$ , si  $q_a < q_b$  entonces la densidad es la de una variable aleatoria exponencial con parámetro  $q_b - q_a$  condicionada al intervalo  $(0, T)$  y si  $q_a > q_b$  entonces, por simetría, se tiene que  $f_{ab}$  es la densidad de la variable aleatoria  $T - V$ , donde  $V$  es una variable aleatoria exponencial con parámetro  $q_a - q_b$  condicionada al intervalo  $(0, T)$ .

Supongamos que  $X_0 = a = X_T$ . El Cuadro 3.1 muestra las probabilidades de los posibles casos para los saltos. Para simplificar la notación, denotamos

$$e_a = e^{-q_a T/2}, \quad r_{ab} = R_{ab}(T/2), \quad p_{ab} = p_{ab}(T/2).$$

En el Cuadro 3.1 vemos que en los primeros 4 casos tenemos que  $X_{T/2} = a$ , mientras que si caemos en los casos adicionales, tenemos que  $X_{T/2} = c \neq a$ .

Por otro lado, si nuestro estado inicial y final son distintos, es decir  $X_0 = a \neq b = X_T$ , entonces el Cuadro 3.2, muestra las siguientes probabilidades para los saltos. De nuevo, observemos que en los primero 4 casos tenemos que  $X_{T/2} = a$ , mientras que los siguientes 4 casos se cumple  $X_{T/2} = b$  y si caemos en alguno de los casos adicionales, entonces  $X_{T/2} = c \notin \{a, b\}$ .

Caso	Saltos en el primer intervalo	Saltos en el segundo intervalo	Probabilidad	Notación
1	0	0	$e_a e_a$	$\alpha_1$
2	0	$\geq 2$	$e_a(p_{aa} - e_a)$	$\alpha_2$
3	$\geq 2$	0	$(p_{aa} - e_a)e_a$	$\alpha_3$
4	$\geq 2$	$\geq 2$	$(p_{aa} - e_a)(p_{aa} - e_a)$	$\alpha_4$
5	1	1	$r_{ac}r_{ca}$	$\alpha_{5,c}$
6	1	$\geq 2$	$r_{ac}(p_{ca} - r_{ca})$	$\alpha_{6,c}$
7	$\geq 2$	1	$(p_{ac} - r_{ac})r_{ca}$	$\alpha_{7,c}$
8	$\geq 2$	$\geq 2$	$(p_{ac} - r_{ac})(p_{ca} - r_{ca})$	$\alpha_{8,c}$

**Cuadro 3.1:** Escenarios posibles del número de saltos cuando los puntos  $X(0) = a$  y  $X(T) = a$  son el mismo.

Caso	Saltos en el primer intervalo	Saltos en el segundo intervalo	Probabilidad	Notación
1	0	1	$e_a r_{ab}$	$\beta_1$
2	0	$\geq 2$	$e_a(p_{ab} - r_{ab})$	$\beta_2$
3	$\geq 2$	1	$(p_{aa} - e_a)r_{ab}$	$\beta_3$
4	$\geq 2$	$\geq 2$	$(p_{aa} - e_a)(p_{ab} - r_{ab})$	$\beta_4$
5	1	0	$r_{ab}e_b$	$\beta_5$
6	1	$\geq 2$	$r_{ab}(p_{bb} - e_b)$	$\beta_6$
7	$\geq 2$	0	$(p_{ab} - r_{ab})e_b$	$\beta_7$
8	$\geq 2$	$\geq 2$	$(p_{ab} - r_{ab})(p_{bb} - e_b)$	$\beta_8$
9	1	1	$r_{ac}r_{cb}$	$\beta_{9,c}$
10	1	$\geq 2$	$r_{ac}(p_{cb} - r_{cb})$	$\beta_{10,c}$
11	$\geq 2$	1	$(p_{ac} - r_{ac})r_{cb}$	$\beta_{11,c}$
12	$\geq 2$	$\geq 2$	$(p_{ac} - r_{ac})(p_{cb} - r_{cb})$	$\beta_{12,c}$

**Cuadro 3.2:** Escenarios posibles del número de saltos cuando los puntos  $X(0) = a$  y  $X(T) = b \neq a$  son distintos.

Para entender el Algoritmo de la Bisección debemos explicar los dos pasos sobre los cuales se sostiene. Para ello, explicaremos ambos pasos con un ejemplo. Supongamos que queremos construir un puente con  $X_0 = a$  y  $X_T = b$ .

En el **Paso de Inicialización** del algoritmo de la bisección: Dependiendo de si  $a = b$  o  $a \neq b$ , elegimos uno de los casos  $k$  del cuadro correspondiente (Cuadro 3.1 si  $a = b$  o Cuadro 3.2 si  $a \neq b$ ) con una probabilidad proporcional al valor de  $\alpha_k$  o  $\beta_k$ , según corresponda. Una vez elegido el caso, construimos el puente de acuerdo a las especificaciones de éste.

Por ejemplo, si tenemos  $X_0 = a = b = X_T$  y seleccionamos el caso con probabilidad  $\alpha_1$ , entonces tenemos que  $X_t = a \forall t \in [0, T]$  y terminamos, porque no hubo saltos. Para  $c \neq a$ , si elegimos el caso con probabilidad  $\alpha_{5,c}$ , entonces:  $X_t = a \forall t \in [0, \tau_1) \cup [\tau_2, T]$  y  $X_t = c \forall t \in [\tau_1, \tau_2)$ , donde  $\tau_1$  y  $\tau_2$  son valores que podemos obtener de (3.5) y terminamos.

Por otro lado, si tenemos que  $X_0 = a \neq b = X_T$  y seleccionamos el caso con probabilidad  $\beta_5$ , entonces:  $X_t = a \forall t \in [0, \tau)$  y  $X_t = b \forall t \in [\tau, T]$ . Para  $c \notin \{a, b\}$ , si elegimos el caso con probabilidad  $\beta_{9,c}$ , entonces:  $X_t = a \forall t \in [0, \tau_1)$ .  $X_t = c \forall t \in [\tau_1, \tau_2)$  y  $X_t = b \forall t \in [\tau_2, T]$ , donde  $\tau$ ,  $\tau_1$  y  $\tau_2$  son valores que obtenemos usando (3.5) y terminamos.

Si el caso que elegimos, es uno donde tenemos al menos dos saltos en alguna (por ejemplo  $\alpha_2$  ó  $\beta_6$ ) o en ambas mitades (como  $\alpha_4$  ó  $\beta_8$ ), completamos la mitad del intervalo con menos de dos saltos (según sea el caso) y le aplicamos el Paso de Recursión a la otra mitad del intervalo. En caso de que ambas mitades tengan 2 o más saltos, aplicamos el Paso de Recursión a ambas mitades.

En el **Paso de Recursión** del algoritmo de la bisección:

Como tenemos dos o más saltos, podemos usar los mismos Cuadros que en el Paso de Inicialización, sin embargo debemos condicionar dichos Cuadros a los casos en los cuales se tienen al menos dos saltos, es decir, en el Cuadro 3.1 debemos eliminar el primer caso y en el Cuadro 3.2 debemos eliminar la posibilidad de tomar el caso 1 y el caso 5. Elegimos uno de los casos restantes con probabilidad proporcional a  $\alpha$  o  $\beta$ , según sea el caso y hacemos lo mismo que en el Paso de Inicialización, es decir, dependiendo de qué caso elijamos, podemos completar alguna o ambas mitades del intervalo y volvemos a aplicar el Paso de Recursión a alguna o ambas mitades si es que así se requiere.

Ya con ambos pasos explicados, podemos usar el Algoritmo 4 para construir un puente entre dos observaciones consecutivas del proceso ( $X(0) = a, X(T) = b$ ). El Código A.8 muestra como implementar el algoritmo en el lenguaje de programación R.

### 3.1.3.2. Método de Aceptación-Rechazo

El método de Aceptación-Rechazo es un algoritmo mucho más sencillo de implementar que el algoritmo de la bisección. Éste aprovecha la ventaja de conocer  $Q$ ; para poder simular trayectorias de un Proceso de Saltos de Markov. En esencia, simularemos trayectorias usando  $Q$ , aceptaremos una trayectoria si ésta pasa por los puntos observados a tiempo discreto y la rechazaremos en caso contrario.

---

**Algoritmo 4** Algoritmo de la Bisección

---

1. Aplicamos el **Paso de Inicialización** al intervalo  $(0, T)$ . Si hay menos de dos saltos en ambos subintervalos  $(0, T/2), (T/2, T)$  terminamos. Si hay dos o más saltos en ambos subintervalos, vamos al paso 2. Si hay dos o más saltos en exactamente uno de los subintervalos, vamos al paso 3.
  2. Aplicamos el **Paso de Recursión** a ambos subintervalos y terminamos.
  3. Aplicamos el **Paso de Recursión** al subintervalo con dos o más saltos y terminamos.
- 

Supongamos que tenemos observaciones del Proceso de Saltos de Markov  $(X_t)_{t \geq 0}$  con Generador Infinitesimal  $Q$ , en los tiempos  $\{t_1, t_2, \dots, t_n\}$  entonces usaremos el Algoritmo 5 para completar la información faltante entre dos observaciones consecutivas  $X_{t_i}, X_{t_{i+1}}$  mediante Aceptación-Rechazo. El Código A.9 muestra la implementación de dicho algoritmo en el lenguaje de programación R.

---

**Algoritmo 5** Algoritmo de Aceptación-Rechazo

---

1. Tomamos el intervalo  $[t_i, t_{i+1}]$ . Conocemos  $X_{t_i}$  y  $X_{t_{i+1}}$  (Supongamos que  $X_{t_i} = a$  y  $X_{t_{i+1}} = b$ ).
  2. Simulamos una trayectoria  $(Y_t)_{t_i \leq t \leq t_{i+1}}$  con Generador  $Q$  y  $\pi = (\pi_1, \dots, \pi_n)$  con  $\pi_a = 1$  usando el Algoritmo 1.
  3. Si  $Y_{t_{i+1}} = b$  entonces aceptamos la trayectoria, hacemos  $X_t = Y_t \forall t \in [t_i, t_{i+1}]$  y terminamos. Si  $Y_{t_{i+1}} \neq b$ , regresamos al paso 2.
- 

El Algoritmo 5 nos proporciona buenos resultados y es fácil de implementar, sin embargo, la condición de aceptación de una trayectoria puede llegar a incrementar el tiempo de término del algoritmo. Matemáticamente,

$$\mathbb{P}(X_{t_{i+1}-t_i} = b \mid X_0 = a) \quad (3.6)$$

es la probabilidad de aceptar una subtrayectoria simulada para el intervalo  $(t_i, t_{i+1})$ . Como cada simulación es independiente de todas las demás, tenemos un comportamiento Geométrico en el número de trayectorias simuladas antes de aceptar alguna de éstas. De (3.6), el número esperado de simulaciones realizadas antes de aceptar una trayectoria es

$$\frac{1}{\mathbb{P}(X_{t_{i+1}-t_i} = b \mid X_0 = a)};$$

dependiendo del semigrupo de transición, (3.6) puede ser pequeña o grande. Esto puede aumentar o reducir el número de intentos esperados antes de aceptar una trayectoria demorando el tiempo de término del programa.

### 3.1.3.3. Método de Uniformización

Recordemos que, por la propiedad (2.6), los saltos ocurren de acuerdo a una ley exponencial, esto nos da una conexión con los procesos Poisson donde los tiempos entre arribos tienen distribución exponencial. Más aún, si en el intervalo  $(0, T)$  tenemos  $N$  saltos, entonces los tiempos de llegada tienen la misma distribución que  $(U_{(1)}, \dots, U_{(N)})$ , donde cada  $U_{(i)}$  es el  $i$ -ésimo estadístico de orden de una muestra aleatoria de  $N$  variables Uniformes en  $(0, T)$ .

En el fondo, la idea del algoritmo es crear  $(Y_n)_{n \in \mathbb{N}}$  una Cadena de Markov a Tiempo Discreto a partir de nuestro Generador  $Q$  y asignarle (a una trayectoria de la cadena) tiempos de salto provenientes de un proceso Poisson con un parámetro específico.

La matriz de transición  $\Sigma$  de nuestra Cadena de Markov a Tiempo Discreto  $(Y_n)_{n \in \mathbb{N}}$  se define como:

$$\Sigma = I + \frac{Q}{\nu}, \quad (3.7)$$

donde  $I$  es la matriz identidad y  $\nu = \max\{q_1, \dots, q_m\}$ . Es decir,  $\nu$  es el máximo de los elementos de la diagonal (en valor absoluto) de nuestro generador infinitesimal  $Q$ .

De nuevo, vamos a suponer que contamos con observaciones a tiempo discreto del Proceso de Saltos de Markov  $(X_t)_{t \geq 0}$  con Generador Infinitesimal  $Q$ , en los tiempos  $\{t_1, t_2, \dots, t_n\}$ .

Entonces, podemos verificar en [5] que si  $N$  es la variable aleatoria que cuenta los saltos en el intervalo  $(t_j, t_{j+1})$ , la probabilidad de tener  $k$  saltos en dicho intervalo, condicionado a  $X_{t_j} = a$  y  $X_{t_{j+1}} = b$  (los valores conocidos en los extremos del intervalo) es:

$$\mathbb{P}(N = k \mid X_{t_j} = a, X_{t_{j+1}} = b) = \frac{e^{-\nu\Delta} (\nu\Delta)^k (\Sigma^k)_{a,b}}{k! p_{a,b}(\Delta)}; \quad (3.8)$$

donde  $\Delta$  es la longitud del intervalo  $(t_j, t_{j+1})$  y  $p_{a,b}(t)$  es la entrada  $a, b$  del semigrupo de transición de  $(X_t)_{t \geq 0}$  evaluado en  $t$ . Con todo lo anterior, ya podemos plantear el algoritmo para simular un puente para cada par de observaciones discretas consecutivas  $X_{t_j} = a, X_{t_{j+1}} = b$ .

El Algoritmo 6 nos proporciona una forma simple de completar los puentes, explotando las propiedades de los Procesos de Saltos de Markov que nos proporciona la

---

**Algoritmo 6** Algoritmo de Uniformización
 

---

1. Si  $X_{t_j} = a$  y  $a$  es un estado absorbente, entonces detenemos el algoritmo, en caso contrario vamos al siguiente paso.
  2. Simulamos  $N$  usando (3.8).
  3. Si  $N = 0$  entonces  $X_t = a \forall t \in [t_j, t_{j+1})$ ,  $X_t = b$  si  $t = t_{j+1}$ , pasamos al intervalo  $[t_{j+1}, t_{j+2}]$  y regresamos al paso 2. (Sólo es posible si  $a = b$ )
  4. Si  $N > 0$  vamos al siguiente paso.
  5. Simulamos  $U_1, \dots, U_N \sim Unif(t_j, t_{j+1})$  y obtenemos  $(U_{(1)}, \dots, U_{(N)})$ .
  6. Simulamos una trayectoria  $(Y_1, Y_2, \dots, Y_{N-1})$  del proceso  $(Y_n)_{n \in \mathbb{N}}$  con  $\pi_a = 1$  y  $\Sigma$  como en (3.7).
  7. Completamos la trayectoria de la siguiente forma:
 
$$X_t = a \text{ para } t \in [t_j, U_{(1)})$$

$$X_t = Y_i \text{ para } t \in [U_{(i)}, U_{(i+1)}) \text{ con } i \in \{1, \dots, N-1\}$$

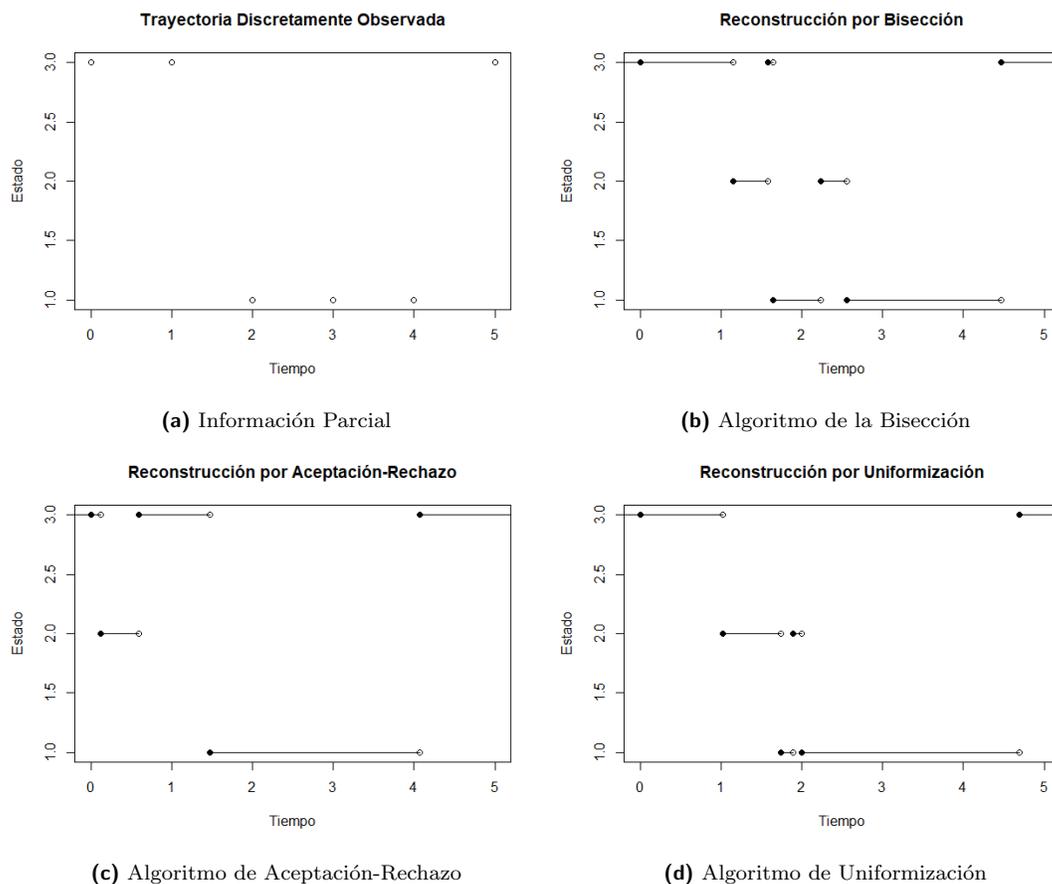
$$X_t = b \text{ para } t \in [U_{(N)}, t_{j+1}].$$
  8. Pasamos al intervalo  $[t_{j+1}, t_{j+2}]$  y regresamos al paso 1.
- 

teoría. El Código A.10 puede usarse para implementar el algoritmo con el lenguaje de programación R. Sin embargo, la debilidad de este algoritmo está en (3.8). Debido a que no es una distribución habitual, su simulación podría no ser óptima y retrasar mucho el tiempo de término del programa. Además, del numerador en la expresión de la derecha en (3.8) podemos observar que si  $\nu\Delta$  es muy grande, entonces la probabilidad de tener muchos saltos crece y el tiempo de término del programa se retrasa más.

### 3.1.3.4. Comparación de Resultados

En esta sección hacemos una comparación cualitativa de los resultados obtenidos para cada uno de los tres métodos de construcción de Puentes Markovianos presentados anteriormente. Por un lado, contrastaremos los tiempos de ejecución y, por otro lado, se presentan los Puentes Markovianos construidos en la Figura 3.5.

Dentro de la Figura 3.5, la primer gráfica (Figura 3.5a) muestra la información parcial que tenemos del PSM (misma información que en la Figura 3.4); mientras que las siguientes gráficas muestran los Puentes Markovianos construidos mediante los métodos de la Bisección, Aceptación-Rechazo y Uniformización (Figuras 3.5b, 3.5c y 3.5d, respectivamente). Además, los tiempos de término reportados fueron respectivamente 0.675653, 0.06513619 y 0.00693202 segundos.



**Figura 3.5:** Puentes construidos con los algoritmos utilizando la información parcial.

Como se puede observar, bajo las mismas condiciones de incompletitud en la información de una trayectoria proveniente de un PSM estándar, el mejor algoritmo

para completar la información faltante es el de Uniformización pues el tiempo que tardó dicho algoritmo en completar la información faltante en el ejemplo, es casi diez veces menor que mediante el método de Aceptación-Rechazo y casi cien veces menor que lo que tarda el método de la Bisección. Dicha velocidad será de gran importancia, pues como se verá más adelante, se requerirá completar la información faltante no sólo una, sino un gran número de veces. Por lo tanto, el método que se usará para este fin, será el de Uniformización.

## 3.2. Estimación del Generador de un Proceso de Saltos de Markov

En la sección anterior se introdujeron métodos de simulación de trayectorias de Procesos de Saltos de Markov cuando se conoce su Generador Infinitesimal y su Distribución Inicial. Ahora, supondremos que tenemos una trayectoria de un Proceso de Saltos de Markov estándar y nos interesará hacer inferencia acerca de su Generador Infinitesimal.

La sección está dividida en dos subsecciones. La primera de éstas está dedicada a la estimación del Generador Infinitesimal cuando se tiene conocimiento de toda la trayectoria a tiempo continuo del PSM. Es decir, se hará inferencia bajo la hipótesis de que el conjunto de observaciones de la trayectoria es de la forma  $\{X_t \mid t \in [0, T]\}$  con  $T < \infty$ . Mientras que la segunda sección tratará el tema de la estimación del Generador Infinitesimal cuando se tiene la observación de una trayectoria a tiempo discreto. Es decir, cuando se tiene que el conjunto de observaciones es discreto  $\{X_t \mid t \in \{t_1, \dots, t_n\} \text{ con } 0 = t_1 < \dots < t_n = T\}$  con  $T < \infty$ .

### 3.2.1. Estimación con Información Completa

Ya sabemos simular trayectorias de un proceso de Saltos de Markov si conocemos su Generador Infinitesimal y una Distribución Inicial. Ahora supongamos que tenemos el problema inverso, es decir, contamos con una trayectoria a tiempo continuo proveniente de un Proceso de Saltos de Markov y queremos estimar su Generador Infinitesimal.

Recordemos que si tenemos una muestra aleatoria  $x_1, x_2, \dots, x_n$  proveniente de una densidad  $f(x|\theta)$ ,  $\theta \in \Theta$  donde  $\theta$  es el parámetro desconocido o un vector de parámetros desconocidos en el espacio parametral  $\Theta$ , entonces la función de verosimilitud de la muestra está dada por

$$L(\theta|x) = \prod_{i=1}^n f(x_i|\theta). \quad (3.9)$$

Basándonos en lo anterior, vamos a definir notación para nuestro caso específico con PSM.

Sea  $(X_t)_{t \geq 0}$  un Proceso de Saltos de Markov con espacio de estados finito  $E = \{1, 2, \dots, m\}$  y Generador Infinitesimal  $Q = (q_{ij})_{i,j \in E}$ . Supongamos que tenemos una trayectoria del proceso hasta el tiempo  $T$ , es decir  $\{X_t \mid 0 \leq t \leq T\}$ . Entonces se puede consultar en [3] que la función de verosimilitud es:

$$L_T^{(c)}(Q) = \prod_{i=1}^m \prod_{j \neq i} q_{ij}^{N_{ij}(T)} \exp(-q_{ij} R_i(T)) \quad (3.10)$$

El superíndice  $(c)$  indica que la observación del proceso es a tiempo continuo,  $N_{ij}(t)$  denota el número de saltos del estado  $i$  al estado  $j$  en la trayectoria en el intervalo  $[0, t]$  y

$$R_i(t) = \int_0^t \mathbb{1}_{\{X_s=i\}} ds,$$

donde  $\mathbb{1}_A$  es la función indicadora del evento  $A$ . Por lo tanto  $R_i(t)$  es el tiempo que el proceso pasa en el estado  $i$  dentro del intervalo  $[0, t]$ .

Buscamos entonces el Estimador Máximo Verosímil para el Generador  $Q$ , el cual viene dado por el siguiente lema cuya prueba se encuentra en [3].

**Lema 3.2.1** (Estimador Máximo Verosímil). *El Estimador Máximo Verosímil (EMV) de un Generador Infinitesimal  $Q$  es:*

$$\hat{q}_{ij}^{(c)}(T) = \frac{N_{ij}(T)}{R_i(T)}.$$

Podemos observar que se requiere  $R_i(T) > 0 \forall i \in E$ , de otra forma, el Estimador Máximo Verosímil no existe. En otras palabras, se requiere que la trayectoria observada contenga observaciones de todos los estados del proceso. Una forma intuitiva de corroborar el resultado propuesto por el Lema 3.2.1 es mediante las siguientes observaciones:

**Observación 3.2.2.** *Como  $p_{ij}$  representa la probabilidad de transición del estado  $i$  al estado  $j$  cuando se efectúa un salto, entonces el estimador (para  $p_{ij}$ ) propuesto de forma natural sería*

$$\hat{p}_{ij}^{(c)}(T) = \frac{N_{ij}(T)}{N_i(T)},$$

donde  $N_i(T)$  es el número de visitas al estado  $i$  en el intervalo  $[0, T]$ .

**Observación 3.2.3.** *De la propiedad (2.6) se tiene que  $\tau_i \sim \text{Exponencial}(q_i)$ , por lo tanto, el Estimador Máximo Verosímil de  $q_i$  es*

$$\hat{q}_i^{(c)}(T) = \frac{N_i(T)}{R_i(T)}.$$

### 3.2. ESTIMACIÓN DEL GENERADOR DE UN PROCESO DE SALTOS DE MARKOV 37

**Observación 3.2.4.** De la propiedad (2.4) se tiene que  $p_{ij} = \frac{q_{ij}}{q_i}$ ; por lo tanto  $q_{ij} = q_i p_{ij}$ . Por lo que usando las Observaciones 3.2.2 y 3.2.3 se tiene que el estimador de  $q_{ij}$  es

$$\hat{q}_{ij}^{(c)}(T) = \hat{q}_i^{(c)}(T) \hat{p}_{ij}^{(c)}(T) = \frac{N_{ij}(T)}{R_i(T)}.$$

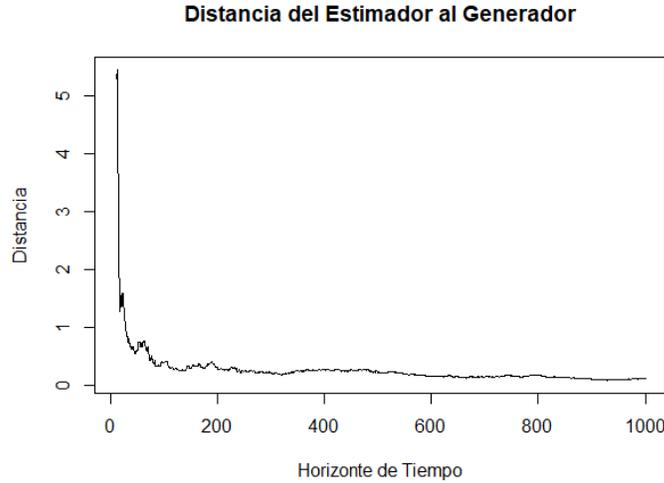
Así, podemos decir que el EMV de  $Q$  es

$$\hat{Q}^{(c)}(T) = (\hat{q}_{ij}^{(c)}(T))_{i,j \in E}. \quad (3.11)$$

Como el cálculo del EMV cuando se tiene información completa, solamente consta de calcular las cantidades que se muestran como resultado del Lema 3.2.1, no hace falta enunciar ningún algoritmo, sin embargo el Código A.6 muestra cómo calcular dicho estimador mediante el lenguaje de programación R. Es intuitivo pensar que conforme  $T$  aumenta, entonces nuestro estimador será mejor. Podemos hacer una prueba para medir la distancia entre el Generador Estimado  $\hat{Q}(t)$  y un Generador  $Q$  con la norma  $\|\cdot\|_1$  incrementando el valor de  $t$ . Para esto simularemos una trayectoria hasta  $T = 1000$ , usando el Algoritmo 1 con

$$\pi = \left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right), Q = \begin{pmatrix} -1 & 0.5 & 0.5 \\ 0.5 & -1 & 0.5 \\ 0.5 & 0.5 & -1 \end{pmatrix}, \quad (3.12)$$

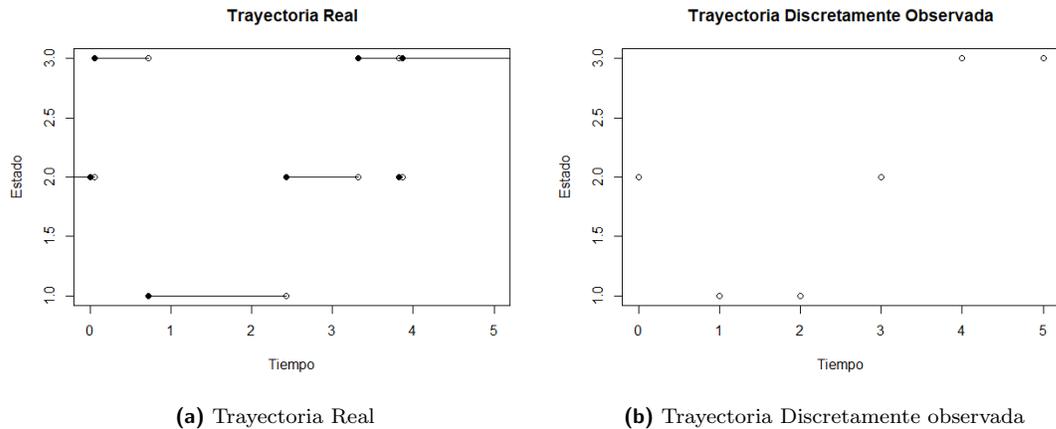
y calcularemos el estimador  $\hat{Q}(t)$  para  $t \in [10, T]$ ; finalmente calcularemos la norma  $\|Q - \hat{Q}(t)\|_1$  para  $t \in [10, T]$ . La Figura 3.6 muestra la gráfica de la norma anterior para  $t \in [10, T]$  con espaciamiento de tamaño 10.



**Figura 3.6:** Distancia de el estimador de  $Q$  usando (3.11) con valor real como en (3.12).

### 3.2.2. Estimación con Información Incompleta

Recordemos que los Procesos de Saltos de Markov, nos servirán para representar variables macroeconómicas que tienen influencia en la dinámica de movimiento de una acción. Sin embargo, la información que se registra en los históricos es el valor del proceso cada cierto tiempo (a veces no necesariamente de la misma longitud), por lo que tenemos información discretamente observada proveniente de un Proceso de Saltos de Markov con Generador conocido  $Q$ . Un ejemplo simple de esto, se muestra en la Figura 3.7.



**Figura 3.7:** Trayectoria real y la observación de la misma a tiempo discreto.

Suponiendo un proceso a tiempo continuo de fondo, nosotros tendremos solamente información parcial acerca de la trayectoria real, es decir, solamente tendremos el proceso discretamente observado en ciertos tiempos.

En la Sección 3.1.3 se introdujeron tres métodos con los cuales es posible reconstruir la trayectoria a tiempo continuo basándose en las observaciones del proceso a tiempo discreto. Sin embargo, aún no se cuenta con un algoritmo que pueda hacer uso de éstos métodos para hacer inferencia sobre el generador infinitesimal del PSM. Para ello se hará uso del Algoritmo Esperanza-Maximización. Dicho algoritmo es de vital importancia para esta tesis, razón por la cual se incluirá a continuación una breve introducción al método así como su adaptación al problema específico de estimación del Generador de un PSM con información faltante. Si se desea conocer más acerca del Algoritmo Esperanza-Maximización, se puede consultar [6].

#### 3.2.2.1. Algoritmo Esperanza-Maximización

Deseamos hallar el EMV del Generador Infinitesimal ( $\hat{Q}$ ) de un PSM. Esto se puede hacer de forma sencilla usando (3.11), sin embargo dicho estimador requiere

que conozcamos la trayectoria completa del PSM hasta un horizonte de tiempo  $T$ , es decir  $\{X_t \mid 0 \leq t \leq T\}$ . En vez de eso nosotros conocemos los valores del PSM  $(X_t)_{t \geq 0}$  sólo en un conjunto de tiempos  $\{t_1, \dots, t_n\}$  (que cumplen  $0 = t_1 < t_2 < \dots < t_n = T$ ) o dicho de otra manera, conocemos  $\{X_{t_1}, \dots, X_{t_n}\}$  que es información incompleta acerca de la trayectoria del PSM. Así, lo que podemos hacer es construir un Puente Markoviano con nuestra información incompleta utilizando alguno de los algoritmos vistos en la sección anterior, obtener el EMV  $\hat{Q}$  utilizando la información completada por medio del puente, utilizar dicho  $\hat{Q}$  para construir el siguiente Puente Markoviano y repetir el proceso.

Para la obtención del EMV del Generador Infinitesimal cuando tenemos información incompleta, usaremos un algoritmo llamado **Algoritmo Esperanza Maximización** o también conocido como **Algoritmo EM**.

El Algoritmo EM es un método iterativo para obtener estimadores máximo verosímiles de algún parámetro cuando tenemos información incompleta. Dicho algoritmo consta de dos pasos básicos que se repiten un número considerable de veces. Como en esta tesis utilizaremos el Algoritmo EM para calcular el EMV del Generador Infinitesimal de un PSM, solamente daremos una breve introducción, si el lector desea conocer más acerca de este algoritmo, puede consultar [6].

Recordemos que la función de verosimilitud de una muestra aleatoria se define como en (3.9). Entonces nos interesa encontrar el EMV usando información incompleta (observada) que denotaremos por  $\mathbf{y}$ , a partir de la función de verosimilitud de los datos completos que llamaremos  $\mathbf{x}$ . Ya que nuestra información no es completa, lo que haremos será calcular la esperanza condicional de dicha verosimilitud dado la información incompleta, es decir

$$S(\theta; \theta^{(k)}) = \mathbb{E}(L(\theta|\mathbf{x}) \mid \mathbf{y}; \theta^{(k)}), \quad (3.13)$$

donde  $k$  corresponde a la  $k$ -ésima iteración del algoritmo y  $\theta^{(k)}$  al  $k$ -ésimo estimador de  $\theta$ .

El primer paso del algoritmo es el **Paso E**, el cual se llama así porque consiste en calcular la esperanza (3.13). Al segundo paso del algoritmo se le llama **Paso M** debido a que se procede a maximizar la esperanza obtenida en el Paso E con el parámetro  $\theta$ . El Algoritmo 7 muestra la forma más general de plantear el Algoritmo EM. Es importante destacar que el algoritmo en esta forma, mejora la estimación de  $\theta$  con cada iteración, por lo que podemos hablar de una convergencia de los estimadores. De modo que una condición de paro que se usa habitualmente, es repetir el ciclo del Algoritmo 7 hasta que la mejora del estimador  $\theta^{(k)}$  sea despreciable con respecto a  $\theta^{(k-1)}$ , o matemáticamente

$$\left| L(\theta^{(k)}|\mathbf{x}) - L(\theta^{(k-1)}|\mathbf{x}) \right| < \epsilon, \quad p.a. \quad \epsilon > 0. \quad (3.14)$$

---

**Algoritmo 7** Algoritmo EM

---

1. Tomar un parámetro inicial  $\theta^{(0)}$ ,  $k = 0$
  2. **Paso E:** Calcular  $S(\theta; \theta^{(k)}) = \mathbb{E}(L(\theta|\mathbf{x}) | \mathbf{y}; \theta^{(k)})$
  3. **Paso M:** Encontrar  $\theta^{(k+1)} = \underset{\theta \in \Theta}{\text{arg máx}} S(\theta; \theta^{(k)})$
  4. Hacer  $k = k + 1$  y regresar al paso 2.
- 

A pesar de que el Algoritmo EM es sencillo de implementar, en ocasiones el problema de éste es que el cálculo de la esperanza en el Paso E, puede ser muy difícil de obtener analíticamente. Por eso, existe una variante del Algoritmo EM llamada **Algoritmo EM MonteCarlo** (o **Algoritmo EMMC**) el cual se diferencia con el primero en que, en vez de utilizar la esperanza del Paso E en su forma analítica, utiliza la estimación de dicha esperanza calculada mediante MonteCarlo, es decir

$$\hat{S}(\theta; \theta^{(k)}) = \frac{1}{r} \sum_{i=1}^r L(\theta|\mathbf{y}, z_i) \quad (3.15)$$

donde  $r$  es el número de observaciones simuladas de los datos completos a partir de los datos incompletos  $\mathbf{y}$ . Se puede consultar más acerca del Algoritmo EMMC en [6]. El Algoritmo 8 es el que utilizamos para obtener el EMV mediante este método. Una

---

**Algoritmo 8** Algoritmo EMMC

---

1. Tomar un parámetro inicial  $\theta^{(0)}$ ,  $k = 0$
  2. Generar una muestra  $z_1, \dots, z_r$  de tamaño  $r$  proveniente de la distribución de la información completa dado la información observada.
  3. **Paso E:** Calcular  $\hat{S}(\theta; \theta^{(k)}) = \frac{1}{r} \sum_{i=1}^r L(\theta|\mathbf{y}, z_i)$
  4. **Paso M:** Encontrar  $\theta^{(k+1)} = \underset{\theta \in \Theta}{\text{arg máx}} \hat{S}(\theta; \theta^{(k)})$
  5. Hacer  $k = k + 1$  y regresar al paso 2.
- 

de las ventajas del Algoritmo 8 es que, a diferencia del Algoritmo 7, no necesitamos hacer cálculos acerca de la esperanza, por lo que esto nos permite aplicar el Algoritmo 8 a una mayor variedad de situaciones. Sin embargo, el punto débil de éste es que, debido a que es muy general, el paso 2 incrementa considerablemente el tiempo de término del programa pues debe simular muchas muestras impactando directamente

en el tiempo de máquina.

Para amortiguar un poco el punto débil del Algoritmo EM Monte Carlo existe una variante donde únicamente se simula un valor proveniente de la distribución de la información completa dada la observada, es decir se aplica el Algoritmo 8 con  $r = 1$ . Esta variante del Algoritmo EM Monte Carlo se llama **Algoritmo EM Estocástico** (o **Algoritmo EME**). El Algoritmo 9 es el que se utiliza para obtener el EMV; una versión específica para nuestro caso se encuentran en el Algoritmo 10.

---

**Algoritmo 9** Algoritmo EME

---

1. Tomar un parámetro inicial  $\theta^{(0)}$ ,  $k = 0$
  2. Generar un valor  $z$  proveniente de la distribución de la información completa dado la información observada.
  3. **Paso E:** Calcular  $\hat{S}(\theta; \theta^{(k)}) = L(\theta | \mathbf{y}, z)$
  4. **Paso M:** Encontrar  $\theta^{(k+1)} = \arg \max_{\theta \in \Theta} \hat{S}(\theta; \theta^{(k)})$
  5. Hacer  $k = k + 1$  y regresar al paso 2.
- 

Un punto vital que debe tomarse en cuenta es que en el Algoritmo 9, al contener un componente estocástico, no se puede hablar de una convergencia como la del Algoritmo 7. En vez de eso, puede demostrarse que la sucesión  $(\theta^{(k)})_{k \in \mathbb{N}}$  de estimadores, es una Cadena de Markov cuyo espacio de estados es el espacio parametral  $\Theta$ . Más aún, dicha Cadena de Markov es irreducible y aperiódica, por lo tanto converge a la distribución estacionaria. Las demostraciones de todos estos resultados teóricos se pueden consultar en [6].

### 3.2.2.2. Estimación del Generador de un PSM con Información Incompleta Vía el Algoritmo EME

No se debe perder de vista que el objetivo principal de esta sección es el de desarrollar un algoritmo que ayude a estimar el Generador de un PSM estándar cuando la trayectoria tiene información faltante, es decir, cuando el conjunto de observaciones de la trayectoria es discreta.

A pesar de que ya contamos con la versión estocástica del Algoritmo EM, es necesario especificar cómo se usará éste para la estimación del Generador de un PSM estándar, razón por la cual es prudente identificar cuáles son los componentes del

algoritmo.

Primero, el parámetro que queremos estimar es un generador infinitesimal  $Q$  de un PSM, por lo tanto, en el primer paso del Algoritmo 9 el parámetro inicial será un Generador Infinitesimal  $Q^{(0)}$ . Luego, recordemos que en este caso la información faltante es la trayectoria a tiempo continuo, pues solamente contamos con observaciones discretas espaciadas a lo largo del intervalo  $[0, T]$ . Por lo tanto, en el siguiente paso del Algoritmo 9, generar un valor proveniente de la distribución de la información completa dado la información observada, se refiere a construir un Puente Markoviano (con alguno de los métodos vistos) usando las observaciones con las que contamos para completar la información faltante. Después, los Pasos 3 y 4 pueden resumirse en uno sólo debido a que el Lema 3.2.1 nos proporciona los EMV de  $(q_{ij})_{i,j \in E, i \neq j}$  cuando tenemos una trayectoria con información completa. Finalmente, usando (3.11) y (2.3) podemos construir la diagonal del generador infinitesimal.

Por lo tanto el Algoritmo 9 se modifica dando lugar al Algoritmo 10 para estimar el Generador Infinitesimal de un PSM estándar. La implementación del Algoritmo EME y el Algoritmo EMMC para estimar el generador infinitesimal de un PSM mediante el lenguaje de programación R se muestra en los Códigos A.12 y A.13 respectivamente.

---

**Algoritmo 10** Algoritmo EME para obtener el EMV de un PSM

---

1. Proponer un Generador inicial  $Q^{(0)}$ ,  $k = 0$ .
  2. Generar un Puente Markoviano  $\{\hat{X}_t \mid 0 \leq t \leq T\}$  utilizando la información incompleta  $\{X_{t_1}, \dots, X_{t_n}\}$  y  $Q^{(k)}$ .
  3. Usar  $\{\hat{X}_t \mid 0 \leq t \leq T\}$  para calcular  $\hat{q}_{ij}^{(c)}(T)$  de acuerdo al Lema 3.2.1 y construir  $\hat{Q}^{(c)}(T)$  usando (3.11) y (2.3).
  4. Hacer  $Q^{(k+1)} = \hat{Q}^{(c)}(T)$
  5. Hacer  $k = k + 1$  y regresar al paso 2.
- 

**3.2.2.2.1. Ejemplo** Para hacer comparables los resultados obtenidos del Algoritmo 10, usaremos una misma observación a tiempo discreto hasta  $T = 50, 75, 100, 200$  con  $n = 51, 76, 101, 201$  generada a partir de los parámetros mostrados en (3.12) y

### 3.2. ESTIMACIÓN DEL GENERADOR DE UN PROCESO DE SALTOS DE MARKOV43

además nuestro Generador inicial será el mismo para todos los métodos

$$Q^{(0)} = \begin{pmatrix} -1 & 0.667 & 0.333 \\ 0.333 & -1 & 0.667 \\ 0.667 & 0.333 & -1 \end{pmatrix}. \quad (3.16)$$

Como en este caso contamos con el Generador  $Q$  real, podemos calcular el error y el número de iteraciones que se hacen para obtener un estimador cercano a  $Q$ . Sin embargo, esto es sólo de carácter ilustrativo ya que, al analizar datos reales, no suelen conocerse los parámetros reales. Los resultados se resumen en el Cuadro 3.3. Por otro

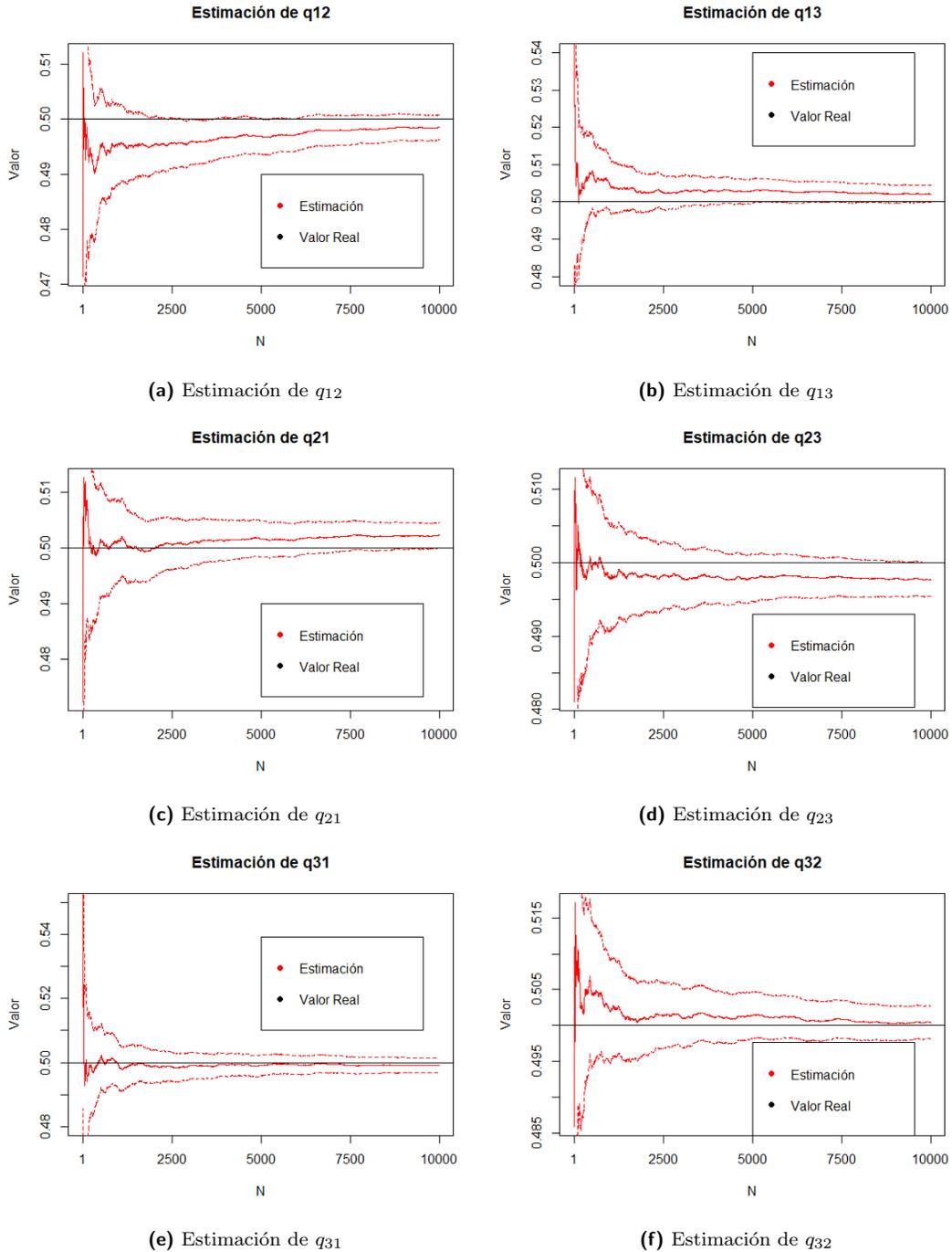
Horizonte ( $T$ )	Iteraciones	Error
100	87575	0.158175
200	9644	0.1155354
500	3457	0.1044876
1000	696	0.06940363

**Cuadro 3.3:** Resultados de las estimaciones.

lado, cuando no contemos con el parámetro real podemos aprovechar la propiedad de Markov y distribución estacionaria que nos proporciona el Algoritmo EME. Lo anterior se hará de la siguiente forma: primero dejaremos que el Algoritmo 10 se itere 1000 veces para dar oportunidad de que la Cadena de Markov llegue a su distribución estacionaria, los estimadores obtenidos durante este tiempo se desecharán; luego guardaremos los siguientes  $N$  estimadores que proporcione el algoritmo; finalmente, tomaremos el promedio de los  $N$  estimadores guardados y dicha media será nuestro estimador definitivo. Los resultados de este método de estimación se muestran en el Cuadro 3.4 y adicionalmente, en la Figura 3.8 se presentan los intervalos de confianza y la comparación con el valor real para el caso  $T = 1000$ .

Horizonte ( $T$ )	Error
100	0.2171750
200	0.1443540
500	0.1163810
1000	0.0440784

**Cuadro 3.4:** Resultados de las estimaciones sin ayuda del parámetro original.



**Figura 3.8:** Estimación del Generador para el caso donde no usamos el parámetro real con horizonte de tiempo  $T = 1000$  e intervalos de confianza al 95 % (línea punteada).

### 3.3. Simulación de Ecuaciones Diferenciales Estocásticas

Cuando simulamos un PSM tenemos la ventaja de que, a pesar de tratarse de un proceso estocástico a tiempo continuo, sus trayectorias son constantes a tramos por lo que, solamente necesitamos los tiempos en los cuales el proceso salta y la sucesión de estados que sigue el proceso para obtener una representación exacta de su trayectoria. A diferencia de los PSM, el MB y las EDE no pueden ser simulados para todos los puntos en un intervalo  $[0, T]$  ( $T > 0$ ), sin importar la magnitud de  $T$ . Esto es debido a que el MB no tiene variación finita por lo que, la cantidad de memoria necesaria para guardar la posición del MB para cualquier  $t \in [0, T]$  sería infinita.

Lo anterior no debe desanimarnos, si bien no podemos guardar la trayectoria completa de éstos procesos para un intervalo completo, lo que sí se podrá hacer será simular dichos procesos en un subconjunto finito de tiempos  $0 \leq t_1 < \dots < t_n \leq T$  tan grande como se quiera y esto, para los fines de esta tesis, será suficiente.

#### 3.3.1. Simulación del Movimiento Browniano

Podemos usar la Definición 2.4.1 para crear un algoritmo que simule trayectorias del MB  $(W_t)_{t \geq 0}$ . No hay que perder de vista que la independencia y estacionariedad del MB se cumple para los incrementos. Un error que se comete habitualmente es hacer caso omiso de la Propiedad 3 en la Definición 2.4.1 y usar la Propiedad 4 de la misma definición con  $s = 0$ . Si bien es correcto que  $W_t \sim Normal(0, t)$ , es incorrecto simular  $W_{t_1}, \dots, W_{t_n}$  como observaciones de variables aleatorias  $Normal(0, t_i)$  independientes.

El primer paso es recordar que de acuerdo con la propiedad 1 de la Definición 2.4.1 tenemos que  $W_0 = 0$ . Entonces para simular  $W_{t_1}$  observemos que

$$W_{t_1} = W_{t_1} + W_0 - W_0 = W_0 + (W_{t_1} - W_0).$$

También,

$$W_{t_2} = W_{t_1} + (W_{t_2} - W_{t_1}),$$

y así sucesivamente. Por la Propiedad 3, los incrementos  $W_{t_i} - W_{t_{i-1}}$  son independientes para cada  $i \in \{1, 2, \dots, n\}$  y podemos simularlos secuencialmente usando la propiedad 4 para obtener  $W_{t_1}, W_{t_2}, \dots, W_{t_n}$  en ese orden.

Supongamos que queremos generar una trayectoria del MB en un conjunto de tiempos  $0 = t_0 < t_1 < \dots < t_n = T$ , entonces el Algoritmo 11 simula una realización del proceso hasta el horizonte de tiempo  $T$  en nuestro conjunto de tiempos, el Código A.14 muestra dicho algoritmo implementado en el lenguaje de programación R. Es

usual que los tiempos en los cuales se observa el MB sean equidistantes entre sí; en ese caso denotaremos como  $\Delta t$  a dicha diferencia, es decir:  $\Delta t = |t_i - t_{i-1}|$ ,  $\forall i = 1, \dots, n$ .

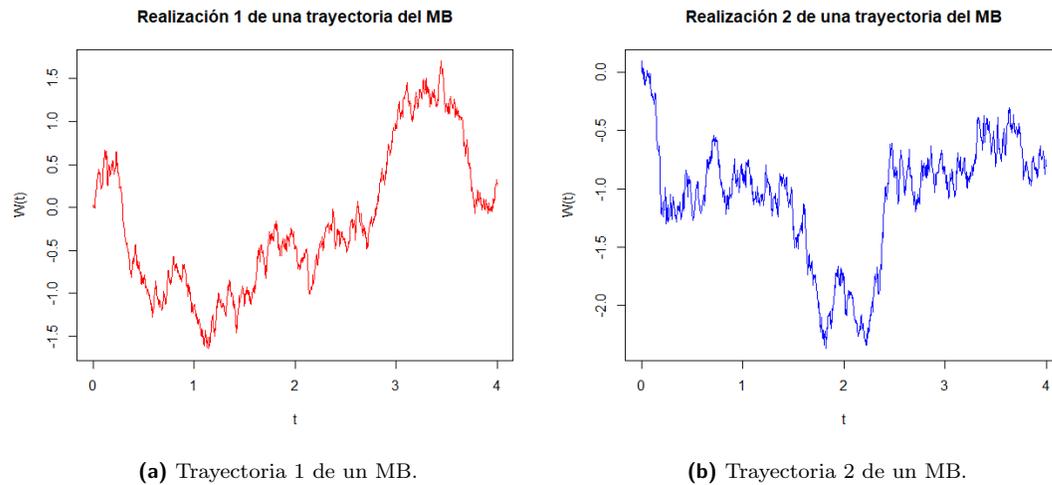
---

**Algoritmo 11** Simulación de un MB

---

1. Hacemos  $W_{t_0} = 0$ ,  $i = 1$ .
  2. Calculamos  $W_{t_i} = W_{t_{i-1}} + Z$ , donde  $Z \sim Normal(0, t_i - t_{i-1})$ .
  3.  $i = i + 1$ .
  4. Si  $i > n$  vamos al paso 5. En caso contrario regresamos al paso 2.
  5. Regresamos el vector  $(W_{t_0}, W_{t_1}, \dots, W_{t_n})$ .
- 

En la Figura 3.9 se muestran dos realizaciones del proceso simuladas mediante el Algoritmo 11.



**Figura 3.9:** Dos simulaciones del MB con  $\Delta t = \frac{1}{252}$  y  $T = 4$ .

### 3.3.2. Simulación de EDE Vía su Solución

Como se mencionó al final de la Sección 2.6, cuando tenemos una expresión analítica para la solución de una EDE, es posible aprovechar dicha expresión para simular trayectorias del proceso solución. En esta sección se retomarán los Ejemplos 2.6.1 y 2.6.2 para desarrollar algoritmos de simulación de los procesos solución utilizando las expresiones analíticas que se obtuvieron en dichos ejemplos.

En el Ejemplo 2.6.1 se demostró que la única solución de la EDE

$$dZ_t = \mu dt + \sigma dW_t, \quad \mu \in \mathbb{R}, \quad \sigma > 0,$$

está dada por  $Z_t = \mu t + \sigma W_t$  (MB con Deriva).

Supongamos ahora que queremos simular una trayectoria de dicho proceso en el conjunto de tiempos  $0 = t_0 < t_1 < \dots < t_n = T$ . Es decir, queremos obtener  $Z_{t_0}, \dots, Z_{t_n}$ . Otra cosa que se mencionó al final de la Sección 2.6, es que podemos darnos cuenta que si tomamos la función  $g(t, x) = \mu t + \sigma x$ , entonces se cumple la igualdad  $Z_t = g(t, W_t)$ . De esta forma, es posible obtener  $Z_{t_0}, \dots, Z_{t_n}$  evaluando la función  $g$  en los puntos  $(t_0, W_{t_0}), \dots, (t_n, W_{t_n})$ . Y recordando que podemos obtener  $W_{t_0}, \dots, W_{t_n}$  mediante el Algoritmo 11, entonces el método para simular trayectorias (en los tiempos  $0 = t_0 < \dots < t_n = T$ ) provenientes de un MB con Deriva se muestra en el Algoritmo 12, la simulación puede hacerse usando el Código A.15 en el lenguaje de programación R.

---

**Algoritmo 12** Simulación de un MB con Deriva

---

1. Tomar el vector de tiempos  $(t_0, \dots, t_n)$  y multiplicar por  $\mu \in \mathbb{R}$  para obtener  $(\mu t_0, \dots, \mu t_n)$ .
  2. Simular un MB en los tiempos  $t_0, \dots, t_n$  mediante el Algoritmo 11 para obtener  $(W_{t_0}, \dots, W_{t_n})$ .
  3. Multiplicar el vector  $(W_{t_0}, \dots, W_{t_n})$  por  $\sigma$  para obtener  $(\sigma W_{t_0}, \dots, \sigma W_{t_n})$ .
  4. Sumar los vectores obtenido en los Pasos 1 y 3 para obtener  $(Z_{t_0}, \dots, Z_{t_n})$ .
- 

Por otro lado, retomando el Ejemplo 2.6.2, se mostró que la única solución de la EDE

$$dS_t = \mu S_t dt + \sigma S_t dW_t, \quad \mu \in \mathbb{R}, \quad \sigma > 0, \quad S_0 > 0,$$

es  $S_t = S_0 \exp\left(\left(\mu - \frac{1}{2}\sigma^2\right)t + \sigma W_t\right)$ ; es decir, es la EDE correspondiente al MBG. De nuevo, buscamos simular trayectorias del proceso en el conjunto de tiempos  $0 = t_0 < \dots < t_n = T$ . Mediante una observación similar que en el caso del MB con Deriva, notamos que tomando  $g(t, x) = S_0 \exp\left(\left(\mu - \frac{1}{2}\sigma^2\right)t + \sigma x\right)$  podemos verificar que  $S_t = g(t, W_t)$ . Más aún, uno puede darse cuenta que

$$\ln\left(\frac{S_t}{S_0}\right) = \left(\mu - \frac{1}{2}\sigma^2\right)t + \sigma W_t, \quad t > 0;$$

entonces  $\left(\ln\left(\frac{S_t}{S_0}\right)\right)_{t \geq 0}$  es un MB con Deriva. Por lo que podemos usar el Algoritmo 12 para simular un MBG. El Algoritmo 13 detalla el procedimiento para simular trayectorias provenientes de un MBG en el conjunto de tiempos  $0 = t_0 < \dots < t_n = T$ . El Código A.16 ayuda a simularlo haciendo uso del lenguaje de programación R.

---

**Algoritmo 13** Simulación de un Movimiento Browniano Geométrico

---

1. Usar el Algoritmo 12 para simular un MB con Deriva de coeficientes  $\mu - \frac{1}{2}\sigma^2$  y  $\sigma$  para obtener  $(Z_{t_0}, \dots, Z_{t_n})$ .
  2. Aplicar la función  $\exp$  a cada coordenada del vector  $(Z_{t_0}, \dots, Z_{t_n})$  para convertirlo en  $(\exp(Z_{t_0}), \dots, \exp(Z_{t_n}))$ .
  3. Finalmente, multiplicar cada coordenada del vector por  $S_0$  y obtener  $(S_{t_0}, \dots, S_{t_n})$ .
- 

En conclusión, cuando podemos expresar la solución a una EDE a través de una función  $g(t, W_t)$ , la simulación de dichos procesos solamente reside en simular un MB y aplicar la función correspondiente a la trayectoria del MB.

### 3.3.3. Simulación de EDE Vía su Discretización

Cabe destacar que, los Ejemplos 2.6.1 y 2.6.2 tienen solución analítica, por lo que ambos pueden ser simulados de forma directa como se explicó en la sección 3.3.2. Sin embargo, aún cuando podamos garantizar la existencia de un proceso solución a una EDE, no siempre podremos encontrar la expresión analítica de dicha solución. Por esto, es necesario que podamos simular trayectorias del proceso solución a una EDE sin necesidad de encontrar dicha solución. Se puede consultar más acerca de los métodos que se presentarán a continuación en [15].

#### 3.3.3.1. Esquema de Euler

El método más conocido para simular trayectorias de una EDE es el esquema de Euler, éste se basa en la discretización de la EDE. Supongamos que tenemos la EDE

$$dZ_t = \alpha(t, Z_t)dt + \beta(t, Z_t)dW_t, \quad Z_0 = z_0,$$

y queremos simular una trayectoria del proceso solución en el conjunto de tiempos  $0 = t_0 < t_1 < \dots < t_n = T$ . Entonces cambiamos  $dZ_t$  por  $\Delta Z_{t_i} = Z_{t_i} - Z_{t_{i-1}}$ ,  $dW_t$  por  $\Delta W_{t_i} = W_{t_i} - W_{t_{i-1}}$  y  $dt$  por  $\Delta t_i = t_i - t_{i-1}$ . Así, la EDE discretizada tiene la forma

$$\Delta Z_{t_i} = \alpha(t_{i-1}, Z_{t_{i-1}})\Delta t_i + \beta(t_{i-1}, Z_{t_{i-1}})\Delta W_{t_i}, \quad Z_0 = z_0,$$

donde  $\Delta W_{t_i} \sim Normal(0, \Delta t_i)$  debido a que es un incremento Browniano. Por lo que, podemos idear un método recursivo para la simulación de la trayectoria, dicho método se describe en el Algoritmo 14.

---

**Algoritmo 14** Esquema de Euler para simular una EDE

---

1. Iniciamos con  $Z_0 = z_0$ ,  $i = 1$ .
  2. Simulamos  $\Delta W_{t_i} \sim Normal(0, \Delta t_i)$ .
  3. Calculamos  $Z_{t_i} = Z_{t_{i-1}} + \alpha(t_{i-1}, Z_{t_{i-1}})\Delta t_i + \beta(t_{i-1}, Z_{t_{i-1}})\Delta W_{t_i}$ .
  4. Hacemos  $i = i + 1$ .
  5. Si  $i > n$  vamos al paso 5. En caso contrario regresamos al paso 2.
  6. Regresamos el vector  $(Z_{t_0}, \dots, Z_{t_n})$ .
- 

### 3.3.3.2. Esquema de Milstein

Otro método que también nos proporciona una forma de simular una EDE es el esquema de Milstein el cual es similar al de Euler en el sentido de que también discretiza la EDE pero además usa el Lema de Itô para mejorar la precisión de la aproximación incluyendo un término adicional, de modo que la expresión de la EDE discretizada toma la forma

$$\begin{aligned} \Delta Z_{t_i} &= \alpha(t_{i-1}, Z_{t_{i-1}})\Delta t_i + \beta(t_{i-1}, Z_{t_{i-1}})\Delta W_{t_i} \\ &\quad + \frac{1}{2}\beta(t_{i-1}, Z_{t_{i-1}})\frac{\partial \beta}{\partial x}(t_{i-1}, Z_{t_{i-1}})\left((\Delta W_{t_i})^2 - \Delta t_i\right), \quad Z_0 = z_0, \end{aligned}$$

donde nuevamente  $\Delta W_{t_i} \sim Normal(0, \Delta t_i)$ . Análogamente, el Algoritmo 15 describe el proceso para simular la EDE con el esquema de Milstein.

Cuando tratamos con este tipo de aproximaciones, es importante no perder de vista que mientras más chico sea  $\Delta t_i$ , la aproximación será más exacta. Una forma de estimar qué tan precisas son distintas aproximaciones es mediante la definición del orden de convergencia a la solución analítica.

**Definición 3.3.1** (Orden Fuerte de Convergencia). *Supongamos que  $(Z_t^\delta)_{0 \leq t \leq T}$  es una discretización (con  $\delta = \max\{t_i - t_{i-1}\}$ ) del proceso a tiempo continuo  $(Z_t)_{0 \leq t \leq T}$ . Se dice que el orden fuerte de convergencia de  $Z_t^\delta$  a  $Z_t$  es  $\gamma$  si  $\forall T > 0$  (horizonte de tiempo) fijo. Se cumple que*

$$\mathbb{E}|Z_T^\delta - Z_T| \leq C\delta^\gamma, \quad \forall \delta < \delta_0,$$

---

**Algoritmo 15** Esquema de Milstein para simular una EDE
 

---

1. Iniciamos con  $Z_0 = z_0$ ,  $i = 1$ .
  2. Simulamos  $\Delta W_{t_i} \sim Normal(0, \Delta t_i)$ .
  3. Calculamos  $Z_{t_i} = Z_{t_{i-1}} + \alpha(t_{i-1}, Z_{t_{i-1}})\Delta t_i + \beta(t_{i-1}, Z_{t_{i-1}})\Delta W_{t_i} + \frac{1}{2}\beta(t_{i-1}, Z_{t_{i-1}})\frac{\partial\beta}{\partial x}(t_{i-1}, Z_{t_{i-1}})\left((\Delta W_{t_i})^2 - \Delta t_i\right)$ .
  4. Hacemos  $i = i + 1$ .
  5. Si  $i > n$  vamos al paso 5. En caso contrario regresamos al paso 2.
  6. Regresamos el vector  $(Z_{t_0}, \dots, Z_{t_n})$ .
- 

con  $\delta_0 > 0$  adecuada y  $C$  una constante positiva que no depende de  $\delta$ .

Ya que las discretizaciones comúnmente utilizan  $\delta$  en  $(0, 1)$  suficientemente pequeño, podemos ver que la diferencia esperada entre las aproximaciones y la solución analítica disminuye a medida que  $\gamma$  aumenta. Es decir, en cuanto a la precisión de las aproximaciones preferimos aquellas con órdenes de convergencia mayores. En nuestro caso, la diferencia entre los esquemas de Euler y Milstein es que el orden de convergencia del esquema de Euler es  $\gamma = \frac{1}{2}$  y el de Milstein es  $\gamma = 1$ . Por lo tanto en esta tesis, si se necesita simular una EDE, se usará Milstein.

### 3.3.3.3. Consistencia en la Discretización

Un punto muy importante que se debe mostrar, es que tanto el Esquema de Euler como el de Milstein, efectivamente preservan las propiedades probabilísticas que tienen las soluciones analíticas. Pues, al tratarse de métodos de aproximación, se busca que los resultados que proporcionan (en este caso las trayectorias) sean consistentes con la teoría. Una manera de verificar que las discretizaciones cumplen con lo anteriormente establecido, es mediante la comparación de las distribuciones finito-dimensionales.

Tomando el Ejemplo 2.6.2 (pues será de gran importancia en esta tesis) la solución analítica  $S_t = S_0 \exp\left((\mu - \frac{1}{2}\sigma^2)t + \sigma W_t\right)$ , satisface

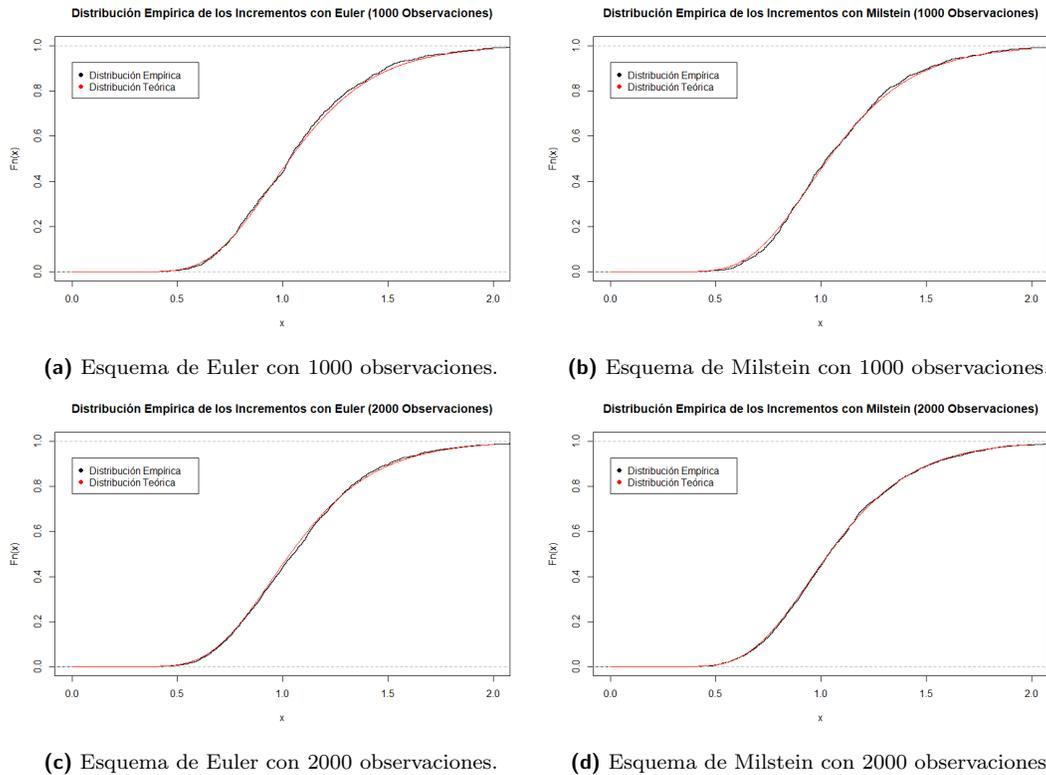
$$\frac{S_1}{S_0} \sim LogNormal\left(\mu - \frac{1}{2}\sigma^2, \sigma^2\right),$$

por lo tanto, dicha ley de probabilidad debería mantenerse al discretizar el intervalo  $[0, 1]$ , simular el MBG mediante los esquemas de discretización mencionados

anteriormente y comparar su valor final. Es decir, supongamos que se discretiza el intervalo  $[0, 1]$  en subintervalos de longitud  $\Delta t$  suficientemente pequeño, se simula un MBG  $(S_0^E, \dots, S_1^E)$  con el esquema de Euler mediante el Algoritmo 14 y análogamente  $(S_0^M, \dots, S_1^M)$  con Milstein usando el Algoritmo 15. Entonces debe cumplirse que

$$\frac{S_1^E}{S_0^E} \sim \text{LogNormal} \left( \mu - \frac{1}{2}\sigma^2, \sigma^2 \right) \quad \text{y} \quad \frac{S_1^M}{S_0^M} \sim \text{LogNormal} \left( \mu - \frac{1}{2}\sigma^2, \sigma^2 \right).$$

Para verificar lo anterior, se simularon muestras mediante los Algoritmos 14 y 15 tomando  $\mu = 0.08$ ,  $\sigma = 0.3$  y  $S_0 = 1$ . Las comparaciones con la distribución teórica se muestran en la Figura 3.10.



**Figura 3.10:** Comparación de las distribuciones finito dimensionales del MBG (rojo) con las obtenida mediante los métodos de Euler y Milstein (negro) con  $\mu = 0.08$ ,  $\sigma = 0.3$ ,  $S_0 = 1$ ,  $\Delta t = \frac{1}{252}$  y horizonte de tiempo  $T = 1$ .

### 3.4. Estimación de Parámetros para EDE

Ahora que se ha explicado cómo simular trayectorias de un proceso solución de una EDE cuando se tienen los parámetros de ésta, es momento de considerar el problema inverso. Es decir, supongamos que contamos con una trayectoria proveniente

de un proceso solución a una EDE y deseamos hacer inferencia sobre sus parámetros.

De nuevo, tomaremos el MBG

$$dS_t = \mu S_t dt + \sigma S_t dW_t, \quad \mu \in \mathbb{R}, \quad \sigma > 0, \quad (3.17)$$

para el desarrollo del método de estimación de parámetros, pues tiene la ventaja de que podemos aprovechar el hecho de conocer la solución analítica. Supongamos que contamos con una trayectoria de un MBG en los tiempos  $0 = t_0 < \dots < t_n = T$ , donde  $|t_i - t_{i-1}| = \frac{T}{n} = \Delta t$  para  $i = 1, \dots, n$ . Es decir, se supondrá que el espaciamiento entre los tiempos de observación es constante.

De nuevo, es importante recordar que si  $(S_t)_{t \geq 0}$  es un MBG, entonces  $\left(\ln\left(\frac{S_t}{S_0}\right)\right)_{t \geq 0}$  es un MB con Deriva. Entonces, usando lo anterior, tenemos que las observaciones

$$\ln\left(\frac{S_{t_i}}{S_{t_0}}\right) = \left(\mu - \frac{1}{2}\sigma^2\right)t_i + \sigma W_{t_i} \quad i \in \{1, \dots, n\},$$

proviene de un MB con Deriva. Además, por la propiedad de independencia en los incrementos y la distribución Normal del MB en la Definición 2.4.1, tenemos que

$$\ln\left(\frac{S_{t_i}}{S_{t_0}}\right) - \ln\left(\frac{S_{t_{i-1}}}{S_{t_0}}\right) = \ln\left(\frac{S_{t_i}}{S_{t_{i-1}}}\right) = \left(\mu - \frac{1}{2}\sigma^2\right)(t_i - t_{i-1}) + \sigma(W_{t_i} - W_{t_{i-1}})$$

entonces

$$\ln\left(\frac{S_{t_i}}{S_{t_{i-1}}}\right) = \left(\mu - \frac{1}{2}\sigma^2\right)\Delta t + \sigma(W_{t_i} - W_{t_{i-1}})$$

son v.a.i.i.d con distribución *Normal*  $\left(\left(\mu - \frac{1}{2}\sigma^2\right)\Delta t, \sigma^2\Delta t\right)$  para  $i \in \{1, \dots, n\}$ . Con el fin de facilitar la notación llamaremos  $x_i := \ln\left(\frac{S_{t_i}}{S_{t_{i-1}}}\right)$  y  $\delta := \left(\mu - \frac{1}{2}\sigma^2\right)$ .

Entonces con la nueva notación, tenemos que la muestra aleatoria  $x_1, \dots, x_n$  proviene de una distribución *Normal*  $(\delta\Delta t, \sigma^2\Delta t)$ . Por lo que, para determinar los EMV de  $\delta\Delta t$  y  $\sigma^2\Delta t$ , la verosimilitud de la muestra es

$$L(\delta\Delta t, \sigma^2\Delta t) = \prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma^2\Delta t}} \exp\left(-\frac{(x_i - \delta\Delta t)^2}{2\sigma^2\Delta t}\right), \quad (3.18)$$

así, los estimadores que maximizan la verosimilitud son

$$\hat{\delta}\Delta t = \frac{1}{n} \sum_{i=1}^n x_i \quad y \quad \hat{\sigma}^2\Delta t = \frac{1}{n} \sum_{i=1}^n (x_i - \hat{\delta}\Delta t)^2.$$

Entonces

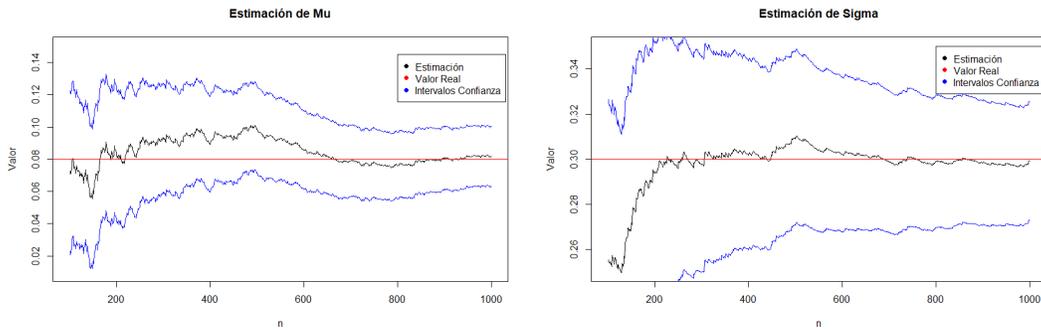
$$\hat{\delta} = \frac{1}{n\Delta t} \sum_{i=1}^n x_i \quad y \quad \hat{\sigma}^2 = \frac{1}{n\Delta t} \sum_{i=1}^n (x_i - \hat{\delta})^2,$$

por lo tanto

$$\hat{\mu} = \hat{\delta} + \frac{1}{2}\hat{\sigma}^2 \quad y \quad \hat{\sigma} = \sqrt{\frac{1}{n\Delta t} \sum_{i=1}^n (x_i - \hat{\delta})^2}, \quad \text{donde } \hat{\delta} = \frac{1}{n\Delta t} \sum_{i=1}^n x_i, \quad (3.19)$$

son los EMV de los coeficientes  $\mu$  y  $\sigma$  del MBG en la ecuación (3.17).

Es importante mencionar que la precisión de los estimadores proporcionados por la ecuación (3.19), está sujeta al número de observaciones  $n$  que se tengan de los incrementos del MBG, mientras más observaciones se tengan, más precisos serán los estimadores; en la Figura 3.11 se muestra lo mencionado anteriormente para un ejemplo específico.



(a) Estimación de  $\mu$  en función del número de observaciones  $n$ . (b) Estimación de  $\sigma$  en función del número de observaciones  $n$ .

**Figura 3.11:** Comparación de las estimaciones de  $\mu$  y  $\sigma$  para trayectorias del MBG usando Milstein con  $\mu = 0.08$ ,  $\sigma = 0.3$ ,  $S_0 = 1$ ,  $\Delta t = 1$ , horizonte de tiempo  $T = n$  e intervalos de confianza al 95%.

Cuando no es posible hallar una solución analítica, es necesario recurrir a otro tipo de métodos, tales como el Método de Euler o de Elerian, que aprovechan la discretización mediante el Esquema de Euler o Milstein respectivamente, de tal forma que pueda obtenerse una función de verosimilitud de la discretización de la EDE. Por otro lado el método de Kessler que en vez de centrarse en las trayectorias discretizadas del proceso, como los dos anteriores, se enfoca en buscar una aproximación directa de la función de verosimilitud mediante una expansión de Taylor de orden mayor a 2, entre otros. Dichos métodos no se incluyen en esta tesis, sin embargo puede consultarse más acerca de éstos en [15].



# Capítulo 4

## Modelo de Difusión Markov-Modulado con Saltos

En los mercados financieros, los precios de las acciones cambian constantemente. Dicho fenómeno se ha estudiado y se han adoptado distintos modelos que puedan asemejar la dinámica del movimiento de estos precios. El modelo más conocido en este ámbito es el Movimiento Browniano Geométrico (MBG); como se explicó anteriormente, el atractivo de este modelo es que nos proporciona dinámicas proporcionales al nivel del precio actual de la acción e intuitivamente es una propiedad que se podría esperar de los precios reales. En la sección 2.6 se introdujeron las Difusiones de Itô; en esta sección explicaremos cómo se usan las difusiones en las Finanzas Cuantitativas para asemejar la dinámica de movimiento de las acciones. Primero, se precisarán algunos detalles técnicos con respecto a la escala del tiempo. Luego, se enunciarán algunas desventajas de suponer dinámicas de dicho tipo y se añadirá un término adicional a las difusiones para incluir la modelación de eventos fortuitos. Esto da lugar al modelo de difusión con saltos más conocido en las Finanzas Cuantitativas. Finalmente, presentaremos una propuesta nueva para modelar los movimientos de los precios de las acciones al que se le llamará Modelo de Difusión Markov-Modulado con Saltos, además daremos métodos de estimación de los parámetros junto con un ejemplo para analizar los resultados de dichas estimaciones.

### 4.1. Difusiones de Itô en las Finanzas Cuantitativas

Una Difusión de Itô es un proceso estocástico  $(S_t)_{t \geq 0}$  que es solución a una EDE de la forma

$$dS_t = \alpha(t, S_t)dt + \beta(t, S_t)dW_t. \quad (4.1)$$

Como mencionamos anteriormente, una de las difusiones más conocidas en las Finanzas Cuantitativas es el MBG, dado por

$$dS_t = \mu S_t dt + \sigma S_t dW_t, \quad \mu \in \mathbb{R}, \quad \sigma > 0.$$

Los coeficientes  $\mu$  y  $\sigma$  del MBG están estrechamente relacionados con la media y la desviación estándar de los log-rendimientos del proceso, pues como se vio en el Ejemplo 2.6.2, tenemos que si  $(S_t)_{t \geq 0}$  es un MBG (que representa el precio de una acción) con coeficientes  $\mu$  y  $\sigma$ , entonces

$$S_t = S_0 \exp \left( \left( \mu - \frac{1}{2} \sigma^2 \right) t + \sigma W_t \right);$$

y por lo tanto

$$\ln \left( \frac{S_t}{S_0} \right) = \left( \mu - \frac{1}{2} \sigma^2 \right) t + \sigma W_t,$$

es un MB con Deriva. De esta forma, por las Propiedades 3 y 4 en la Definición 2.4.1, si tenemos el histórico de precios de la acción, es decir las observaciones del proceso, en los tiempos  $\{t_0, \dots, t_n\}$  (con  $t_{i-1} < t_i$  y  $\Delta t = t_i - t_{i-1}$  es el espaciamiento entre las observaciones), podemos estimar  $\mu$  y  $\sigma$  definiendo los log-rendimientos (en ocasiones nos referiremos a éstos simplemente como **rendimientos**) de los precios de la siguiente forma

$$R_1 = \ln \left( \frac{S_{t_1}}{S_{t_0}} \right), \dots, R_n = \ln \left( \frac{S_{t_n}}{S_{t_{n-1}}} \right).$$

Así, tenemos que  $R_i \sim N \left( \left( \mu - \frac{1}{2} \sigma^2 \right) \Delta t, \sigma^2 \Delta t \right)$  y además por hipótesis, pues son incrementos provenientes de un MBG, son variables aleatorias independientes. Por lo tanto

$$\hat{\mu} = \frac{1}{2} \hat{\sigma}^2 + \frac{1}{\Delta t} \bar{R}, \quad \hat{\sigma} = \sqrt{\frac{1}{n \Delta t} \sum_{i=1}^n (R_i - \bar{R})^2}; \quad \text{con } \bar{R} = \frac{1}{n} \sum_{i=1}^n R_i,$$

son los EMV de  $\mu$  y  $\sigma$ . En Finanzas Cuantitativas, al coeficiente  $\sigma$  se le conoce como **Volatilidad** y así le llamaremos de ahora en adelante.

Usualmente, para simplificar los cálculos, se toma  $\Delta t=1$  que corresponde a observaciones diarias. Es decir, cada unidad de tiempo representa un día y podemos escalar dichos parámetros si queremos cambiar la escala de tiempo. Por ejemplo, si tenemos que  $\mu_D$  y  $\sigma_D$  son la media y volatilidad diaria, es posible obtener la media y volatilidad anuales  $(\mu_A, \sigma_A)$ . Esto se hace recordando que, si  $\sigma_D$  es la volatilidad diaria y nuestra unidad de tiempo representa un día, entonces

$$\text{Var} \left( \ln \left( \frac{S_i}{S_{i-1}} \right) \right) = \sigma_D^2.$$

Además, el mercado solamente opera los días hábiles hay 252 días en un año financiero; de modo que nos interesa conocer

$$\sigma_A^2 = Var \left( \ln \left( \frac{S_{252}}{S_0} \right) \right) = Var \left( \sum_{i=1}^{252} \ln \left( \frac{S_i}{S_{i-1}} \right) \right),$$

usando la independencia de los rendimientos tenemos que

$$\sigma_A^2 = \sum_{i=1}^{252} Var \left( \ln \left( \frac{S_i}{S_{i-1}} \right) \right) = 252\sigma_D^2.$$

Por lo tanto, aplicando lo mismo a la media, obtenemos

$$\mu_A = 252\mu_D, \quad \sigma_A = \sqrt{252}\sigma_D, \quad (4.2)$$

con los cuales podemos modelar el comportamiento del precio de la acción, donde cada unidad de tiempo representa un año.

La idea de escalar los parámetros a 252 días en vez de a 365 es debido a que las acciones no cotizan en fines de semana. Además, si los fines de semana ocurren eventos que pudiesen impactar el precio de las acciones, dicho impacto toma lugar el día hábil siguiente posterior a dichas eventualidades.

#### 4.1.1. Consideraciones acerca del MBG

Tomando en cuenta lo anterior, cuando suponemos que el proceso de precios de una acción es un MBG  $(S_t)_{t \geq 0}$  con coeficientes  $\mu$  y  $\sigma$ , estamos suponiendo, que los log-rendimientos (ya sean diarios, mensuales, anuales, etc.) de la acción se distribuyen *Normal* con parámetros proporcionales a  $\mu$  y  $\sigma$  (dependiendo de la escala de tiempo).

Sin embargo, asumir una dinámica del tipo MBG es controversial debido a que bajo este supuesto, las colas de los log-rendimientos, por ser de una Normal, deberían ser ligeras y en la realidad se observan colas mucho más pesadas que pocas veces se asemejan a las de una Normal, porque dicha variable aleatoria concentra la mayor parte de su densidad alrededor de su media por lo que los valores alejados de ésta tienen una probabilidad insignificante, mientras que en la realidad dichos valores alejados de la media se observan con frecuencia.

Por otro lado, en la práctica, los precios de las acciones presentan cambios abruptos ocasionales, por lo tanto los rendimientos asociados a estos cambios no corresponden a una distribución Normal, debido a que con frecuencia los incrementos exceden los intervalos de confianza de ésta. Dicho problema no significa que haya un error en el ajuste de los parámetros, sino que estos cambios abruptos obedecen a eventos fortuitos que ocurren a diario como por ejemplo: noticias, la subida en las tasas, emisión

de bonos por parte de la empresa, etc.

Es por esto que en ocasiones se añade un término adicional a la EDE (4.1) con lo que obtenemos la forma

$$dS_t = \alpha(t, S_t)dt + \beta(t, S_t)dW_t + d \sum_{i=1}^{N_t} \Delta P_i,$$

donde  $(N_t)_{t \geq 0}$  es un Proceso Poisson de tasa  $\lambda > 0$ . Así, cada ocurrencia del Proceso Poisson representa los cambios abruptos en el precio de las acciones y  $\Delta P_i$  es el tamaño del salto. Este tipo de Difusiones cuenta con su propia versión del Lema de Itô, el cual es útil para la resolución de este tipo de EDE. Éste se enuncia a continuación, su demostración puede consultarse en [10].

**Lema 4.1.1.** *Sea  $(S_t)_{t \geq 0}$  una Difusión de Itô con Saltos, definida como la suma de una Difusión de Itô y un Proceso Poisson Compuesto de tasa  $\lambda > 0$  con*

$$dS_t = \alpha(t, S_t)dt + \beta(t, S_t)dW_t + d \sum_{i=1}^{N_t} \Delta P_i, \quad (4.3)$$

donde los coeficientes  $\alpha$  y  $\beta$  satisfacen las propiedades del Teorema 2.6.1 y  $\Delta P_i$  es el tamaño del salto definido como  $\Delta P_i = S_{\tau_i} - S_{\tau_i^-}$  donde  $\{\tau_1, \tau_2, \dots\} \subset \mathbb{R}^+$  son los tiempos de ocurrencia del Proceso Poisson  $(N_t)_{t \geq 0}$ . Entonces, si  $f : [0, T] \times \mathbb{R} \rightarrow \mathbb{R}$  es una función de clase  $C^{1,2}$ , el proceso  $Z_t = f(t, S_t)$  tiene la representación

$$\begin{aligned} f(t, S_t) - f(0, S_0) &= \int_0^t \left( \frac{\partial f}{\partial t}(r, S_r) + \alpha(r, S_r) \frac{\partial f}{\partial x}(r, S_r) + \frac{1}{2} \beta(r, S_r)^2 \frac{\partial^2 f}{\partial x^2}(r, S_r) \right) dr \\ &+ \int_0^t \beta(r, S_r) \frac{\partial f}{\partial x}(r, S_r) dW_r + \sum_{\tau_i \leq t} \left( f(S_{\tau_i^-} + \Delta P_i) - f(S_{\tau_i^-}) \right). \end{aligned} \quad (4.4)$$

Algo muy importante que debemos entender de (4.4) es que, como ahora añadimos un componente de saltos a la difusión, ésta se vuelve discontinua en los tiempos de ocurrencia  $\{\tau_1, \tau_2, \dots\}$  del Proceso Poisson, por lo tanto tiene sentido hablar de los límites por la derecha e izquierda  $(S_{\tau_i}, S_{\tau_i^-})$  de la difusión con saltos.

## 4.2. El Modelo de Merton

Antes de presentar el modelo que proponemos en esta tesis, se presentará el Modelo de Merton, como un ejemplo introductorio a las difusiones con saltos.

Tomemos una dinámica proporcional al nivel, es decir un MBG

$$dS_t = \mu S_t dt + \sigma S_t dW_t, \quad \mu \in \mathbb{R}, \sigma > 0.$$

La idea del modelo de Merton es modificar el MBG agregando un componente de saltos como en (4.3) pero sin perder la propiedad de incrementos proporcionales, es decir una dinámica del tipo

$$dS_t = \mu S_t dt + \sigma S_t dW_t + d \sum_{k=1}^{N_t} \Delta P_k, \quad (4.5)$$

donde el tamaño del salto puede descomponerse en la forma  $\Delta P_k = (e^{J_k} - 1)S_{\tau_k^-}$  donde  $(J_k)_{k \in \mathbb{N}}$  es una sucesión de variables aleatorias independientes e idénticamente distribuidas,  $(N_t)_{t \geq 0}$  es un Proceso Poisson de tasa  $\lambda > 0$ ,  $\{\tau_1, \tau_2, \dots\}$  son sus tiempos de ocurrencia y

$$\sum_{k=1}^{N_t} \Delta P_k = 0, \quad \text{si } N_t = 0, t > 0$$

o sea, la suma se interpreta como cero mientras el Proceso Poisson no tenga ninguna ocurrencia.

Aplicando el Lema 4.1.1 a (4.5) con  $f(x) = \ln(x)$ , tenemos que  $\ln(S_t)$  puede expresarse de la siguiente manera

$$\begin{aligned} \ln(S_t) - \ln(S_0) &= \int_0^t \left( \mu S_r \frac{1}{S_r} - \frac{1}{2} \sigma^2 S_r^2 \frac{1}{S_r^2} \right) dr + \int_0^t \sigma S_r \frac{1}{S_r} dW_r \\ &+ \sum_{k \geq 1, \tau_k \leq t} \left( \ln(S_{\tau_k^-} + (e^{J_k} - 1)S_{\tau_k^-}) - \ln(S_{\tau_k^-}) \right). \end{aligned}$$

Lo que se reduce a

$$\ln \left( \frac{S_t}{S_0} \right) = \left( \mu - \frac{1}{2} \sigma^2 \right) t + \sigma W_t + \sum_{k \geq 1, \tau_k \leq t} J_k,$$

Por lo tanto, obtenemos la expresión

$$S_t = S_0 \exp \left( \left( \mu - \frac{1}{2} \sigma^2 \right) t + \sigma W_t + \sum_{k \geq 1, \tau_k \leq t} J_k \right).$$

Y haciendo un cambio de notación en la suma tenemos

$$S_t = S_0 \exp \left( \left( \mu - \frac{1}{2} \sigma^2 \right) t + \sigma W_t + \sum_{k=1}^{N_t} J_k \right); \quad (4.6)$$

donde

$$\sum_{k=1}^{N_t} J_k = 0, \quad \text{si } N_t = 0, \quad t > 0.$$

Se supone además, que  $(J_k)_{k \in \mathbb{N}}$ ,  $(N_t)_{t \geq 0}$ ,  $(W_t)_{t \geq 0}$  son independientes.

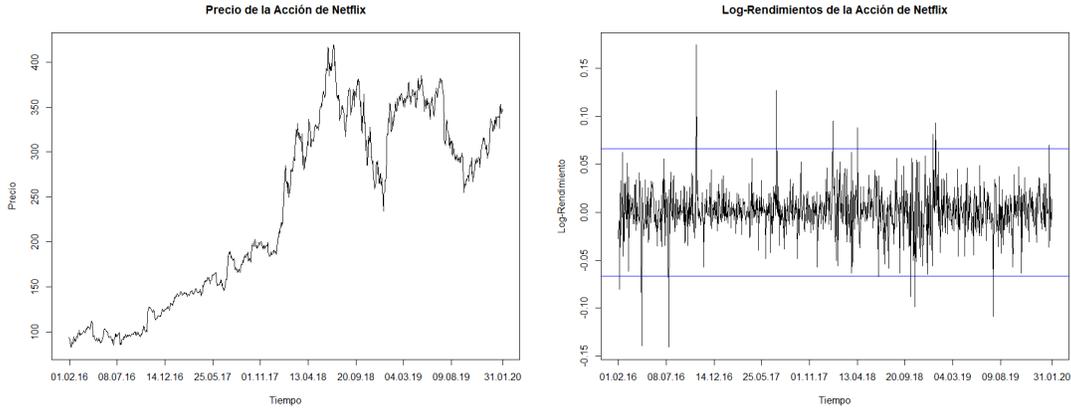
Dependiendo de la distribución de  $J_k$  en (4.6), se tienen distintos modelos. Por ejemplo, cuando  $J_k \sim N(\eta, \delta^2)$  tenemos el Modelo de Merton, que es el modelo de difusión con saltos más conocido en las Finanzas Matemáticas. La elección de la distribución Normal se debe a que los cálculos con esta distribución son relativamente sencillos, por lo que facilita la valuación de instrumentos financieros simples. Por otro lado, si  $J_k \sim \text{Laplace}(0, \eta)$  tenemos el modelo de Kou (se puede consultar más acerca de este modelo en [14]), cuya ventaja no solamente consiste en la facilidad de los cálculos con esta distribución sino también permite encontrar expresiones cerradas para valuar algunos instrumentos financieros exóticos como se muestra en [14]. Observemos que, como  $(N_t)_{t \geq 0}$  es un Proceso Poisson de tasa  $\lambda > 0$ , cada cierto tiempo hay una ocurrencia de dicho proceso, lo que provoca que tengamos una dinámica del tipo MBG pero, en cada salto del proceso Poisson, el MBG es alterado a razón de una variable aleatoria  $\exp(J_k)$  externa.

### 4.2.1. Desventajas

La adición de saltos al modelo de difusión busca replicar los cambios abruptos que tienen los precios de las acciones de forma ocasional sin necesidad de alterar los parámetros del MBG.

Cuando pensamos en las justificaciones de incorporar los saltos al MBG, tenemos que es algo intuitivo, sin embargo cuando se observa una serie de precios, podemos notar que no solamente hay presencia de saltos, sino que también pareciera que después de la aparición de un salto, hay un cambio en la tendencia y volatilidad de los precios como puede apreciarse en la Figura 4.1.

El cambio de comportamiento en las difusiones se ha abordado en varias ocasiones en el contexto de seguros, por ejemplo en [13] se supone que las reservas de una compañía aseguradora se comportan como una difusión cuyos parámetros cambian de acuerdo a un PSM al mismo tiempo que dicho proceso regula la ocurrencia y el tamaño de las reclamaciones a la compañía. Otro trabajo similar al anterior pero en el contexto financiero es [12] donde se utiliza una difusión con saltos para modelar el precio del oro, donde la tasa del Proceso Poisson Compuesto cambia de acuerdo a un PSM, sin embargo la difusión no cambia de parámetros con el tiempo.



(a) Precio de la Acción de Netflix de Feb de 2016 a Feb de 2020 (b) Log Rendimientos de la Acción de Netflix de Feb de 2016 a Feb de 2020

**Figura 4.1:** Serie de precios de la acción de Netflix donde pueden observarse saltos, seguidos de cambios en la tendencia y volatilidad.

### 4.3. Modelo de Difusión Markov-Modulado con Saltos

En esta tesis se propone una modificación del Modelo de Kou, en el cual los parámetros de la difusión cambian cada que se efectúa un salto de un Proceso de Saltos de Markov (PSM), la motivación detrás de éste es capturar los cambios en la dinámica de movimiento del precio de la acción después de que se efectúa un salto.

Tomemos un PSM estándar  $(X_t)_{t \geq 0}$  con espacio de estados finito  $\{1, 2, \dots, m\}$ , vamos a asumir que dichos estados representan los distintos ambientes de los mercados financieros: tanto favorables como desfavorables, para el precio de una acción  $(S_t)_{t \geq 0}$ . Denotaremos por  $\{\tau_1, \tau_2, \dots\}$  a los tiempos de salto del PSM. Por ejemplo, el estado 1 refleja economía favorable para la acción (buen crecimiento en el sector de la empresa o confianza en la acción por parte de los inversionistas) y ésta puede descender hasta el estado  $m$  que simboliza un ambiente desfavorable para la acción (crisis en el sector de la empresa). Por lo tanto, la situación actual afecta directamente sobre la media y volatilidad del precio de la acción: los estados  $j \in \{1, 2, \dots, m\}$  cercanos al estado 1 provocan que la acción tenga media  $\mu_j$  alta y volatilidad  $\sigma_j$  baja; los estados  $j \in \{1, 2, \dots, m\}$  cercanos al estado  $m$  ocasionan que la acción tenga una media  $\mu_j$  muy baja (posiblemente negativa) y gran volatilidad  $\sigma_j$ . Así, el modelo matemático queda de la siguiente forma.

Supongamos que  $(X_t)_{t \geq 0}$  es un PSM con espacio de estados  $E = \{1, 2, \dots, m\}$  y Generador Infinitesimal  $Q = (q_{ij})_{i,j \in E}$ ,  $(W_t)_{t \geq 0}$  un Movimiento Browniano independiente de  $(X_t)_{t \geq 0}$ , un proceso de conteo  $(N_t)_{t \geq 0}$  que registra el número de saltos del

PSM hasta el instante  $t$ ,  $(J_k)_{k \in \mathbb{N}}$  una sucesión de variables aleatorias independientes con distribución  $Laplace(0, \eta)$  (es decir,  $|J_k| \sim Exponencial(\eta)$  para  $k \in \mathbb{N}$ ),  $S_0$  una constante real positiva y dos conjuntos de números reales  $\{\mu_1, \dots, \mu_m\}$  y  $\{\sigma_1, \dots, \sigma_m\}$  con  $\sigma_j > 0$ ,  $j = 1, \dots, m$ . Entonces el Modelo de Difusión Markov-Modulado con Saltos se define como

$$dS_t = \mu_{X_t} S_t dt + \sigma_{X_t} S_t dW_t + d \sum_{k=1}^{N_t} \Delta P_k; \quad (4.7)$$

donde  $\Delta P_k = \left( \exp(J_k) - 1 \right) S_{\tau_k^-}$ ,  $\{\tau_1, \tau_2, \dots\}$  son los tiempos de salto del PSM y

$$\sum_{k=1}^{N_t} \Delta P_k = 0, \quad \text{si } N_t = 0, \quad t > 0.$$

Lo primero que podemos observar del modelo (4.7) es que, debido a la independencia de  $(W_t)_{t \geq 0}$  con  $(X_t)_{t \geq 0}$ , podemos resolver la EDE aplicando el Lema de Itô. Así, si existe  $h > 0$  tal que  $X_t = j$  para todo  $t \in [r, r + h)$ , entonces

$$S_t = S_r \exp \left[ \left( \mu_j - \frac{1}{2} \sigma_j^2 \right) (t - r) + \sigma_j (W_t - W_r) \right], \quad t \in [r, r + h). \quad (4.8)$$

Recordemos que  $N_t$  representa el número de saltos que tiene el PSM hasta el tiempo  $t$ . Entonces, podemos observar de (4.7) que cada que el PSM efectúa un salto, la difusión también tiene un salto y además, cambia de parámetros una vez que dicho salto ocurre. Así, usando (4.8) podemos afirmar que si  $X_t = i$  para  $t \in [r, u)$  y  $X_t = j$  para  $t \in [u, u + h)$  para algún  $h > 0$ , entonces

$$S_t = S_r \exp \left[ \left( \mu_i - \frac{1}{2} \sigma_i^2 \right) (u - r) + \sigma_i (W_u - W_r) \right. \\ \left. + J_k + \left( \mu_j - \frac{1}{2} \sigma_j^2 \right) (t - u) + \sigma_j (W_t - W_u) \right], \quad t \in [u, u + h) \quad (4.9)$$

con  $J_k \sim Laplace(0, \eta)$ .

La razón por la cual decidimos modificar el Modelo de Kou en lugar del Modelo de Merton es la suficiencia de información para estimar los parámetros de los saltos, se hablará de esto más a fondo en la siguiente sección.

## 4.4. Estimación

Para poder hacer la estimación del modelo, debemos tener en mente el hecho de que la difusión cambia de parámetros cada que ocurre un salto. Entonces lo primero

que debemos hacer es identificar los saltos. En este contexto, como estamos suponiendo que los saltos del proceso serán cambios abruptos en el valor del proceso, podemos visualizar dichos cambios si nos fijamos en los log-rendimientos del proceso. Supongamos que  $\{S_0, S_1, \dots, S_M\}$  son  $M+1$  observaciones del proceso  $(S_t)_{t \geq 0}$  (observemos que, para mayor facilidad, tomamos  $\Delta t = 1$ , esto no importa, debido a que como vimos anteriormente, podemos escalar los parámetros al finalizar la estimación), entonces los log-rendimientos del proceso están dados por

$$R_1 = \log \left( \frac{S_1}{S_0} \right), \dots, R_M = \log \left( \frac{S_M}{S_{M-1}} \right). \quad (4.10)$$

Así, tenemos  $\{R_1, \dots, R_M\}$  que constituyen  $M$  observaciones de los log-rendimientos del proceso y bajo la hipótesis que en el intervalo  $[n-1, n]$  no ocurrieron saltos, de (4.8) tenemos

$$R_n = \left( \mu_j - \frac{1}{2} \sigma_j^2 \right) + \sigma_j (W_n - W_{n-1}) \quad (4.11)$$

De esta forma  $R_n \sim N(\mu_j - \frac{1}{2} \sigma_j^2, \sigma_j^2)$ .

Por otro lado, en caso de que haya ocurrido un salto dentro del intervalo  $[n-1, n]$  (por ejemplo en el instante  $n-1+h$  para  $0 < h < 1$ ) de (4.9) deducimos

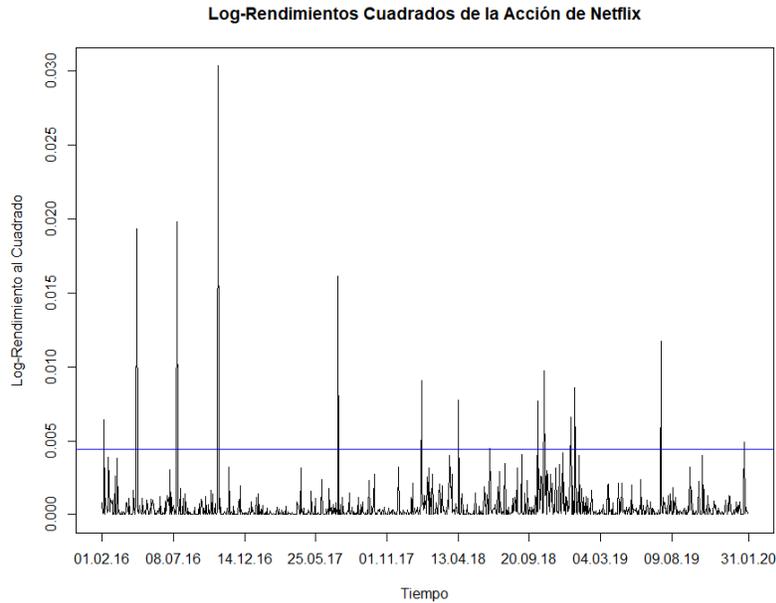
$$R_n = \left( \mu_i - \frac{1}{2} \sigma_i^2 \right) (1-h) + \sigma_i (W_{n-1+h} - W_{n-1}) + J_k + \left( \mu_j - \frac{1}{2} \sigma_j^2 \right) h + \sigma_j (W_n - W_{n-1+h}) \quad (4.12)$$

Estas dos últimas observaciones son muy importantes, debido a que podemos usarlas para decidir si ocurrió un salto en el intervalo  $[n, n+1]$ . Es decir, si  $R_{n^*}$  (p.a.  $n^* \in \{1, \dots, M\}$ ) sobrepasa un umbral (el cual fijaremos nosotros) podremos decir que ocurrió un salto en dicho intervalo; así podremos identificar los rendimientos que contengan un salto y estimar los parámetros de la difusión usando el resto de los  $R_n$ .

#### 4.4.1. Identificación de Saltos

Un problema muy delicado es decidir cómo se hará la identificación de los saltos, algunos textos como [11] implementan ciertos métodos para poder hacer la estimación de todos los parámetros, sin embargo esa metodología no nos sirve debido a que tenemos cambios de parámetros después de cada salto. Una alternativa es usar algoritmos de punto de cambio, los cuales ayudan a estimar los tiempos de cambio de comportamiento de un fenómeno cuando son desconocidos, se puede consultar más acerca de estos algoritmos en [19] y [20]. Otros artículos como [12] simplemente dan intervalos y asumen que cualquier rendimiento que salga de dichos intervalos constituye un salto, si bien no es un método sustentado teóricamente, este tiene la ventaja de obedecer a nuestra intuición. Esta observación sobre nuestra intuición será de gran importancia, pues debido a que contamos con un proceso que cambia de parámetros, podemos observar la gráfica de los log-rendimientos y escoger intervalos de confianza que se

adapten a nuestro modelo. Un ejemplo, con una serie de datos reales, se muestra en la Figura 4.2.



**Figura 4.2:** Log-Rendimientos al cuadrado de la acción de Netflix de Feb 2016 a Feb de 2020 y tolerancia a la altura 0.0044 fijada para captar el salto entre el 13 de abril de 2018 y el 20 de septiembre de 2018.

Para poder visualizar mejor los comportamientos de los rendimientos entre los que catalogaremos como saltos, la Figura 4.2 muestra los log-rendimientos al cuadrado del precio de la acción de Netflix entre los años 2016 y 2020. Podemos observar que si damos una tolerancia de 0.0044 a los rendimientos cuadrados y llamamos **saltos** a todos los rendimientos que excedan dicha cantidad, entonces se detectan algunos cambios en el comportamiento de los rendimientos inmediatamente después del salto. Por ejemplo entre el 1 de Noviembre de 2017 y el 13 de Abril de 2018 existe un salto y notamos que después de éste, los rendimientos presentan un comportamiento más errático que antes de que se haya efectuado el salto. Así, podemos fijar una tolerancia para los rendimientos, con la idea de captar los cambios de parámetros del proceso después de cada salto.

#### 4.4.2. Estimación de Coeficientes de la Difusión

Supongamos que identificamos  $M - N$  saltos (con  $N < M$ ) y éstos se encuentran en los rendimientos  $\{y_1, \dots, y_{M-N}\} \subset \{R_1, \dots, R_M\}$  entonces los rendimientos que no contienen un salto son elementos del conjunto  $\{R_1, \dots, R_M\} \setminus \{y_1, \dots, y_{M-N}\} =$

$\{x_1, \dots, x_N\}$ . Por lo tanto, hemos separado el conjunto de los rendimientos en aquellos que contienen un salto  $\{y_1, \dots, y_{M-N}\}$  y aquellos que no contienen ningún salto  $\{x_1, \dots, x_N\}$ . Por el momento nos quedamos con  $\{x_1, \dots, x_N\}$ ; como estamos considerando que los saltos están excluidos de este subconjunto, tenemos que (4.11) queda reducido a

$$x_n = (\mu_j - \frac{1}{2}\sigma_j^2) + \sigma_j(W_{n+1} - W_n), \quad n \in \{1, \dots, N\},$$

para algún  $j \in E$ . A partir de este momento llamaremos  $\delta_j = \mu_j - \frac{1}{2}\sigma_j^2$ .

El problema de la estimación de los parámetros  $\mu_j$  y  $\sigma_j$  es que, a pesar de que conocemos la distribución que siguen dichos rendimientos, no sabemos si la observación  $R_n$  pertenece; por ejemplo a la distribución Normal con parámetros  $(\delta_1, \sigma_1^2)$  o a la que tiene parámetros  $(\delta_m, \sigma_m^2)$ . Este problema es similar al de estimar los parámetros de una mezcla de distribuciones, donde se tienen observaciones  $x_1, \dots, x_N$  provenientes de una densidad de la forma

$$f(x, \theta) = \sum_{j=1}^m \pi_j f_j(x, \theta_j), \quad \text{con} \quad \sum_{j=1}^m \pi_j = 1 \quad \text{y} \quad \pi_j \geq 0, j \in \{1, \dots, m\} \quad (4.13)$$

donde las densidades  $f_j$  son conocidas, pero los parámetros  $\theta = (\theta_1, \dots, \theta_m), (\pi_1, \dots, \pi_m)$  son desconocidos. En general, cuando se tiene un conjunto de datos que proviene de una mezcla, el primer paso es determinar el número de densidades  $m$  de la que está compuesta, sin embargo este es un problema muy complicado al grado que los especialistas en este tema no han podido determinar un método exacto para la resolución de dicho problema. No obstante, éste no es nuestro caso, pues desde el inicio afirmamos que son  $m$  densidades debido a que el PSM tiene  $m$  estados.

Entonces, el problema de estimar los parámetros  $\mu_j$  y  $\sigma_j$ , para  $j = 1, \dots, m$  de la difusión es equivalente a resolver un problema de mezcla de distribuciones debido a que,  $\forall n \in \{1, \dots, N\}$  se tiene que  $x_n \sim N(\delta_j, \sigma_j^2)$  para algún  $j \in \{1, \dots, m\}$ , por lo tanto conocemos las densidades  $f_1, \dots, f_m$  y están dadas por

$$f_j(x; \theta_j) = \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp\left(-\frac{1}{2\sigma_j^2}(x - \delta_j)^2\right) \quad j \in \{1, \dots, m\}, \quad \text{con} \quad \theta_j = (\delta_j, \sigma_j^2).$$

Decidir a qué densidad  $f_j$  corresponde el rendimiento  $x_n$  es un problema difícil de abordar por sí solo. Lo que podemos hacer es considerar un conjunto de variables  $\{z_1, \dots, z_N\}$ . Estas variables auxiliares  $z_i \in \{1, \dots, m\}$  nos indicarán la densidad de la cual proviene la observación  $x_i$ , de esta forma si  $z_i = j$  esto quiere decir que la observación  $x_i$  proviene de la densidad  $f_j$ . Por lo tanto la densidad marginal de la observación  $x_i$  está dada por

$$f(x_i, \theta) = \sum_{j=1}^m f_j(x_i; \theta_j) \mathbb{P}(z_i = j) = \sum_{j=1}^m f_j(x_i; \theta_j) \pi_j, \quad (4.14)$$

entonces, usando (4.14) la log-verosimilitud tiene la forma

$$l(\theta|x_1, \dots, x_N) = \sum_{i=1}^N \log \left( \sum_{j=1}^m \pi_j f_j(x_i; \theta_j) \right). \quad (4.15)$$

Y para las parejas de observaciones  $(x_i, z_i)$  tenemos que

$$l(\theta|x_1, \dots, x_N, z_1, \dots, z_N) = \sum_{i=1}^N \log (\pi_{z_i} f_{z_i}(x_i; \theta_{z_i})). \quad (4.16)$$

Debido a que no se tienen las observaciones  $(z_1, \dots, z_N)$  tenemos un problema con datos faltantes que puede resolverse mediante el algoritmo EM, para esto necesitamos calcular las probabilidades  $\mathbb{P}(z_i = k|x_i)$ . Puede verificarse en [6] que dichas probabilidades, a las que llamaremos  $\gamma_{z_i}(k)$ , están dadas por

$$\gamma_{z_i}(k) = \mathbb{P}(z_i = k|x_i) = \frac{\pi_k f_k(x_i; \theta_k)}{\sum_{j=1}^m \pi_j f_j(x_i; \theta_j)}. \quad (4.17)$$

Además, el Paso E del algoritmo EM queda de la siguiente manera

$$S(\theta; \theta^{(t)}) = \sum_{j=1}^m \sum_{i=1}^N \gamma_{z_i}(k) \log \left( \pi_j f_j(x_i; \theta_j) \right), \quad (4.18)$$

y los estimadores que se obtienen en el Paso M del algoritmo son

$$\hat{\pi}_k = \frac{\sum_{i=1}^N \gamma_{z_i}(k)}{N}, \quad \hat{\delta}_k = \frac{\sum_{i=1}^N x_i \gamma_{z_i}(k)}{\sum_{i=1}^N \gamma_{z_i}(k)}, \quad \hat{\sigma}_k^2 = \frac{\sum_{i=1}^N (x_i - \delta_k)^2 \gamma_{z_i}(k)}{\sum_{i=1}^N \gamma_{z_i}(k)}. \quad (4.19)$$

Entonces, el procedimiento que debe seguirse para estimar  $\mu_1, \dots, \mu_m$  y  $\sigma_1, \dots, \sigma_m$  queda resumido en el Algoritmo 16, para implementarlo en el lenguaje de programación R puede usarse el Código A.20.

### 4.4.3. Ventajas del Modelo de Kou

Por otro lado, tenemos los rendimientos que contienen un salto  $\{y_i, \dots, y_{M_N}\}$ . La razón por cual se eligió partir del Modelo de Kou es que, los saltos en los precios de las acciones ocurren pocas veces al año, por lo que no tenemos suficiente información para confiar en las estimaciones de los parámetros de una variable aleatoria Normal. Esto lo justificamos con la información de Fisher definida como

$$\mathcal{I}(\theta) = \mathbb{E} \left[ \left( \frac{\partial}{\partial \theta} \ln f(X; \theta) \right)^2 \middle| \theta \right] = \int_{-\infty}^{\infty} \left( \frac{\partial}{\partial \theta} \ln f(x; \theta) \right)^2 f(x; \theta) dx.$$

---

**Algoritmo 16** Algoritmo EM para estimar  $\mu_j$  y  $\sigma_j$  de la difusión

---

1. Tomar parámetros iniciales  $\theta^{(0)} = (\pi_j^{(0)}, \delta_j^{(0)}, \sigma_j^{2(0)})_{j=1, \dots, m}$ ,  $t = 0$ ,  $\epsilon > 0$ .
2. Evaluar  $S(\theta; \theta^{(t)})$  usando (4.18).
3. **Paso E:** Calcular  $\gamma_{z_i}(k)^{(t+1)} = \frac{\pi_k^{(t)} f_k(x_i; \theta_k^{(t)})}{\sum_{j=1}^m \pi_j^{(t)} f_j(x_i; \theta_j^{(t)})}$ .
4. **Paso M:** Definir

$$\begin{aligned} \pi_k^{(t+1)} &= \frac{\sum_{i=1}^N \gamma_{z_i}(k)^{(t+1)}}{N}, & \delta_k^{(t+1)} &= \frac{\sum_{i=1}^N x_i \gamma_{z_i}(k)^{(t+1)}}{\sum_{i=1}^N \gamma_{z_i}(k)^{(t+1)}}, \\ \sigma_k^{2(t+1)} &= \frac{\sum_{i=1}^N (x_i - \delta_k^{(t+1)})^2 \gamma_{z_i}(k)^{(t+1)}}{\sum_{i=1}^N \gamma_{z_i}(k)^{(t+1)}} \end{aligned}$$

5. Evaluar  $S(\theta; \theta^{(t+1)})$  usando (4.18).
  6. Si  $|S(\theta; \theta^{(t)}) - S(\theta; \theta^{(t+1)})| < \epsilon$ , hacemos  $\hat{\mu}_j = \delta_j^{(t+1)} + \frac{1}{2} \sigma_j^{2(t+1)}$ ,  $\hat{\sigma}_j = \sqrt{\sigma_j^{2(t+1)}}$  y terminamos, en otro caso vamos a 7.
  7. Hacer  $t = t + 1$  y regresar al paso 2.
- 

La información de Fisher es una forma de medir la cantidad de información que posee una variable aleatoria  $X$  acerca de un parámetro desconocido  $\theta$ . Por lo tanto, mientras más grande sea la información de Fisher, necesitaremos menos observaciones para obtener una buena estimación de un parámetro desconocido. Otra forma de interpretar la información de Fisher es a través de la cota de Crammer-Rao la cual nos dice que si  $\theta$  es un parámetro desconocido y  $\hat{\theta}$  un estimador de dicho parámetro, entonces

$$\text{Var}(\hat{\theta}) \geq \frac{1}{\mathcal{I}(\theta)}.$$

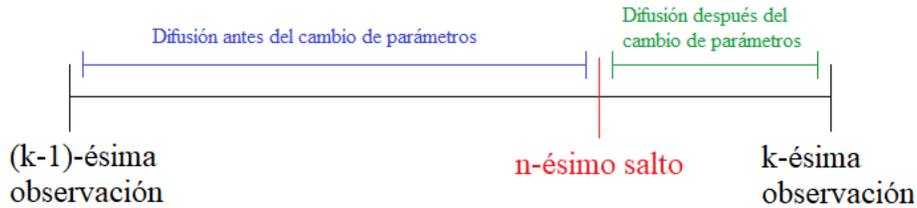
Por lo tanto la varianza de cualquier estimador de un parámetro desconocido, siempre será mayor o igual que  $\mathcal{I}(\theta)^{-1}$ , razón por la cual, vamos a buscar que  $\mathcal{I}(\theta)$  sea lo más grande posible. En el caso de la variable aleatoria Normal( $\mu, \sigma^2$ ) se tiene que la información de Fisher para cada parámetro está dada por

$$\mathcal{I}(\mu) = \frac{1}{\sigma^2}, \quad \mathcal{I}(\sigma) = \frac{1}{2\sigma^4},$$

mientras que para una variable aleatoria Exponencial( $\eta$ ) tenemos que

$$\mathcal{I}(\eta) = \frac{1}{\eta^2}.$$

Por lo tanto, suponer una distribución Exponencial para los saltos es mejor, no solamente por la simplicidad de estimar un solo parámetro en vez de dos, sino que también la variable aleatoria Exponencial requiere menos información para obtener un buen estimador de su parámetro. Además un resultado famoso de Teoría de la Información, dice que la variable aleatoria Exponencial maximiza la entropía para las variables aleatorias con soporte en  $[0, \infty)$ .



**Figura 4.3:** Ocurrencia de  $\tau_n$  entre dos observaciones del proceso.

#### 4.4.4. Estimación de los Saltos

Antes de estimar el parámetro de los saltos, debemos tomar en cuenta un detalle técnico importante. Debido a que nuestras observaciones del Proceso (4.7) son a tiempo discreto y los saltos del valor del proceso son puntuales, es decir ocurren entre dos observaciones del proceso, como se muestra en la Figura 4.3, entonces los rendimientos  $\{y_1, \dots, y_{M-N}\}$  (ordenados cronológicamente para facilitar la notación) que contienen un salto están perturbados por dos términos adicionales, por lo que dichos rendimientos son de la forma

$$y_n = \Delta P_{n,1} + J_n + \Delta P_{n,2},$$

donde

$$\Delta P_{n,1} \sim N(\delta_i(\tau_n - (k-1)), \sigma_i^2(\tau_n - (k-1)))$$

y

$$\Delta P_{n,2} \sim N(\delta_j(k - \tau_n), \sigma_j^2(k - \tau_n)) \quad \text{con } i \neq j.$$

Pues el  $n$ -ésimo rendimiento contiene el  $n$ -ésimo salto donde el PSM salta de un estado  $i$  a otro estado  $j$  en el tiempo  $\tau_n$ , el cual ocurre en el intervalo  $(k-1, k)$  para algún  $k \in \{1, \dots, M\}$ . No obstante, esto no debe preocuparnos tanto, debido a que factorizando tenemos

$$y_n = J_n \left( 1 + \frac{\Delta P_{n,1} + \Delta P_{n,2}}{J_n} \right);$$

donde

$$\left| \frac{\Delta P_{n,1} + \Delta P_{n,2}}{J_n} \right| \approx 0,$$

pues el tamaño de  $J_n$  es un valor independiente de la distancia entre las observaciones del proceso y  $\Delta P_{n,1} + \Delta P_{n,2}$  depende de dicha longitud mientras que el salto  $J_n$  sucede en un intervalo de tamaño infinitesimal. Además, se trata de una suma de variables aleatorias Normales, la cual es una distribución de colas ligeras, adicionalmente, los saltos abruptos que ocurren en los precios superan en gran medida el tamaño de los rendimientos diarios, de modo que podemos afirmar que dicho cociente es muy pequeño y cercano a cero, por lo que la perturbación es despreciable.

Como  $J_n \sim Laplace(0, \eta)$ , entonces  $|J_n| \sim Exponencial(\eta)$  por lo tanto, el EMV de  $\eta$  está dado por

$$\hat{\eta} = \frac{M - N}{\sum_{n=1}^{M-N} |y_n|}. \quad (4.20)$$

#### 4.4.5. Estimación del Generador Infinitesimal

Antes de iniciar con la estimación del generador infinitesimal, recordemos que para tratar el problema de las mezclas en un momento calculamos  $\mathbb{P}(z_i = k|x_i)$ . Es decir, la probabilidad de que la observación  $x_i$  provenga de la  $k$ -ésima densidad, dicha probabilidad estaba dada por

$$\mathbb{P}(z_i = k|x_i) = \gamma_{z_i}(k) = \frac{\pi_k f_k(x_i; \theta_k)}{\sum_{j=1}^m \pi_j f_j(x_i; \theta_j)}.$$

Esto es muy importante pues una vez que hemos optimizado los parámetros de la mezcla, podemos usarlos para hacer la clasificación de las observaciones o en otras palabras, hacer la inferencia acerca de la densidad de la cual proviene cada observación  $x_i$  para  $i = 1, \dots, N$ .

Para que el método de clasificación sea entendible, es necesario que hagamos un cambio en la notación de las observaciones. Recordemos que habíamos separado los rendimientos con salto ordenados cronológicamente  $\{y_1, \dots, y_{M-N}\}$  de los rendimientos sin salto  $\{x_1, \dots, x_N\}$ . Para hacer la clasificación debemos particionar aún más el conjunto de los rendimientos sin salto. Para esto, lo que se hará es usar los rendimientos con salto  $y_i$  para separar los que no tienen salto  $x_j$ . Lo anterior lo podemos hacer, pues una vez que identificamos los saltos  $\{y_1, \dots, y_{M-N}\}$ , tenemos observaciones de rendimientos sin salto  $x_j$  separadas por las  $y_i$ . Es decir, podemos hallar subconjuntos

$$\{x_1^0, \dots, x_{L_0}^0\}, \{x_1^1, \dots, x_{L_1}^1\}, \dots, \{x_1^{M-N}, \dots, x_{L_{M-N}}^{M-N}\},$$

de modo que  $\{x_1^j, \dots, x_{L_j}^j\}$  son los rendimientos que ocurren cronológicamente después del salto  $y_j$  y antes del salto  $y_{j+1}$  para  $j = 1, \dots, M - N - 1$ . Adicionalmente,

$\{x_1^0, \dots, x_{L_0}^0\}$  y  $\{x_1^{M-N}, \dots, x_{L_{M-N}}^{M-N}\}$  son los rendimientos que ocurren antes del primer salto  $y_1$  y después del último salto  $y_{M-N}$  respectivamente.

La idea de hacer la separación es facilitar el algoritmo de clasificación. En los problemas de mezclas, cuando se tienen observaciones  $\{x_1, \dots, x_N\}$  provenientes de una mezcla, se clasifica cada observación por separado. Sin embargo, como sabemos que la difusión, entre saltos tiene parámetros constantes, tenemos la ventaja de que una vez identificados los saltos  $\{y_1, \dots, y_{M-N}\}$ , las observaciones  $\{x_1^j, \dots, x_{L_j}^j\}$  para  $j = 0, \dots, M - N$  provienen de la misma densidad. Por lo tanto, en vez de clasificar cada observación  $x_1, \dots, x_N$  por separado, podemos clasificar cada bloque  $\{x_1^j, \dots, x_{L_j}^j\}$  para aumentar la precisión de la clasificación.

Por lo tanto con la notación que usamos para el problema de las mezclas y usando la independencia, tenemos que la probabilidad de que el bloque  $\{x_1^j, \dots, x_{L_j}^j\}$  provenga de la  $k$ -ésima densidad es

$$\Gamma_j(k) := \mathbb{P}(z_1^j = k, \dots, z_{L_j}^j = k | x_1^j, \dots, x_{L_j}^j) = \prod_{i=1}^{L_j} \mathbb{P}(z_i^j = k | x_i^j) = \prod_{i=1}^{L_j} \gamma_{z_i^j}(k) \quad (4.21)$$

para  $j = 0, \dots, M - N$  y  $k = 1, \dots, m$ .

Recordemos además que cada densidad está asociada con un estado del PSM por lo tanto, la expresión (4.21) se refiere a la probabilidad de que el PSM estuviera en el estado  $k$  al momento de observar los rendimientos  $\{x_1^j, \dots, x_{L_j}^j\}$ . Por otro lado, sabemos que si los rendimientos  $\{x_1^j, \dots, x_{L_j}^j\}$  provienen de la  $k$ -ésima densidad entonces los rendimientos  $\{x_1^{j+1}, \dots, x_{L_{j+1}}^{j+1}\}$  no pueden provenir de la  $k$ -ésima densidad pues un PSM no puede efectuar saltos al mismo estado en el cual se encuentra.

Algo importante que debemos notar es que las probabilidades  $\Gamma_j(k)$  no pueden proporcionarnos una certeza en lo que se refiere a la clasificación. Una posible solución a este problema, sería asignarle a la trayectoria del PSM el estado  $k$  tal que  $\Gamma_j(k)$  sea la probabilidad más alta para cada bloque  $j$  fijo, sin embargo no haremos esto pues afectaría a la aleatoriedad del PSM. Por lo tanto, usaremos las  $\Gamma_j(k)$  para simular el estado en el cual se encontraba el PSM al momento de observar dichos rendimientos. Los detalles del método se presentan en el Algoritmo 17 y puede implementarse dicho algoritmo en el lenguaje de programación R usando el Código A.21.

Una vez que tenemos la trayectoria del PSM reconstruida, podemos implementar el Algoritmo 10 para estimar el Generador Infinitesimal del PSM restringido a que solamente ocurra un salto entre bloques. Finalmente, el Algoritmo 18 indica los pasos a seguir para estimar todos los parámetros.

---

**Algoritmo 17** Algoritmo de Reconstrucción de la trayectoria del PSM
 

---

1.  $j = 0$
  2. Tomar  $\{x_1^j, \dots, x_{L_j}^j\}$  y calcular  $\Gamma_j(k)$  para  $k = 1, \dots, m$ .
  3. Si  $j \neq 0$  hacer  $\Gamma_j(k) = 0$  para el  $k$  elegido en la iteración anterior.
  4. Elegir aleatoriamente  $k \in \{1, \dots, m\}$  con probabilidad proporcional a  $\Gamma_j(k)$ .
  5. Agregar  $L_j + 1$  observaciones del PSM en el estado  $k$ .
  6. Si  $j = M - N$  terminamos, en caso contrario hacer  $j = j + 1$  y volver al paso 2.
- 

---

**Algoritmo 18** Algoritmo de estimación de parámetros para el Modelo de Difusión Markov-Modulado con Saltos
 

---

1. Calcular los log-rendimientos  $\{R_1, \dots, R_M\}$  de la serie.
  2. Identificar los rendimientos  $\{y_1, \dots, y_{M-N}\} \subset \{R_1, \dots, R_M\}$  que contengan saltos fijando una tolerancia.
  3. Aplicar el Algoritmo 16 a los rendimientos  $\{x_1, \dots, x_N\} = \{R_1, \dots, R_M\} \setminus \{y_1, \dots, y_{M-N}\}$  que no contienen salto.
  4. Usar (4.20) y los rendimientos con salto  $\{y_1, \dots, y_{M-N}\}$  para estimar el parámetros de los saltos.
  5. Ordenar por clusters los rendimientos sin salto  $\{x_1, \dots, x_N\}$  para obtener  $\{x_1^0, \dots, x_{L_0}^0\}, \{x_1^1, \dots, x_{L_1}^1\}, \dots, \{x_1^{M-N}, \dots, x_{L_{M-N}}^{M-N}\}$  y reconstruir la trayectoria del PSM mediante el Algoritmo 17.
  6. Usar la trayectoria reconstruida en el Paso 5 para estimar el Generador del PSM usando 10.
- 

## 4.5. Estudio de Simulación

Se concluye la sección con un ejemplo donde se utiliza el Algoritmo 18 para la estimación de los parámetros del modelo. Éste es el algoritmo principal de la tesis pues indica cuáles son los pasos a seguir para estimar todos los parámetros del Modelo de Difusión Markov-Modulado.

Para analizar los resultados proporcionados por los métodos de estimación pre-

sentados en la sección 4.4, crearemos un ejemplo con parámetros similares a los que podríamos hallar en el caso financiero. Supondremos que solamente tenemos 3 estados para la economía, es decir el espacio de estados para el PSM es  $E = \{1, 2, 3\}$ . Como se explicó anteriormente, el estado 1 representa un ambiente financiero óptimo para el crecimiento del precio de la acción, el estado 2 representa un ambiente financiero con volatilidad en el mercado y 3 representa crisis en el sector de la acción. Supongamos además, que los ambientes duran en promedio 1.5 años,  $2/3$  de año y  $1/2$  año respectivamente, las probabilidades de transición entre cada estado serán  $p_{12} = 7/8$ ,  $p_{13} = 1/8$ ,  $p_{21} = 2/3$ ,  $p_{23} = 1/3$ ,  $p_{31} = 3/4$  y  $p_{32} = 1/4$ . Entonces por la Proposición 2.3.5, el Generador del PSM está dado por

$$Q = \begin{pmatrix} -\frac{2}{3} & \frac{7}{12} & \frac{1}{12} \\ 1 & -\frac{3}{2} & \frac{1}{2} \\ \frac{3}{2} & \frac{1}{2} & -2 \end{pmatrix}.$$

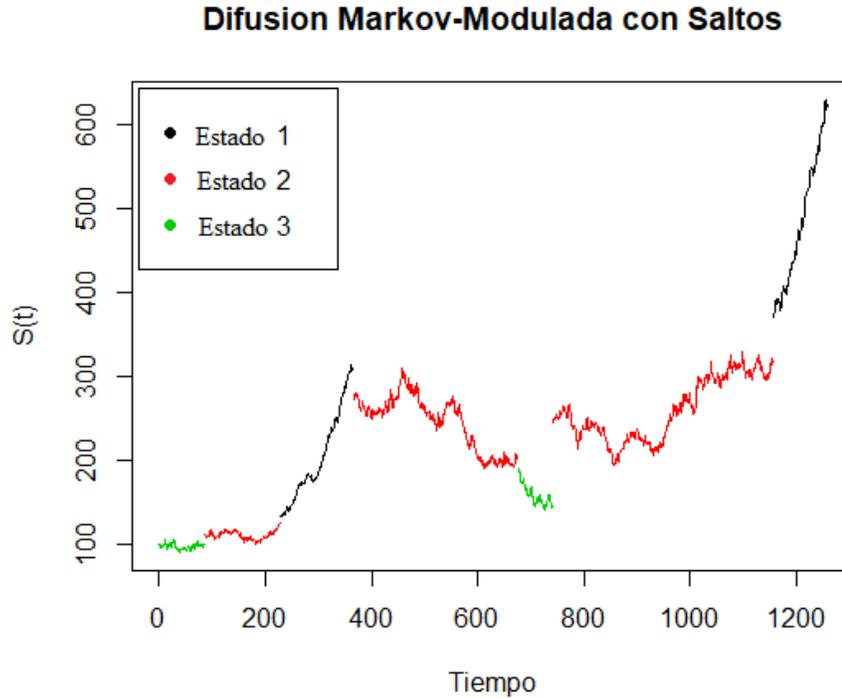
Además, la media y volatilidad anual de la acción para cada estado es  $\mu_1=1.5$ ,  $\sigma_1=0.15$ ,  $\mu_2=0.3$ ,  $\sigma_2 = \sqrt{0.08}$ ,  $\mu_3=-0.25$  y  $\sigma_3=0.35$ . Para simplificar, vamos a escalar todos los parámetros de modo que la unidad de tiempo sea el día en vez de el año, los parámetros escalados se presentan en el Cuadro 4.1. Se simularon observaciones del modelo (4.7) con los siguientes parámetros:  $\eta = 250/33$ ,  $\Delta t = 1$ ,  $S_0 = 100$ ,  $\mathbb{P}(X_0 = 1) = 1$ ,

$$Q = \begin{pmatrix} -0.002645503 & 0.002314815 & 0.00033068785 \\ 0.003968254 & -0.005952381 & 0.0019841270 \\ 0.005952381 & 0.001984127 & -0.0079365079 \end{pmatrix}, \quad (4.22)$$

j	$\mu_j$	$\sigma_j$
1	0.0059523810	0.009449112
2	0.0011904762	0.017817415
3	-0.0009920635	0.022047928

**Cuadro 4.1:** Medias y Volatilidades de la simulación

y horizontes de tiempo  $T = 1260, 2520, 3780, 5040, 6300, 7560, 8820$  (que representan 5, 10, 15, 20, 25, 30 y 35 años, respectivamente). Se muestra una trayectoria del proceso con estos parámetros en la Figura 4.4.



**Figura 4.4:** Una trayectoria del modelo con  $T = 1260$  y los parámetros del Cuadro 4.1.

Los resultados de las estimaciones con tolerancias  $\pm 0.069$  para los saltos y  $\epsilon = 0.05$  para el Algoritmo 16, se muestran en los Cuadros 4.2 y 4.3, los  $\pi_j$  de la mezcla para el caso  $T = 8820$  se muestran en el Cuadro 4.4 pues son los que se utilizarán para la estimación del Generador Infinitesimal.

	$j = 1$	$j = 2$	$j = 3$	Error
$\mu_j$	0.0059523810	0.0011904762	-0.0009920635	
$\hat{\mu}_j^{1260}$	0.0059469097	0.0027668870	0.0019743179	0.00001128
$\hat{\mu}_j^{2520}$	0.0066422130	0.0034594822	0.0025009336	0.00001782
$\hat{\mu}_j^{3780}$	0.0066102728	0.0032372128	0.0020031847	0.00001359
$\hat{\mu}_j^{5040}$	0.0063053813	0.0028402039	0.0020858060	0.00001231
$\hat{\mu}_j^{6300}$	0.0059171341	0.0031332369	0.0017120427	0.00001108
$\hat{\mu}_j^{7560}$	0.0060495312	0.0030009829	0.0013917915	0.00000897
$\hat{\mu}_j^{8820}$	0.0060614045	0.0029881755	0.0008520252	0.00000664

**Cuadro 4.2:** Resultados de las Estimaciones de  $\mu_j$ .

	$j = 1$	$j = 2$	$j = 3$	Error
$\sigma_j$	0.009449112	0.017817415	0.022047928	
$\hat{\sigma}_j^{1260}$	0.012769167	0.015921710	0.017703704	0.00003348
$\hat{\sigma}_j^{2520}$	0.009535770	0.015774837	0.017788880	0.00002231
$\hat{\sigma}_j^{3780}$	0.009327915	0.015878260	0.018081729	0.00001950
$\hat{\sigma}_j^{5040}$	0.009104289	0.015999326	0.017911584	0.00002053
$\hat{\sigma}_j^{6300}$	0.009154862	0.015454727	0.018728368	0.00001668
$\hat{\sigma}_j^{7560}$	0.009237009	0.014904877	0.014904877	0.00001700
$\hat{\sigma}_j^{8820}$	0.014904877	0.014492026	0.019310947	0.00001854

**Cuadro 4.3:** Resultados de las Estimaciones de  $\sigma_j$  obtenidos mediante el Algoritmo 16.

$\pi_1$	$\pi_2$	$\pi_3$
0.5522	0.2712	0.1766

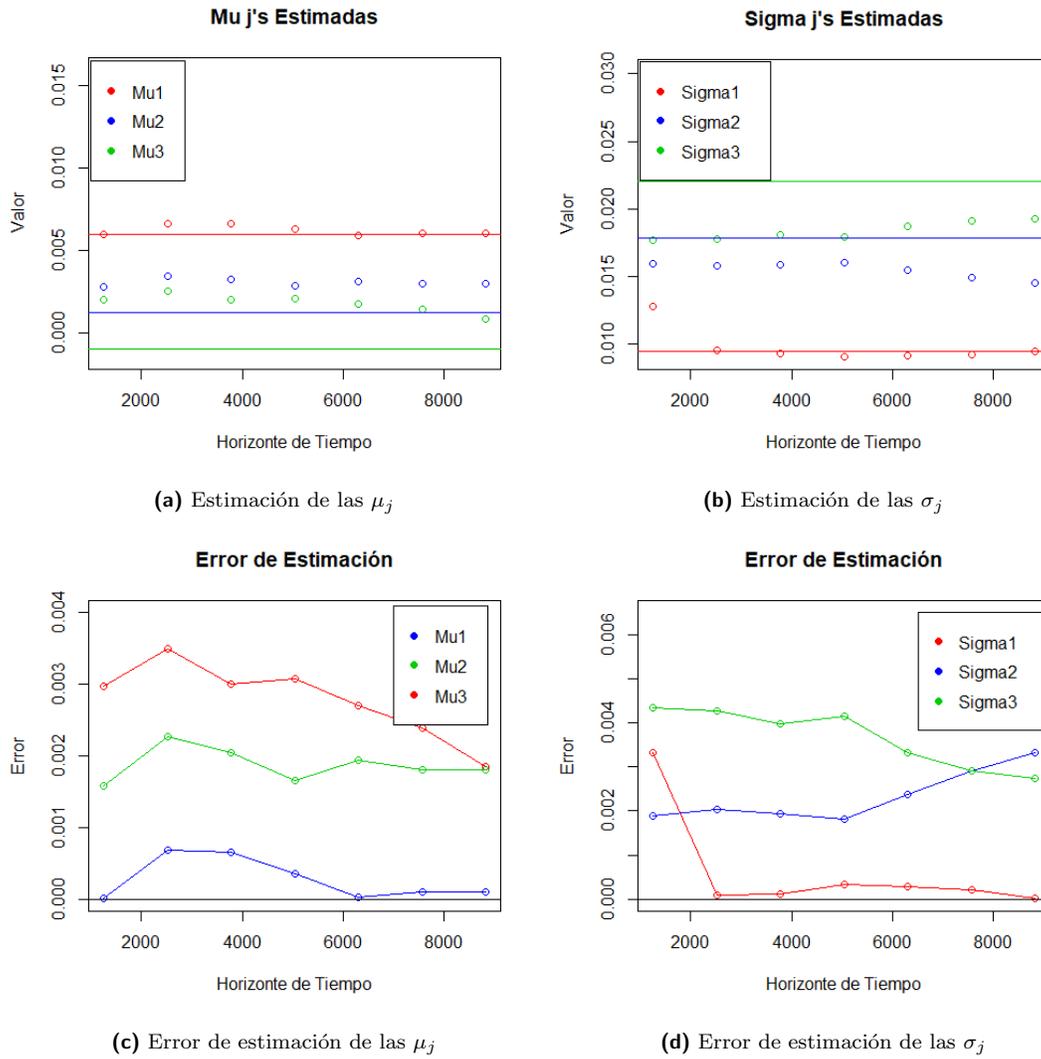
**Cuadro 4.4:**  $\pi_j$  óptimos de la mezcla para  $T = 8820$  obtenidos con el Algoritmo 16.

Como se puede apreciar en la Figura 4.5 conforme se aumenta el horizonte de tiempo, el error de los estimadores disminuye. Sin embargo, una vez que cruzamos del horizonte  $T = 6300$  la velocidad con la que desciende el error es significativamente menor a la del inicio; es decir, la precisión que se gana a partir de ese punto es casi despreciable.

De igual forma, se estimó  $\eta$  usando las mismas realizaciones para los distintos horizontes de tiempo, los resultados se muestran en el Cuadro 4.5.

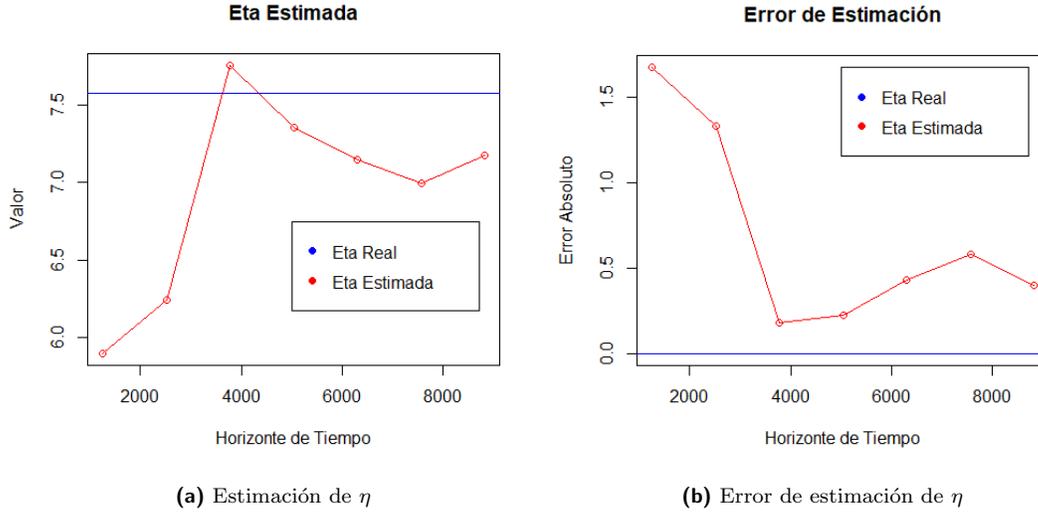
	Valor	Error
$\eta$	7.575757	
$\hat{\eta}^{1260}$	5.897666	1.678092
$\hat{\eta}^{2520}$	6.241154	1.334603
$\hat{\eta}^{3780}$	7.756247	0.1804897
$\hat{\eta}^{5040}$	7.352301	0.2234567
$\hat{\eta}^{6300}$	7.144313	0.4314449
$\hat{\eta}^{7560}$	6.994469	0.5812881
$\hat{\eta}^{8820}$	7.175147	0.4006109

**Cuadro 4.5:** Resultados de las Estimaciones de  $\eta$  usando (4.20).



**Figura 4.5:** Gráficas de la estimación de los parámetros de la difusión mediante el Algoritmo 16.

En la Figura 4.6 podemos observar que, la convergencia al valor real del parámetro es rápida, debido a que a partir del horizonte de tiempo  $T = 3780$  el valor de  $\hat{\eta}$  se acerca mucho al real.

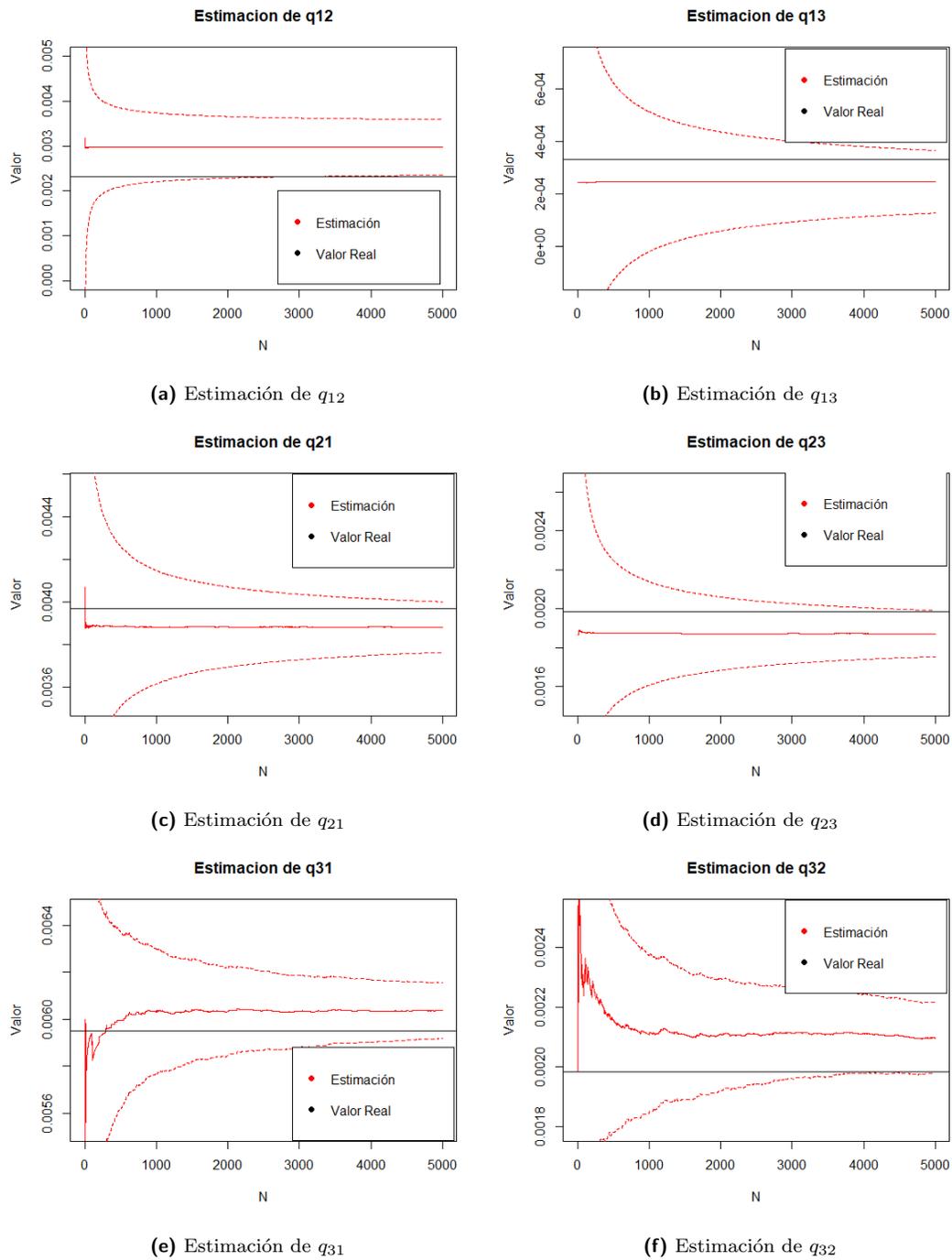


**Figura 4.6:** Gráficas de la Estimación de  $\eta$  usando (4.20).

Por último, aunque en la Sección 3.2.2.1 ya presentamos los resultados de una estimación de un Generador Infinitesimal es necesario cercioranos de que la reconstrucción del PSM a través de la información proporcionada por la difusión es suficiente para obtener un buen estimador del Generador, en (4.23) se muestra el estimador obtenido y en la Figura 4.7 se exponen los resultados de la estimación de forma gráfica. Primero tenemos que

$$\hat{Q} = \begin{pmatrix} -0.003215818 & 0.002969198 & 0.0002466196 \\ 0.003882034 & -0.005754555 & 0.0018725209 \\ 0.006037886 & 0.002097409 & -0.0081352957 \end{pmatrix}; \quad (4.23)$$

mientras que el error del estimador (4.23) fue 0.00767666. Es importante recordar que (4.23) se obtuvo mediante el método alternativo descrito en el Ejemplo de la Sección 3.2.2, donde se toma el promedio de los estimadores proporcionados por el Algoritmo 10 después de un tiempo de calentamiento de 1000 observaciones.



**Figura 4.7:** Estimación del Generador para el caso donde no usamos el parámetro real con horizonte de tiempo  $T = 8820$  e intervalos de confianza al 95 % (línea punteada).



# Capítulo 5

## Ajuste del Modelo de Difusión Markov-Modulado con Saltos a Datos Reales

En este capítulo se utilizarán los métodos de estimación desarrollados en el Capítulo 4 para hacer el ajuste a una serie de datos reales. Esta vez la estimación de los parámetros está sujeta a condiciones distintas debido a que en el estudio de simulación en la Sección 4.5 se conocían los parámetros reales del Modelo de Difusión Markov-Modulado con Saltos y se podía calcular el error en las estimaciones para asegurarse de la precisión de los estimadores. Sin embargo, en este caso no contamos con los parámetros reales de los datos escogidos, por esto no se puede asegurar que la estimación sea del todo precisa. Una forma de verificar los resultados obtenidos mediante la estimación es comparar los resultados no para una, sino para dos series financieras dentro del mismo sector. Si se toman los datos históricos de los precios de las acciones de dos empresas que pertenezcan al mismo sector, es de esperarse que el Generador Infinitesimal del PSM estándar (que recordemos, modela el ambiente financiero favorable o desfavorable para la acción) sea muy parecido pues las condiciones macroeconómicas que le favorecen a la empresa de un sector, deben ser también favorables para otras acciones del su mismo sector y viceversa para las condiciones macroeconómicas desfavorables.

### 5.1. Dos Acciones del bloque FAANG

Amazon y Netflix son empresas tecnológicas que cotizan en la bolsa de valores NASDAQ. Éstas forman parte de un bloque de 5 empresas, dentro del cual se encuentran Facebook, Amazon, Apple, Netflix y Google es ampliamente conocido y es comúnmente denominado como FAANG, debido a que está conformado por las 5 empresas norteamericanas de la industria tecnológica con mejor desempeño en los

mercados financieros. Estas empresas están dentro de las más influyentes en su sector, además no solamente son conocidas a nivel mundial, sino que también su índice de capitalización (el cual es una medida de una empresa o su dimensión económica) en conjunto, supera los 4.1 trillones de dólares. Son compañías muy importantes a nivel mundial y debido a la relación que guardan entre ellas por estar dentro del mismo sector, frecuentemente se habla de éstas en conjunto bajo su acrónimo FAANG. Debido a lo anterior, es plausible asumir que el comportamiento del precio de las acciones que conforman FAANG está sujeto a condiciones macroeconómicas similares.

Con el objetivo de verificar la coherencia del Modelo de Difusión Makov-Modulado con Saltos bajo la estimación que se hará en este capítulo, se escogieron las series de precios de las acciones correspondientes a Netflix y Amazon. Como se detalló anteriormente, al formar parte de FAANG, ambas están sujetas a condiciones macroeconómicas similares.



(a) Precios de la acción de Netflix (USD) del 1 de julio de 2005 al 30 de junio de 2020.



(b) Precios de la acción de Amazon (USD) del 1 de julio de 2005 al 30 de junio de 2020.

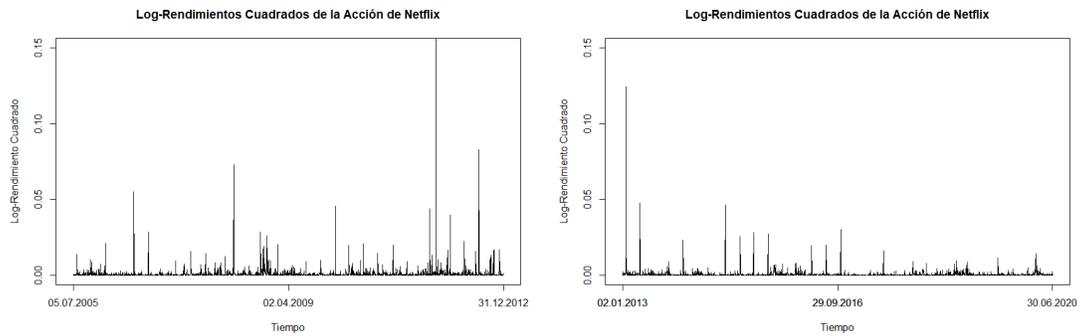
**Figura 5.1:** Comparación de los precios de las acciones de Amazon y Netflix (USD) para el mismo rango de fechas (en un rango de 15 años).

En la Figura 5.1, se muestran las dos series para las cuales se hará el ajuste del Modelo de Difusión Markov-Modulado con Saltos. Puede observarse que ambas

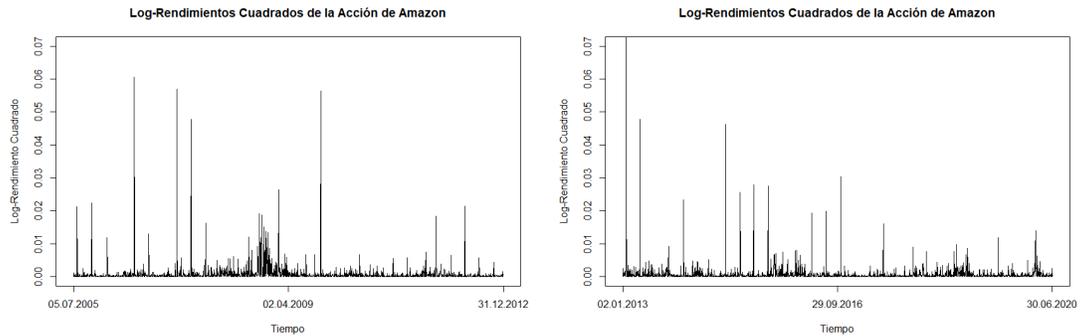
acciones presentan un comportamiento similar a lo largo del mismo rango de fechas.

## 5.2. El Problema de la Identificación de los Saltos

Recordemos que el Algoritmo 18 hace el ajuste del Modelo de Difusión Markov-Modulado. Así, el Paso 1 indica que se deben calcular los rendimientos de la serie, en este caso para no tener problemas con el signo de los log-rendimientos de la serie, se calcularán los log-rendimientos cuadrados de los datos, mismos que se muestran en la Figura 5.2 y a los cuales les llamaremos rendimientos. Al igual que en la Sección 4.5 se supondrá que el PSM que modela el ambiente económico de la acción tiene 3 estados.



(a) Log-Rendimientos cuadrados de la acción de Netflix desde el 5 de julio de 2005 hasta el 31 de diciembre de 2012. (b) Log-Rendimientos cuadrados de la acción de Netflix desde el 2 de enero de 2013 hasta el 30 de junio de 2020.



(c) Log-Rendimientos cuadrados de la acción de Amazon desde el 5 de julio de 2005 hasta el 31 de diciembre de 2012. (d) Log-Rendimientos cuadrados de la acción de Amazon desde el 2 de enero de 2013 hasta el 30 de junio de 2020.

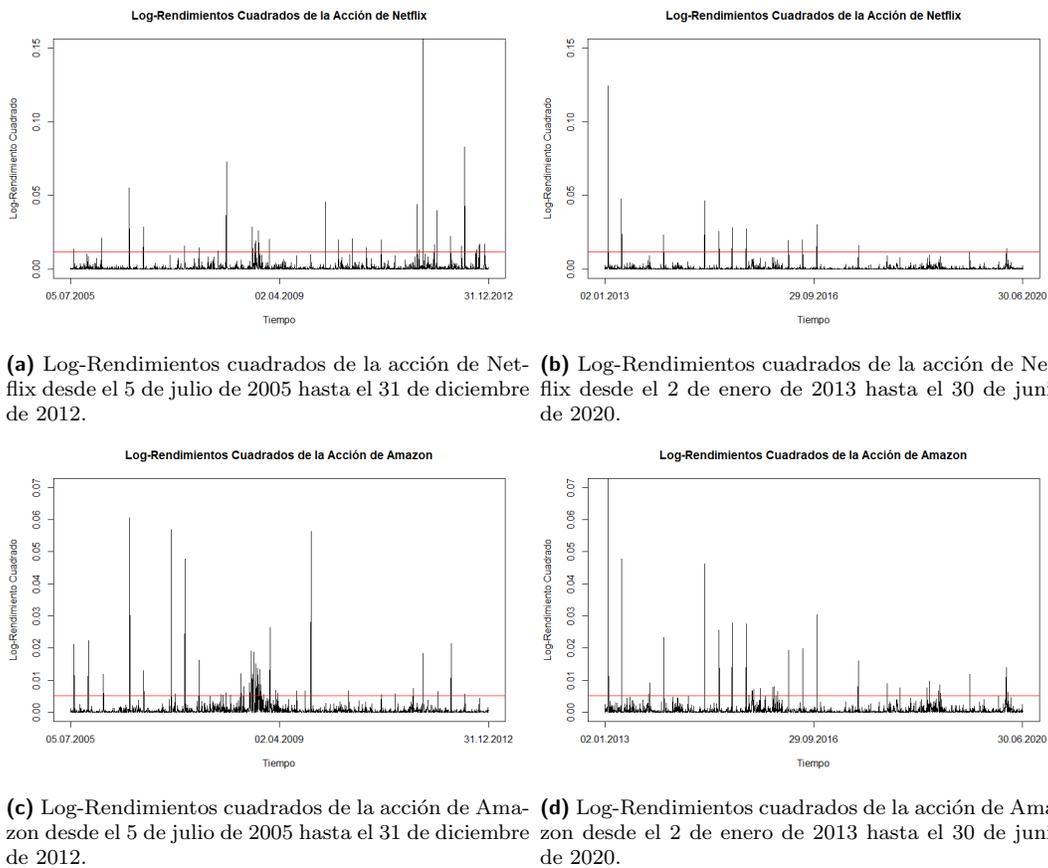
**Figura 5.2:** Log-Rendimientos cuadrados de las acciones de Netflix y Amazon (en USD) desde el 5 de julio de 2005 hasta el 30 de junio de 2020.

De acuerdo al Algoritmo 18, el Paso 2 es analizar los rendimientos de la serie y así fijar una tolerancia para los saltos de la difusión.

Con el objetivo de captar el cambio de dinámica que sufre el precio de la acción de Netflix a partir del 12 de marzo de 2020, que se aprecia en las observaciones finales de la Figura 5.2b, pues a partir de ese mes varios países comenzaron a entrar en confinamiento debido a la situación mundial por el COVID-19 y la acción de Netflix obtuvo una tendencia alcista significativa. Por esto, la tolerancia a los saltos para la acción de Netflix se fijó en  $\pm\sqrt{0.01174792}$ .

Por otro lado, el 31 de enero de 2020, la acción de Amazon tuvo un cambio positivo en su precio que daría pie a que días después el precio de su acción alcanzara un máximo histórico hasta ese momento, claramente esto puede ser visto como un cambio en la dinámica de la acción debido a que ese día el volumen de la acción fue el triple del valor que tenía hasta ese momento. Esta es la razón por la cual, la tolerancia que se le dará a los rendimientos es  $\pm\sqrt{0.005068828}$ .

La Figura 5.3 muestra la tolerancia aplicada a los rendimientos cuadrados para separar los cambios de dinámica de las acciones.



**Figura 5.3:** Tolerancia aplicada a los Log-Rendimientos cuadrados de las acciones de Netflix y Amazon (en USD).

Ya con la tolerancia fija, se considera que los rendimientos que contienen un salto, son aquellos que excedan la tolerancia previamente establecida, dichos rendimientos para la acción de Netflix son los correspondientes a las fechas mostradas en el Cuadro 5.1 y de manera equivalente los correspondientes a la acción de Amazon se muestran en el Cuadro 5.2.

Fecha	Rendimiento	Fecha	Rendimiento
26 julio 2005	0.11686434	26 enero 2012	0.19899571
25 enero 2006	0.14479912	24 abril 2012	-0.14946522
25 julio 2006	-0.23501313	5 julio 2012	0.12584890
24 octubre 2006	0.16961491	25 julio 2012	-0.28768207
23 julio 2007	-0.1254013	9 octubre 2012	-0.1149300
23 octubre 2007	0.1201443	24 octubre 2012	-0.1266685
27 febrero 2008	0.1118611	26 octubre 2012	0.1229523
22 abril 2008	-0.27004493	31 octubre 2012	0.13000405
7 octubre 2008	-0.16970309	4 diciembre 2012	0.13099595
21 octubre 2008	-0.1318521	24 enero 2013	0.35232652
28 octubre 2008	0.1384023	25 enero 2013	0.1436091
13 noviembre 2008	0.16184303	23 abril 2013	0.21871788
27 enero 2009	0.14248194	23 enero 2014	0.15241139
28 enero 2010	0.21320421	16 octubre 2014	-0.21525496
22 abril 2010	0.14154426	21 enero 2015	0.15989659
22 julio 2010	-0.14454222	16 abril 2015	0.16731454
21 octubre 2010	0.120015	16 julio 2015	0.16565779
27 enero 2011	0.14134021	19 abril 2016	-0.13892281
15 septiembre 2011	-0.20965346	19 julio 2016	-0.14071372
29 septiembre 2011	-0.1160637	18 octubre 2016	0.17418904
25 octubre 2011	-0.4296058	18 julio 2017	0.12701671
9 enero 2012	0.1291626	16 marzo 2020	-0.1180953

**Cuadro 5.1:** Fechas y valor de los rendimientos con salto en la acción de Netflix del 5 de julio de 2005 al 30 de junio de 2020 con tolerancia  $\pm\sqrt{0.01174792}$ .

Del Cuadro 5.1 podemos observar que hay un problema en considerar dichas fechas como saltos. El primer problema es que se consideran las fechas 7, 21 y 28 de octubre de 2008. Por lo que bajo el supuesto de que la acción de Netflix se comporta de acuerdo al Modelo de Difusión Markov-Modulado con Saltos, significaría que el cambio de dinámica de la difusión ocurrió después de pasar 14 y 7 días respectivamente en un estado de la economía. Lo anterior es poco apegado a la realidad pues los cambios en la economía que puedan llegar a afectar el movimiento de las acciones no duran 14 días ni mucho menos 7, por lo que es más probable es que los rendimientos

Fecha	Rendimiento	Fecha	Rendimiento
27 julio 2005	0.14548274	3 diciembre 2008	0.09312279
26 octubre 2005	-0.14972040	16 diciembre 2008	0.07453192
3 febrero 2006	-0.10890237	30 enero 2009	0.16245893
26 julio 2006	-0.24618153	10 marzo 2009	0.08277306
25 octubre 2006	0.11371092	23 marzo 2009	0.07726805
25 abril 2007	0.23862092	24 julio 2009	-0.08188205
26 abril 2007	0.09992418	16 septiembre 2009	0.08211210
21 mayo 2007	0.07602444	23 octubre 2009	0.23740203
25 julio 2007	0.21871499	29 junio 2010	-0.08122479
23 octubre 2007	0.09929550	28 enero 2011	-0.07489649
24 octubre 2007	-0.12773870	27 abril 2011	0.07567011
18 marzo 2008	0.07483777	23 agosto 2011	0.08633994
1 abril 2008	0.07300538	26 octubre 2011	-0.13532540
18 abril 2008	0.07867037	1 febrero 2012	-0.08017129
19 mayo 2008	0.07348187	27 abril 2012	0.14622537
24 julio 2008	0.10971733	27 julio 2012	0.07573644
11 agosto 2008	0.08997762	26 abril 2013	-0.07516142
17 septiembre 2008	-0.09576754	25 octubre 2013	0.08970934
22 septiembre 2008	-0.07789481	31 enero 2014	-0.11650287
29 septiembre 2008	-0.10977067	25 abril 2014	-0.10405960
30 septiembre 2008	0.13849145	25 julio 2014	-0.10146107
7 octubre 2008	-0.10855091	24 octubre 2014	-0.08708685
9 octubre 2008	-0.08584999	30 enero 2015	0.12849516
13 octubre 2008	0.09765087	24 abril 2015	0.13217788
14 octubre 2008	-0.10460835	24 julio 2015	0.09346458
15 octubre 2008	-0.13675894	28 enero 2016	0.08538868
28 octubre 2008	0.12247819	29 enero 2016	-0.07915121
5 noviembre 2008	-0.11731266	29 abril 2016	0.09136098
6 noviembre 2008	-0.09604150	27 octubre 2017	0.12413063
12 noviembre 2008	-0.10800379	26 octubre 2018	-0.08142353
12 noviembre 2008	0.07796756	26 diciembre 2018	0.09025398
14 noviembre 2008	-0.07340627	31 enero 2020	0.07119570
21 noviembre 2008	0.07795440	12 marzo 2020	-0.08253502
24 noviembre 2008	0.11534483	1 mayo 2020	-0.07901521

**Cuadro 5.2:** Fechas y valor de los rendimientos con salto en la acción de Amazon del 5 de julio de 2005 al 30 de junio de 2020 con tolerancia  $\pm\sqrt{0.005068828}$ .

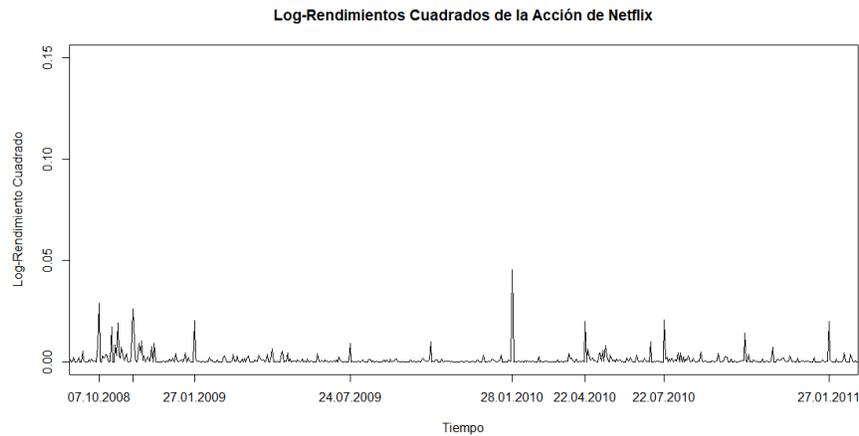
del 21 y 28 de octubre del 2008 sean la consecuencia del cambio de dinámica después del salto efectuado el 7 de octubre de 2008.

Otro problema es que algo similar a lo anterior ocurre el 15 y el 29 de septiembre de 2011, de nuevo el cambio en la dinámica se daría después de permanecer solamente 14 días en un estado, por lo que, al igual que en el caso anterior, el rendimiento del 29 de septiembre de 2011 es probable que sea producto del cambio de dinámica ocurrido el 15 de septiembre de 2011.

Por otro lado, se puede observar en el Cuadro 5.2 que con la acción de Amazon ocurre el mismo problema que se tiene con las fechas de saltos de la acción de Netflix, sin embargo, esto ocurre con mayor frecuencia. Por ejemplo si suponemos que la acción de Amazon obedece al Modelo de Difusión Markov-Modulado con Saltos, se tendría que, en los meses de octubre y noviembre del año 2008, la acción cambiaría de dinámica 6 y 7 veces respectivamente en el mismo mes. Lo cual de nuevo supone un problema desde el punto de vista económico. De igual forma, sería más plausible suponer que los rendimientos adicionales en esos mismos meses obedecen a los parámetros después un cambio de dinámica en vez de pensar que está cambiando cada vez que se sobrepasa el umbral establecido anteriormente.

Los Cuadros 5.1 y 5.2 muestran más situaciones similares a las que se describieron anteriormente, esta es la razón por la cual será necesario hacer una identificación manual de los saltos. Suponiendo la dinámica del Modelo de Difusión Markov-Modulado con Saltos, se deben identificar clusters de rendimientos en los cuales pueden observarse rendimientos con un comportamiento similar entre ellos, dichos clusters deberán estar delimitados por un rendimiento que pueda ser caracterizado como un salto o mejor dicho, los clusters deberán estar delimitados por un rendimiento a partir del cual sea posible apreciar un “antes” y un “después” en el comportamiento de los rendimientos.

Para ilustrar lo anterior, tomaremos como ejemplo los rendimientos de la acción de Netflix mostrados en la Figura 5.4. En dicha figura, podemos observar que el 7 de octubre de 2008 hay un rendimiento que destaca por marcar una diferencia en los rendimientos, ya que después de éste se observan rendimientos (cuadrados) más pronunciados, mismos que después del 13 de noviembre de 2008 continúan siendo pronunciados pero en menor escala, para así finalmente volver a ser pequeños después del rendimiento ubicado el 27 de enero de 2009. Más tarde el 24 de julio de 2009 se observa otro cambio de comportamiento pues los rendimientos se vuelven menos grandes. De igual forma el 22 de abril de 2010 marca una diferencia en el comportamiento de los rendimientos antes y después.



**Figura 5.4:** Separación de los rendimientos de la acción de Netflix (del 1 de septiembre de 2008 al 5 de febrero de 2011) en las fechas marcadas que delimitan los clusters en los cuales se infiere que los rendimientos de la difusión tienen los mismos parámetros.

Fecha	Rendimiento
26 julio 2005	0.11686434
25 enero 2006	0.14479912
25 julio 2006	-0.23501313
24 octubre 2006	0.16961491
22 abril 2008	-0.27004493
7 octubre 2008	-0.16970309
13 noviembre 2008	0.16184303
27 enero 2009	0.14248194
24 julio 2009	-0.09636495
28 enero 2010	0.21320421
22 abril 2010	0.14154426
22 julio 2010	-0.14454222
27 enero 2011	0.14134021
15 septiembre 2011	-0.20965346
26 enero 2012	0.19899571
24 abril 2012	-0.14946522
5 julio 2012	0.12584890
25 julio 2012	-0.28768207
31 octubre 2012	0.13000405
4 diciembre 2012	0.13099595
24 enero 2013	0.35232652
23 abril 2013	0.21871788
22 octubre 2013	-0.09596115
23 enero 2014	0.15241139
16 octubre 2014	-0.21525496
21 enero 2015	0.15989659
16 abril 2015	0.16731454
16 julio 2015	0.16565779
19 abril 2016	-0.13892281
19 julio 2016	-0.14071372
18 octubre 2016	0.17418904
18 julio 2017	0.12701671
23 enero 2018	0.09511842
17 abril 2018	0.08790447
10 octubre 2018	-0.08755589
4 enero 2019	0.09279294
18 julio 2019	-0.10838782
12 marzo 2020	-0.10433858
13 abril 2020	0.06778368

**Cuadro 5.3:** Fechas y valor de los rendimientos con salto en la acción de Netflix del 5 de julio de 2005 al 30 de junio de 2020 identificados de forma manual.

Fecha	Rendimiento	Fecha	Rendimiento
27 julio 2005	0.14548274	26 octubre 2012	0.06646561
26 octubre 2005	-0.14972040	26 abril 2013	-0.07516142
3 febrero 2006	-0.10890237	25 octubre 2013	0.08970934
26 julio 2006	-0.24618153	31 enero 2014	-0.11650287
25 octubre 2006	0.11371092	25 abril 2014	-0.10405960
25 abril 2007	0.23862092	25 julio 2014	-0.10146107
25 julio 2007	0.21871499	24 octubre 2014	-0.08708685
24 octubre 2007	-0.12773870	30 enero 2015	0.12849516
24 julio 2008	0.10971733	24 abril 2015	0.13217788
17 septiembre 2008	-0.09576754	24 julio 2015	0.09346458
30 enero 2009	0.16245893	29 abril 2016	0.09136098
24 julio 2009	-0.08188205	28 octubre 2016	-0.05273753
23 octubre 2009	0.23740203	3 febrero 2017	-0.03606124
29 junio 2010	-0.08122479	27 octubre 2017	0.12413063
28 enero 2011	-0.07489649	2 abril 2018	-0.05346515
27 abril 2011	0.07567011	10 octubre 2018	-0.06349824
23 agosto 2011	0.08633994	26 diciembre 2018	0.09025398
26 octubre 2011	-0.13532540	3 junio 2019	-0.04752088
1 febrero 2012	-0.08017129	31 enero 2020	0.07119570
27 abril 2012	0.14622537	1 mayo 2020	-0.07901521
27 julio 2012	0.07573644		

**Cuadro 5.4:** Fechas y valor de los rendimientos con salto en la acción de Amazon del 5 de julio de 2005 al 30 de junio de 2020 identificados de forma manual.

Aplicando este proceso a ambas series, es posible identificar los cambios de comportamiento en los rendimientos y registrar como “saltos” los rendimientos que delimiten dichos comportamientos.

Así, el Cuadro 5.3 muestra los rendimientos de la acción de Netflix que se considerarán como saltos debido a que cumplen con la regla especificada anteriormente para la identificación manual de los saltos en los rendimientos y el Cuadro 5.4 expone los correspondientes a la acción de Amazon.

Algo que es necesario recalcar es que, con la identificación manual de los saltos se le da coherencia al Modelo de Difusión Markov-Modulado con Saltos. Lo anterior, en el sentido de que los saltos del PSM subyacente no deberían ocurrir en días tan juntos entre sí, debido a que tendríamos que el tiempo de estancia en un estado de la economía sería muy cercano a un día, lo que provocaría inconsistencia con lo que se observa en la realidad. Además, identificar de forma manual los saltos también permite capturar parte del comportamiento cíclico que es propio de las series de tiempo

financieras.

Como se muestra en el Cuadro 5.5, la mayoría de los saltos tanto para Netflix como para Amazon, están concentrados alrededor de los meses: enero, abril, julio y octubre y éstos ocurren usualmente en la misma semana, lo que nos confirma que los ambientes financieros que favorecen o desfavorecen el precio de las acciones de dichas empresas, es similar y es reflejado adecuadamente mediante la identificación manual de los saltos.

2005				2006				2007			
01	04	07	10	01	04	07	10	01	04	07	10
-	-	26	-	25	-	25	24	-	-	-	-
-	-	27	26	3/02	-	26	25	-	25	25	24
2008				2009				2010			
01	04	07	10	01	04	07	10	01	04	07	10
-	22	-	7,13/11	27	-	24	-	28	22	22	-
-	-	24	17/09	30	-	24	23	-	-	29/06	-
2011				2012				2013			
01	04	07	10	01	04	07	10	01	04	07	10
27	-	-	15/09	26	24	5,25	31	24	23	-	22
28	27	23/08	26	1/02	27	27	26	-	26	-	25
2014				2015				2016			
01	04	07	10	01	04	07	10	01	04	07	10
23	-	-	16	21	16	16	-	-	19	19	18
31	25	25	24	30	24	24	-	-	29	-	28
2017				2018				2019			
01	04	07	10	01	04	07	10	01	04	07	10
-	-	18	-	23	17	-	10	4	-	18	-
3/02	-	-	27	-	2	-	10	26/12	-	3/06	-
2020											
01	04	07	10								
-	13	-	-								
31	1/05	-	-								

**Cuadro 5.5:** Fechas de saltos concentradas alrededor de los meses: Enero, Abril, Julio y Octubre. Las fechas en rojo corresponden a los saltos de la acción de Netflix y las azules corresponden a Amazon.

### 5.3. Ajuste del Modelo a las series de Netflix y Amazon

En la sección 5.2 se resolvió el problema de la identificación de los saltos, pues como indica el Algoritmo 18, es necesario identificarlos antes de empezar con los métodos de estimación.

Ya con los saltos identificados podemos pasar a la estimación de los parámetros del Modelo de Difusión Markov-Modulado que comienza con el Paso 3 del Algoritmo 18. Por lo anterior, los rendimientos mostrados en los Cuadros 5.3 y 5.4, son los que quedarán excluidos del Paso 3 del Algoritmo 18, pues se hará la estimación de los parámetros de la difusión  $\{\mu_j\}_{j=1,2,3}$  y  $\{\sigma_j\}_{j=1,2,3}$  vía el Algoritmo 16.

De acuerdo con el Algoritmo 16 es necesario iniciar con parámetros  $\{\mu_j^{(0)}\}_{j=1,2,3}$ ,  $\{\sigma_j^{(0)}\}_{j=1,2,3}$  y  $\{\pi_j^{(0)}\}_{j=1,2,3}$ , por lo cual para ambas acciones, tomaremos de estimaciones iniciales los parámetros reales del ejemplo en la sección 4.5 que se muestran en el Cuadro 4.1 y los  $\pi_j$  expuestos en el Cuadro 4.4.

Por último, la tolerancia  $\epsilon$  necesaria para la condición de paro en el Algoritmo 16 será  $\epsilon = 0.05$ . Los resultados de la estimación se presentan en los Cuadros 5.6 y 5.7.

j	$\hat{\pi}_j$	$\hat{\mu}_j$	$\hat{\sigma}_j$
1	0.44797175	-0.00026218670	0.01402849
2	0.53779080	0.0031193879	0.03213458
3	0.01423745	-0.00634970790	0.09737263

**Cuadro 5.6:** Pesos, medias y volatilidades (diarias) de la acción de Netflix estimadas mediante el Algoritmo 16 con tolerancia  $\epsilon = 0.05$ .

j	$\hat{\pi}_j$	$\hat{\mu}_j$	$\hat{\sigma}_j$
1	0.45787160	0.0009578106	0.01093079
2	0.48193159	0.0012029475	0.02213378
3	0.06019681	0.0041515128	0.04834662

**Cuadro 5.7:** Pesos, medias y volatilidades (diarias) de la acción de Amazon estimadas mediante el Algoritmo 16 con tolerancia  $\epsilon = 0.05$ .

Desde este momento se pueden apreciar algunas similitudes con respecto a la estimación de los pesos  $\{\pi_j\}_{j=1,2,3}$ , pues la proporción de la mezcla coincide en que ambas tienen más de 0.9 de proporción en los dos primeros estados de la economía con alrededor de 0.47 cada una. Esto es buena señal pues de acuerdo al modelo, deberían tener proporciones similares debido a que las condiciones macroeconómicas a las que

están sujetas ambas empresas son similares.

Para el Paso 4 del Algoritmo 18, solamente usamos (4.20) para estimar el parámetro  $\eta$  de la variable aleatoria que rige el comportamiento de los saltos usando los rendimientos del Cuadro 5.3 para la acción de Netflix y el Cuadro 5.4 para la acción de Amazon.

La estimación del parámetro  $\eta$  obtuvo como resultado  $\hat{\eta} = 6.359798$  para la acción de Netflix y  $\hat{\eta} = 9.119797$  para la acción de Amazon.

Luego, el Paso 5 en el Algoritmo 18 consiste en hacer la clasificación de los rendimientos usando los parámetros del Cuadro 5.6, observemos que los parámetros que necesitan ser usados deben transformarse, pues debido al último paso del Algoritmo 16, se calcula  $\hat{\mu}_j = \hat{\delta}_j + \frac{1}{2}\hat{\sigma}_j^2$ . Por lo tanto, los parámetros a usar para hacer la clasificación en el Algoritmo 17, no son los del Cuadro 5.6, sino que se deben calcular  $\hat{\delta}_j, j = 1, 2, 3$  y utilizarlas. Por lo tanto, los parámetros utilizados fueron los que se muestran en el Cuadro 5.8 para la acción de Netflix y el Cuadro 5.9 son los correspondientes a la acción de Amazon.

j	$\hat{\pi}_j$	$\hat{\delta}_j$	$\hat{\sigma}_j$
1	0.44797175	-0.00016378740	0.01402849
2	0.53779080	0.003635704	0.03213458
3	0.01423745	-0.0016089930	0.09737263

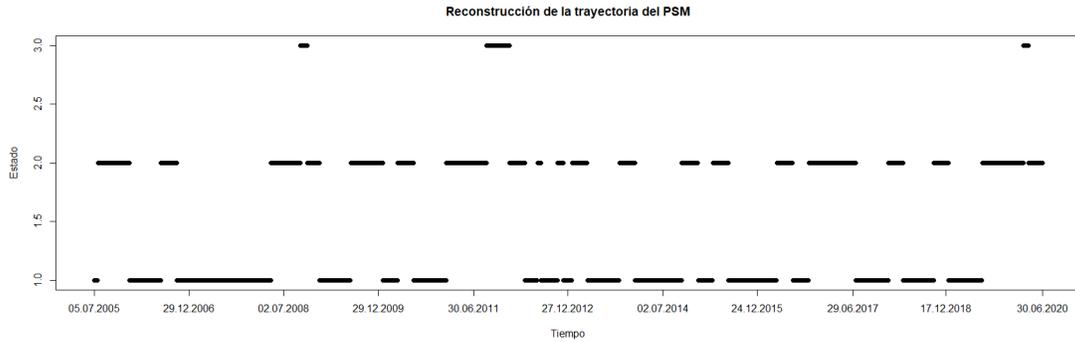
**Cuadro 5.8:** Parámetros utilizados para la clasificación de clusters de rendimientos de la acción de Netflix mediante el Algoritmo 17.

j	$\hat{\pi}_j$	$\hat{\delta}_j$	$\hat{\sigma}_j$
1	0.45787160	0.001017552	0.01093079
2	0.48193159	0.001447900	0.02213378
3	0.06019681	0.005320211	0.04834662

**Cuadro 5.9:** Parámetros utilizados para la clasificación de clusters de rendimientos de la acción de Amazon mediante el Algoritmo 17.

La reconstrucción de la trayectoria a tiempo discreto utilizando la clasificación obtenida en el Paso 5 para cada rendimiento observado de la acción de Netflix se muestra en la Figura 5.5 y para la acción de Amazon en la Figura 5.6.

De los resultados obtenidos en el Paso 5 para ambas acciones, podemos resaltar que ambas tienen poca estancia en el estado 3 de la economía que, recordemos representa un mal ambiente financiero para la acción. Esto tiene coherencia debido a que a lo largo de la historia de ambas acciones, han sido pocas las ocasiones que éstas se han



(a) Reconstrucción de la trayectoria del PSM a tiempo discreto para la acción de Netflix mediante la clasificación de los rendimientos sin salto y los resultados de la estimación de parámetros del Cuadro 5.8

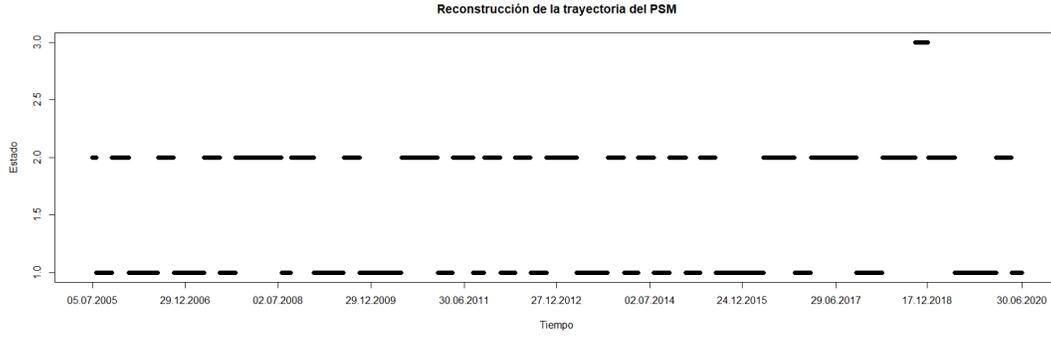


(b) Precios de la acción de Netflix (USD) del 1 de julio de 2005 al 30 de junio de 2020.

**Figura 5.5:** Comparación de la trayectoria del PSM a tiempo discreto con el precio de la acción de Netflix.

visto en caída. Además, las transiciones observadas en cada una de las trayectorias reconstruidas del PSM a tiempo discreto, muestran que en las ocasiones que ambas se han encontrado en el estado 3, siempre ha sido a través del estado 2 de la economía, esto significa que desde un punto de vista económico, la transición entre el ambiente financiero favorable (estado 1) y desfavorable (estado 2) tiene poca probabilidad pues sería un cambio muy brusco en la economía.

Con la clasificación lista, el Paso 6 es estimar el Generador del PSM mediante el Algoritmo 10 usando la reconstrucción de la trayectoria obtenida en el Paso 5. Con el fin de mantener las mismas condiciones de estimación para las trayectorias del PSM de ambas acciones, se tomó como estimación inicial  $Q^{(0)}$  el Generador del ejemplo en la sección 4.5 que puede consultarse en la ecuación (4.22), los resultados de la estimación para cada entrada en el caso de la acción de Netflix se muestran en la



(a) Reconstrucción de la trayectoria del PSM a tiempo discreto para la acción de Amazon mediante la clasificación de los rendimientos sin salto y los resultados de la estimación del Cuadro 5.7.



(b) Precios de la acción de Amazon (en USD) del 1 de julio de 2005 al 30 de junio de 2020.

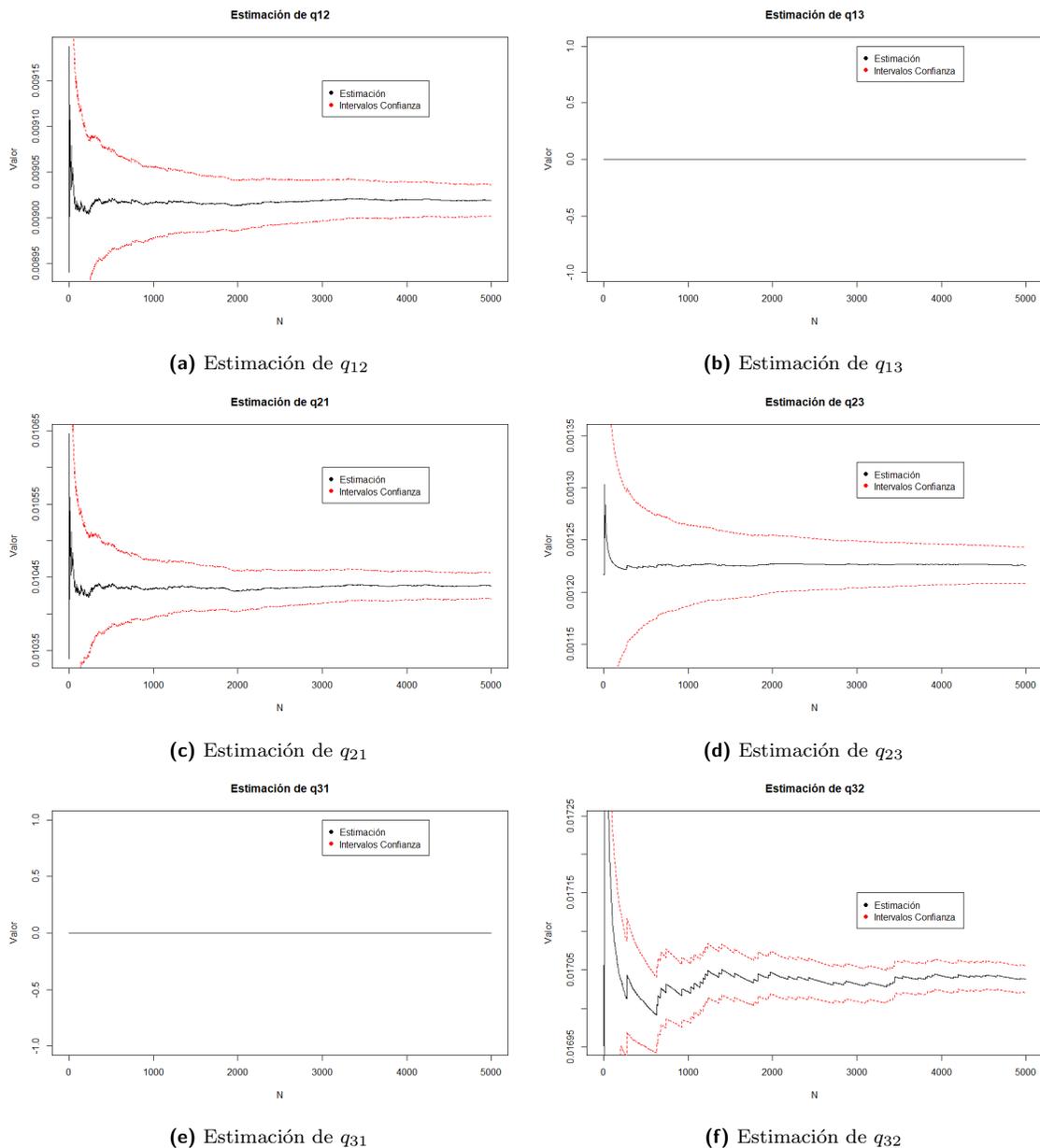
**Figura 5.6:** Comparación de la trayectoria del PSM a tiempo discreto con el precio de la acción de Amazon.

Figura 5.7 y el estimador obtenido se expone en la ecuación (5.1).

$$\hat{Q} = \begin{pmatrix} -0.009019071 & 0.009019071 & 0.000000000 \\ 0.010438460 & -0.011664262 & 0.001225801 \\ 0.000000000 & 0.017037859 & -0.017037859 \end{pmatrix}. \quad (5.1)$$

Para Amazon, los resultados de la estimación por entrada se muestran en la Figura 5.8 y el estimador del Generador Infinitesimal se muestran en la ecuación (5.2).

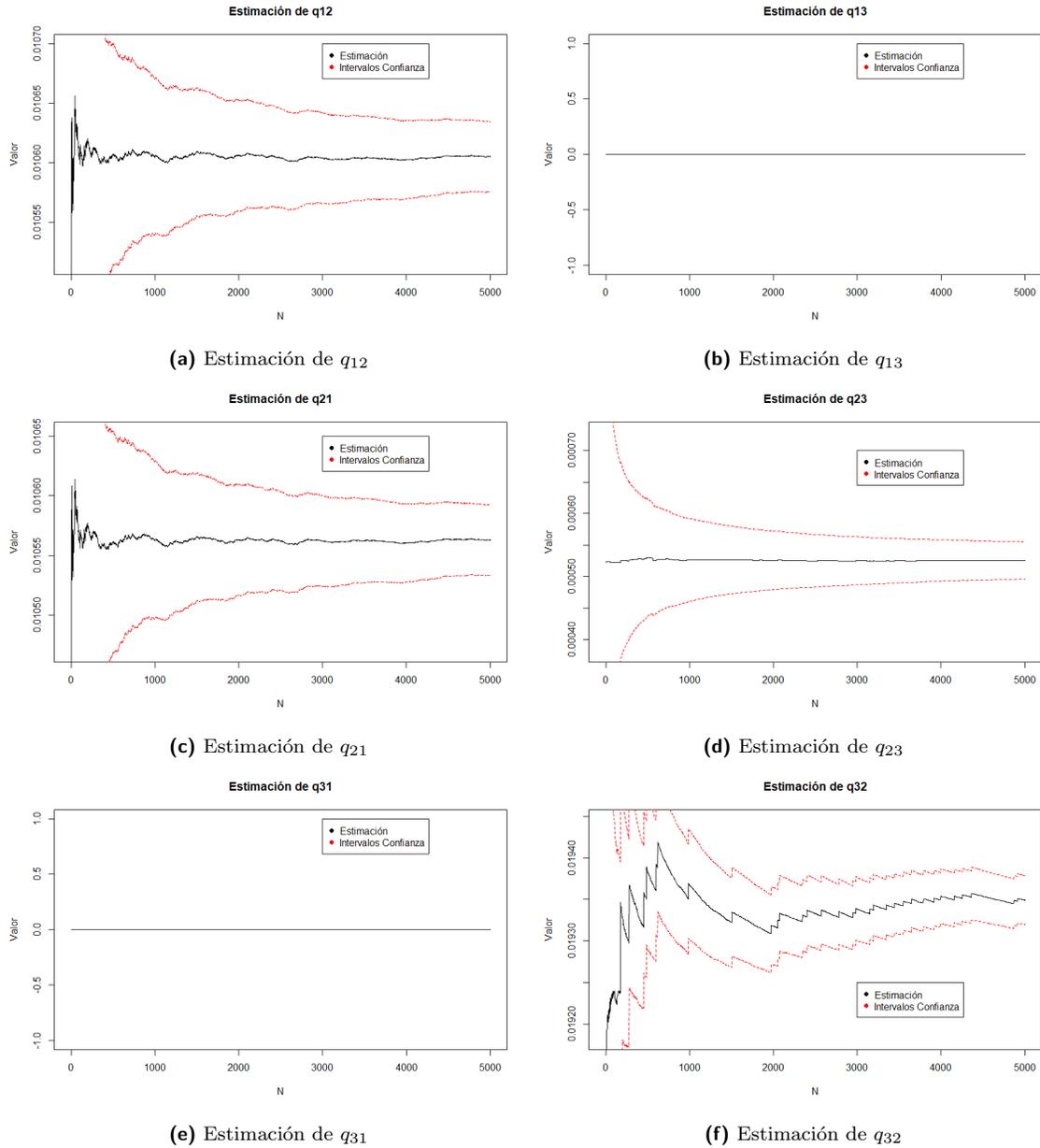
$$\hat{Q} = \begin{pmatrix} -0.01060526 & 0.01060526 & 0.000000000 \\ 0.01056306 & -0.01108866 & 0.0005255957 \\ 0.000000000 & 0.01934920 & -0.0193491966 \end{pmatrix}. \quad (5.2)$$



**Figura 5.7:** Estimación del Generador para la acción de Netflix e intervalos de confianza al 95 % (línea punteada).

## 5.4. Comparación del Generador Infinitesimal

En esta sección se analizarán los estimadores del Generador Infinitesimal bajo el Modelo de Difusión Markov-Modulado obtenidos en la sección 5.3 con el objetivo de validar el ajuste del Modelo de Difusión Markov-Modulado a las series de precios de las acciones de Netflix y Amazon (en USD).



**Figura 5.8:** Estimación del Generador para la acción de Amazon e intervalos de confianza al 95 % (línea punteada).

El estimador del Generador Infinitesimal para la acción de Netflix se muestra en la ecuación (5.3).

$$\hat{Q}_{NFLX} = \begin{pmatrix} -0.009019071 & 0.009019071 & 0.000000000 \\ 0.010438460 & -0.011664262 & 0.001225801 \\ 0.000000000 & 0.017037859 & -0.017037859 \end{pmatrix}. \quad (5.3)$$

El estimador del Generador Infinitesimal para la acción de Netflix puede verse en la ecuación (5.4).

$$\hat{Q}_{AMZN} = \begin{pmatrix} -0.01060526 & 0.01060526 & 0.0000000000 \\ 0.01056306 & -0.01108866 & 0.0005255957 \\ 0.00000000 & 0.01934920 & -0.0193491966 \end{pmatrix}. \quad (5.4)$$

Recordemos que la escala de tiempo está dada en días, esto será muy importante, pues lo primero que se hará será analizar la diagonal de los Generadores (5.3) y (5.4) que de acuerdo con las propiedades del Generador Infinitesimal vistas en el Capítulo 2, está relacionado con el tiempo de estancia en cada uno de los estados.

En el Generador  $Q_{NFLX}$  se puede observar que la v.a. Exponencial que modela el tiempo de estancia en el estado 1 de la economía, tiene parámetro  $\lambda = 0.009019071$ . Lo que por propiedades de la v.a. Exponencial, dicha economía se mantiene en promedio por 110.876177 días, por otro lado, en el caso de Amazon, al contar con un parámetro  $\lambda = 0.01060526$ , se tiene que el promedio de tiempo de estancia en ese estado es 94.292832 días. De hecho, tomando en cuenta los resultados obtenidos para el segundo estado de la economía en ambas empresas, se tiene que el promedio de estancia en dicho estado es 85.731971 y 90.1822222 días para Netflix y Amazon respectivamente. Finalmente, al considerar el tercer estado de la economía, se tiene que el tiempo medio de estancia para Netflix en dicho estado es de 58.69282 días, mientras que para Amazon es 51.68173 días. Podemos concluir entonces, que al comparar los tiempos de estancia, son consistentes entre ellos pues la media de estancia no difiere más allá de los 16 días.

Una similitud que es importante resaltar, es que tanto  $Q_{NFLX}$  como  $Q_{AMZN}$ , no presentan posibilidad alguna de saltar del estado 1 al estado 3, lo cual tiene coherencia con la realidad, pues las condiciones macroeconómicas es poco probable que sufran cambios tan bruscos.



# Capítulo 6

## Conclusiones

Recordemos que el propósito de esta tesis es proponer un modelo matemático, haciendo uso de las EDE con respecto al MB, con el objetivo de solucionar algunas de las debilidades que tiene el modelo propuesto por Black y Scholes para asemejar el movimiento de las acciones. El Modelo de Difusión Markov-Modulado con Saltos sirve para incorporar cambios en la dinámica al proceso propuesto por Kou el cual ya contiene un componente de saltos en la difusión para asemejar los cambios abruptos que se observan en los precios de las acciones.

Entre las dificultades que surgen al proponer una dinámica como la del Modelo de Difusión Markov-Modulado con Saltos, pueden mencionarse: el problema de elegir cuántas distribuciones componen una serie de observaciones de una variable aleatoria mixta, la determinación de una regla de decisión para aislar los rendimientos que contengan un salto y la implementación computacional de los algoritmos de estimación de parámetros del Modelo de Difusión Markov-Modulado con Saltos. Por otro lado, podemos resaltar que dichos algoritmos de estimación son lo suficientemente robustos pues como se mostró en el estudio de simulación realizado en el Capítulo 4, al conocer los parámetros reales de la trayectoria simulada, fue posible compararlos con las estimaciones realizadas con los algoritmos desarrollados en dicho capítulo y nos permitió llegar a la conclusión de que, con suficiente información los estimadores tienen una alta precisión.

En cuanto al ajuste hecho para las series de Amazon y Netflix, se puede concluir de los resultados obtenidos en el Capítulo 5 que el Modelo de Difusión Markov-Modulado con Saltos refleja adecuadamente el comportamiento de dichas acciones. Esto es porque durante el ajuste del Modelo de Difusión Markov-Modulado con Saltos a ambas series, se pudo observar coherencia en los estimadores obtenidos. Sin embargo, es importante destacar como una dificultad adicional, la necesidad de cambiar el método de identificación de saltos, pues como observamos a inicios del Capítulo 5, los datos reales presentan diferencias significativas con respecto al ejemplo planteado en

el Capítulo 4. Esto se debe, sobre todo, por el efecto que tienen los cambios de la economía en el comportamiento de las acciones.

Uno de los mayores beneficios que tiene el Modelo de Difusión Markov-Modulado con Saltos es poder hacer inferencia acerca de los ambientes financieros que favorecen a una acción, lo cual puede llegar a ser muy útil si se quiere especular acerca del valor de la acción a futuro.

Finalmente, el Modelo de Difusión Markov-Modulado con Saltos podría generalizarse aún más si en vez de considerar tiempos de estancia exponenciales (supuestos bajo el PSM) se consideran tiempos de estancia provenientes de una distribución Tipo Fase y partir del Modelo de Merton en lugar del Modelo de Kou, sin embargo el ajuste para una serie de datos real se volvería problemático, pues como se menciona en el Capítulo 4, la suficiencia de información es un factor muy limitante.

# Apéndice A

## Códigos

### A.1. Simulación de un PSM

El Código [A.1](#) simula un PSM estándar desde el tiempo 0 hasta un horizonte de tiempo dado.

**Parámetros de entrada:**

**Q:** Es un objeto tipo matrix que contiene el Generador Infinitesimal del PSM.

**V:** Es un vector con la distribución inicial del proceso.

**Ti:** Es un número positivo que indica el horizonte de tiempo de simulación del PSM.

**Salida:**

**Est:** Estados visitados (en orden) en la simulación.

**TiempSalto:** Tiempos en los que el proceso dio un salto.

**Código A.1:** Simulación de un PSM Estándar

```
Tray<-function(Q,V,Ti){
  t<-0 #Contara el tiempo que lleva el proceso
  Trayec<-c() #Contendra la trayectoria del proceso
  Saltos<-c(0) #Contiene los tiempos de salto
  x<-sample(c(1:length(V)),size=1,prob = V) #Estado inicial
  Trayec<-c(Trayec,x)
  t<-t+rexp(n=1,rate = -Q[x,x])
  Saltos<-c(Saltos,t)
  while (t<Ti) {
    V<-Q[x,]
    V[x]<-0
    V<- -V/Q[x,x]
    x<-sample(c(1:length(V)),size=1,prob = V) #Estado
    siguiente
  }
}
```

```

    Trayec<-c(Trayec , x)
    t<-t+rexp(1, -Q[x, x])
    Saltos<-c(Saltos , t)
  }
  Saltos [length(Saltos)]<-Ti
  return(list(Est=Trayec , TiempSalto=Saltos))
}

```

## A.2. Simulación de un PSM con Estado Absorbente

El Código A.2 simula una trayectoria de un PSM con estado absorbente hasta el momento de absorción.

### Parámetros de entrada:

**Q:** Es un objeto tipo matrix que contiene el Generador Infinitesimal del PSM.

**V:** Es un vector con la distribución inicial del proceso.

**E:** Es un entero que indica el estado al que debe llegar el proceso para detenerse.

### Salida:

**Est:** Estados visitados (en orden) durante la simulación.

**TiempSalto:** Tiempos en los que el proceso dio un salto.

### Código A.2: Simulación de un PSM con Estado Absorbente

```

Absorb<-function(Q,V,E){ #Simulacion hasta que se llegue al
  estado absorbente E
  t<-0 #Contara el tiempo que lleva el proceso
  Trayec<-c() #Contendra la trayectoria del proceso
  Saltos<-c(0) #Contiene los tiempos de salto
  x<-sample(c(1:length(V)),1,prob = V)
  Trayec<-c(Trayec , x)
  t<-t+rexp(1, -Q[x, x])
  Saltos<-c(Saltos , t)
  while (TRUE) { #Ciclo infinito
    V<-Q[x, ]
    V[x]<-0
    V<- -V/Q[x, x]
    x<-sample(c(1:length(V)),1,prob = V) #Estado siguiente
    Trayec<-c(Trayec , x)
    if (x==E) { #Si ya se llega al estado se regresa la
      trayectoria
      return(list(Est=Trayec , TiempSalto=Saltos))
    }
  }
}

```

```

}
else{ #Si no se ha llegado al estado
  TiE<-rexp(1,-Q[x,x])
  t<-t+TiE
  Saltos<-c(Saltos , t)
}
}
}
}

```

### A.3. Simulación de una Distribución Tipo Fase

El Código A.3 simula una realización de una distribución tipo fase.

**Parámetros de entrada:**

**Q:** Es un objeto tipo matrix que contiene el Generador Infinitesimal del PSM.

**V:** Es un vector con la distribución inicial del proceso.

**E:** Es un entero que indica el estado absorbente del proceso.

**Salida:**

**t:** Valor proveniente de una distribución tipo fase.

**Código A.3:** Simulación de una Distribución Tipo Fase

```

TipoFase<-function(Q,V,E){#Simulacion de un valor de una
  distribucion Tipo Fase
  t<-0 #Contara el tiempo que lleva el prceoso
  x<-sample(c(1:length(V)),1,prob = V)
  t<-t+rexp(1,-Q[x,x])
  while (True) {
    V<-Q[x,]
    V[x]<-0
    V<- -V/Q[x,x]
    x<-sample(c(1:length(V)),1,prob = V) #Estado siguiente
    if (x==E) { #Si ya se llega al estado se regresa el
      tiempo
      return(t)
    }
    else{ #Si no se ha llegado al estado
      TiE<-rexp(1,-Q[x,x])
      t<-t+TiE #Se le suma el tiempo de estancia al total
    }
  }
}
}
}

```

## A.4. Estimación del tiempo medio de absorción vía Monte Carlo

El Código A.4 estima el tiempo medio de absorción de un PSM con un estado absorbente vía el método Monte Carlo crudo.

### Parámetros de entrada:

**Q:** Es un objeto tipo matrix que contiene el Generador Infinitesimal del PSM.

**V:** Es un vector con la distribución inicial del proceso.

**E:** Es un entero que indica el estado absorbente del proceso.

**n:** Es un entero que fija el número de simulaciones que se realizarán para la estimación.

### Salida:

**Media:** Devuelve la media muestral de los tiempos de absorción observados en las simulaciones.

#### Código A.4: Estimación del Tiempo Medio de Absorción Mediante Monte Carlo

```
MCAbsorb<-function(Q,V,E,n){ #Estimacion del tiempo medio de
  absorcion
  Obs<-c() #Contiene las observaciones
  for(i in 1:n){
    Obs[i]<-TipoFase(Q,V,E) #Simulacion
  }
  return(mean(Obs)) #Media
}
```

## A.5. Momentos de una distribución Tipo Fase

El Código A.5 calcula de forma analítica el m-ésimo momento de una distribución tipo fase.

### Parámetros de entrada:

**Q:** Es un objeto tipo matrix que contiene el Generador Infinitesimal del PSM.

**V:** Es un vector con la distribución inicial del proceso.

**m:** Es un entero que indica el m-ésimo momento que se calculará.

### Salida:

**Momento:** Valor del m-ésimo momento de la distribución tipo fase.

#### Código A.5: Cálculo del m-ésimo Momento Analítico de una Distribución Tipo Fase

```
EspFase<-function(Q,V,m){ #m-esimo momento (analitico) de la
  distribucion tipo fase
  V<-matrix(V)
  SubInt<-Q[,-length(Q[1,])] ]
```

## A.6. ESTIMACIÓN DEL GENERADOR INFINITESIMAL DE UN PSM CON INFORMACIÓN COM

```
SubInt<-SubInt[-length(Q[,1]),]  
InvSubInt<-solve(SubInt)  
Potencia<-InvSubInt %*% m  
e<-matrix(1,length(Potencia[1,]),1)  
Mult<-factorial(m)*((-1)^m)*t(V) %*% Potencia %*% e #Calculo  
return(Mult[1,1]) #Solucion  
}
```

### A.6. Estimación del Generador Infinitesimal de un PSM con Información Completa

El Código A.6 obtiene el EMV del generador infinitesimal de un PSM con información completa.

#### Parámetros de entrada:

**Trayectoria:** Es un vector que contiene los estados visitados por el proceso (en orden).

**Tiem:** Es un vector que contiene los tiempos de salto del proceso.

**EspacioEstados:** Es un vector que contiene el espacio de estados del proceso.

#### Salida:

**Gen:** EMV del generador infinitesimal del proceso.

**Verosimilitud:** Valor máximo de la verosimilitud alcanzado con el EMV.

#### Código A.6: Cálculo del EMV de un PSM Estándar con Información Completa

```
Estados<-function(Trayectoria){ #La funcion recibe una  
  trayectoria y devuelve los estados visitados  
  #V es un vector que contiene la trayectoria del proceso  
  Est<-c() #Vector vacio  
  for (i in 1:length(Trayectoria)) { #Recorremos la  
    trayectoria  
    if(!(Trayectoria[i] %in% Est)){  
      Est<-c(Est, Trayectoria[i])  
    }  
  }  
  return(sort(Est)) #Regesamos los estados observados (  
    ordenados)  
}  
  
TiEst<-function(Trayectoria, Tiem, EspacioEstados){ #La funcion  
  sirve para regresar los tiempos de estancia en los estdos  
  #a partir de los tiempos de salto.  
  TE<-diff(Tiem) #Tiempos estancia
```

```

Ri<-EspacioEstados*0
for (j in 1:length(TE)) {
  Actual<-Trayectoria[j]
  Ri[Actual]<-Ri[Actual]+TE[j]
}
return(Ri)
}

Salto<-function(Act, Dest, Trayectoria) {#La funcion nos
  devuelve el numero de saltos de un estado a otro
  #Act es el estado desde el cual debe saltar el proceso
  #Dest es el estado de destino hacia el cual saltara el
  proceso
  #Trayectoria es un vector con la trayectoria del proceso
  Num<-0
  for (i in 1:(length(Trayectoria)-1)) {
    if (Trayectoria[i]==Act & Trayectoria[i+1]==Dest) {#
      Contamos saltos de Act a Dest
      Num<-Num+1
    }
  }
  return(Num)
}

Estimacion<-function(Trayectoria, Tiem, EspacioEstados) { #
  Estima el generador Q

  m<-length(EspacioEstados)
  Q<-diag(0, m, m)
  R<-TiEst(Trayectoria, Tiem, EspacioEstados) #Obtenemos los
  tiempos de estancia en cada estado
  N<-diag(0, m, m)
  Verosim<-1
  for (i in 1:m) { #Estimamos el generador
    for (j in 1:m) {
      if (!(i==j)) { #La diagonal se estima al final
        N[i, j]<-Salto(i, j, Trayectoria)
        Q[i, j]<-N[i, j]/R[i]
        Verosim<-Verosim*(Q[i, j]^N[i, j])*exp(-Q[i, j]*R[i])
      }
    }
  }
  Q[i, i]<-(-1)*sum(Q[i, ]) #Usamos las propiedades del

```

```

    generador para estimar la diagonal
  }
  return( list( Gen=Q, Verosimilitud=Verosim))
}

```

## A.7. Método Auxiliar de Puentes Markovianos

El Código [A.7](#) es un programa auxiliar que sirve para eliminar saltos “fantasma” que se generan al construir un Puente Markoviano por cualquier método.

### Parámetros de entrada:

**Trayectoria:** Es un vector que contiene los estados visitados por el proceso (en orden).

**Tiempos:** Es un vector que contiene los tiempos de salto del proceso.

### Salida:

**Est:** Es un vector que contiene los estados visitados por el proceso (en orden y sin saltos “fantasma”)

**Tiempos:** Es un vector que contiene los tiempos de salto del proceso.

### Código A.7: Método Auxiliar para Eliminar Saltos Incongruentes en las Trayectorias Completadas

```

Depurar<-function( Trayectoria ,Tiempos){#La funcion elimina
  saltos fantasma en las trayectoriasreconstruidas por
  cualquiera de los algoritmos y les da formato para
  graficarlos directamente

  TrayDep<-c() #Contendra la trayectoria Depurada
  TiemDep<-c() #Contendra los tiempos de salto Depurados
  i<-1 #Iniciamos en la primera entrada
  while (i<=length(Trayectoria)) {
    Actual<-Trayectoria[i]
    TrayDep<-c(TrayDep, Actual)
    TiemDep<-c(TiemDep, Tiempos[i])
    j<-0 #Hacemos un contador para eliminar repetidos
    while (Actual==Trayectoria[i+j]) { #Mientras el estado
      actual se repita en los consecutivos
      j<-j+1
      if (i+j>length(Trayectoria)){ #Esto es para no salirnos
        del vector
        break
      }
    }
  }
}

```

```

    i<-i+j #Avanzamos hasta que deje de repetirse el estado
          actual
  }
  return ( list ( Est=c ( TrayDep [ 1 ] , TrayDep ) , Tiempos=TiempDep ) )
}

```

## A.8. Método de la Bisección

El Código A.8 aplica el Método de la Bisección a un conjunto de observaciones discretas de un PSM estándar.

### Parámetros de entrada:

**Trayectoria:** Es un vector que contiene observaciones discretas de la trayectoria de un PSM.

**Tiempos:** Es un vector que contiene los tiempos de observación de la trayectoria del PSM.

**Q:** Una matriz que contiene el generador infinitesimal del PSM.

### Salida:

**Est:** Estados visitados por el proceso (en orden) en la trayectoria reconstruida

**Tiempos:** Tiempos de salto del proceso en la trayectoria reconstruida.

**Código A.8:** Método de la Bisección

```

SemGT<-function (Q, t) { #Devuelve el semigrupo de transicion a
  partir del generador
  return (expm(Q*t))
}

Pab<-function (a, b, t, Q) { #Funcion auxiliar para devolver la
  entrada del semigrupo
  P<-SemGT(Q, t)
  return (P[a, b])
}

Rab<-function (a, b, t, Q) {#Funcion auxiliar que nos ayuda a
  calcular la proba de un salto
  if (Q[a, a]==Q[b, b]) {
    return (Q[a, b] *t*exp(Q[a, a] *t))
  }
  else {
    return (Q[a, b] * (exp(Q[a, a] *t) -exp(Q[b, b] *t)) / (-Q[b, b]+Q[a,
    a]))
  }
}

```

```

}

ea<-function(a,t,Q){#Funcion auxiliar para calcular la proba
  de estancia en un estado
  qa<-(-1)*Q[a,a]
  return(exp(-qa*t))
}

SimSalto<-function(a,b,ta,tb,Q){#Simula un tiempo de salto de
  'a' a 'b' entre ta y tb (ta<tb)
  if (a==b) {
    return("Mismo Estado")
  }
  qa<-(-1)*Q[a,a] #Obtenemos los parametros
  qb<-(-1)*Q[b,b]
  t<-abs(tb-ta) #Obtenemos la longitud del intervalo
  U<-runif(1)
  if(qa>qb){ #Caso 1
    return(ta+(-1/(qa-qb))*log(1-U*(1-exp(-t*(qa-qb)))))
  }
  if(qb>qa){ #Caso 2
    return(tb-(-1/(qb-qa))*log(1-U*(1-exp(-t*(qb-qa)))))
  }
  else{
    return(ta+(U*t))
  }
}

OrdenarBis<-function(Trayectoria,Tiempos){ #La funcion ordena
  las trayectorias devueltas por el metodo de la Biseccion
  debido a que se ordenan por pares pero no necesariamente
  en orden
  if (sum(duplicated(Tiempos))>0) {
    return("Duplicados")
  }
  else{
    TiemposOrd<-sort(Tiempos)
    TrayecOrd<-c()
    for (Tiempo in TiemposOrd) {
      Indice<-match(Tiempo,Tiempos) #Como estan pareados
      tiempos y estados, encontramos el indice del estado
      mediante el indice del tiempo desordenado
    }
  }
}

```

```

    TrayecOrd<-c(TrayecOrd, Trayectoria [Indice])
  }
  return(list(Est=TrayecOrd, Tiempos=TiempesOrd))
}
}
Bisec<-function(x1,x2,t1,t2, Trayectoria, Tiempes,Q, Condicion){
  #Aplica el algoritmo de la biseccion al intervalo [t1,t2]
  con estados x1 y x2 de inicio y final respectivamente
  NumEst<-length(Q[1,])
  t<-abs(t2-t1)/2
  if(x1==x2){ #Si el estado de inicio es el mismo que el de
    final
    Probs<-c(ea(x1,t*2,Q),ea(x1,t,Q)*(Pab(x1,x1,t,Q)-ea(x1,t,
      Q)),ea(x1,t,Q)*(Pab(x1,x1,t,Q)-ea(x1,t,Q)),(Pab(x1,x1,
      t,Q)-ea(x1,t,Q))^2)
    Casos<-c(1:4) #Contendra los casos posibles para saltos
    Otros<-c(1:NumEst)[-x1] #Obtenemos los posibles estado de
      salto
    for (i in Otros) { #Llenamos los vectores con los casos y
      las probas de cada caso
      Probs<-c(Probs,Rab(x1,i,t,Q)*Rab(i,x1,t,Q),Rab(x1,i,t,Q)
        )*(Pab(i,x1,t,Q)-Rab(i,x1,t,Q)),Rab(i,x1,t,Q)*(Pab(
          x1,i,t,Q)-Rab(x1,i,t,Q)),(Pab(x1,i,t,Q)-Rab(x1,i,t,Q)
          ))*(Pab(i,x1,t,Q)-Rab(i,x1,t,Q))
      Casos<-c(Casos,(Casos[length(Casos)]+1):(Casos[length(
        Casos)]+4))
    }
    if(Condicion==2){ #Si queremos condicionar a al menos dos
      saltos
      Probs[1]<-0 #Eliminamos la posibilidad de obtener el
        caso 1
    }
    Eleccion<-sample(Casos, size=1,prob = Probs) #Obtenemos la
      decision
    if(Eleccion<=4){
      if(Eleccion==1){ #Esto es simplemente unir puntos
        return(list(Traye=Trayectoria, Tiempes=Tiempes)) #
          Regreso lo que ya tenia, no es necesario agregar
          nada
      }
      if(Eleccion==2){ #El primer intervalo es constante y al

```

```

        segundo se le aplica de nuevo
        #Trayectoria<-c(Trayectoria ,x1)
        #Tiempos<-c(Tiempos ,t1+t)
        return(Bisec(x1 ,x2 ,t1+t ,t2 ,Trayectoria ,Tiempos ,Q,2))
    }
    if(Eleccion==3){ #El segundo intervalo es constante y
        al primero se le aplica de nuevo
        #Trayectoria<-c(Trayectoria ,x2)
        #Tiempos<-c(Tiempos ,t1+t)
        return(Bisec(x1 ,x2 ,t1 ,t1+t ,Trayectoria ,Tiempos ,Q,2))
    }
    if(Eleccion==4){ #Se le aplica de nuevo a ambos
        intervalos
        PrimeraMitad<-Bisec(x1 ,x2 ,t1 ,t1+t ,Trayectoria ,Tiempos
            ,Q,2)
        return(Bisec(x1 ,x2 ,t1+t ,t2 ,PrimeraMitad$Traye ,
            PrimeraMitad$Tiempos ,Q,2))
    }
}
else{#Si la eleccion no fue menor a 5 entonces debemos
    considerar los casos adicionales
    #por cada estado que se tenga en el proceso y que sea
    distinto del actual
    IndiceEstado<-floor((Eleccion-1)/4) #Obtenemos el
        indice del estado de salto
    Estado<-Otros[IndiceEstado] #Obtenemos el estado de
        salto
    Subcaso<-((Eleccion-1)%4)+1 #Obtenemos el caso elegido
    if(Subcaso==1){#Salta al nuevo estado en la primera
        mitad y regresa en la segunda
        PrimerSalto<-SimSalto(x1 ,Estado ,t1 ,t1+t ,Q) #Salto del
            actual al nuevo
        SegundoSalto<-SimSalto(Estado ,x2 ,t1+t ,t2 ,Q) #Salto
            del nuevo de regreso al actual
        Trayectoria<-c(Trayectoria ,Estado ,x2) #Agregamos los
            estados (cont por la derecha)
        Tiempos<-c(Tiempos ,PrimerSalto ,SegundoSalto) #
            Agregamos tiempos de salto
        Ordenado<-OrdenarBis(Trayectoria ,Tiempos) #Ordenamos
            los saltos y estados
        return(list(Traye=Ordenado$Est ,Tiempos=Ordenado$
            Tiempos))
    }
}

```

```

}
if (Subcaso==2){#Salta al nuevo estado en la primera
  mitad y salta dos o mas veces en la segunda
  PrimerSalto<-SimSalto(x1,Estado,t1,t1+t,Q) #Salto del
    actual al nuevo
  Trayectoria<-c(Trayectoria,Estado) #Agregamos el
    estado al cual salta
  Tiempos<-c(Tiempos,PrimerSalto) #Agreganis el tiempo
    de salto
  Ordenado<-OrdenarBis(Trayectoria,Tiempos) #Ordenamos
    los saltos y estados
  return(Bisec(Estado,x2,t1+t,t2,Ordenado$Est,Ordenado$
    Tiempos,Q,2))
}
if (Subcaso==3){#Salta dos o mas veces primero y despues
  salta una vez para regresar
  SegundoSalto<-SimSalto(Estado,x2,t1+t,t2,Q) #Salto
    del nuevo de regreso al actual
  Trayectoria<-c(Trayectoria,x2) #Agregamos el estado (
    cont por la derecha)
  Tiempos<-c(Tiempos,SegundoSalto) #Agregamo el tiempo
    de salto
  Ordenado<-OrdenarBis(Trayectoria,Tiempos)
  return(Bisec(x1,Estado,t1,t1+t,Ordenado$Est,Ordenado$
    Tiempos,Q,2))
}
if (Subcaso==4){#Salta dos o mas veces en ambos
  intervalos pasando por Estado
  PrimeraMitad<-Bisec(x1,Estado,t1,t1+t,Trayectoria,
    Tiempos,Q,2)
  return(Bisec(Estado,x2,t1+t,t2,PrimeraMitad$Traye,
    PrimeraMitad$Tiempos,Q,2))
}
}
}
else{#Si el estado de inicio no coincide con el de el final
  Probs<-c(ea(x1,t,Q)*Rab(x1,x2,t,Q),ea(x1,t,Q)*(Pab(x1,x2,
    t,Q)-Rab(x1,x2,t,Q)),(Pab(x1,x1,t,Q)-ea(x1,t,Q))*Rab(
    x1,x2,t,Q),(Pab(x1,x1,t,Q)-ea(x1,t,Q))*(Pab(x1,x2,t,Q)
    -Rab(x1,x2,t,Q)))
  Probs<-c(Probs,Rab(x1,x2,t,Q)*ea(x2,t,Q),Rab(x1,x2,t,Q)*
    (Pab(x2,x2,t,Q)-ea(x2,t,Q)),(Pab(x1,x2,t,Q)-Rab(x1,x2,t,

```

```

    ,Q)) * ea(x2, t, Q), (Pab(x1, x2, t, Q) - Rab(x1, x2, t, Q)) * (Pab(
    x2, x2, t, Q) - ea(x2, t, Q)))
Casos<-c(1:8) #Contendra los casos posibles para saltos
Otros<-c(1:NumEst)[-c(x1, x2)] #Eliminamos estado inicial
y final
for (i in Otros) { #Llenamos los vectores con los casos y
las probas de cada caso
  Probs<-c(Probs, Rab(x1, i, t, Q) * Rab(i, x2, t, Q), Rab(x1, i, t, Q)
) * (Pab(i, x2, t, Q) - Rab(i, x2, t, Q)), (Pab(x1, i, t, Q) - Rab(
x1, i, t, Q)) * Rab(i, x2, t, Q), (Pab(x1, i, t, Q) - Rab(x1, i, t, Q)
)) * (Pab(i, x2, t, Q) - Rab(i, x2, t, Q)))
  Casos<-c(Casos, (Casos[length(Casos)]+1):(Casos[length(
Casos)]+4))
}
if (Condicion==2){ #Si queremos condicionar a al menos dos
saltos
  Probs[1]<-0 #Eliminamos posibilidades de elegir los
casos 1 y 5
  Probs[5]<-0
}
Eleccion<-sample(Casos, size=1, prob = Probs)
if (Eleccion <=8){
  if (Eleccion==1){ #Constante en la primera mitad y salto
en la segunda mitad
    Salto<-SimSalto(x1, x2, t1+t, t2, Q) #Simulamos el salto
en la segunda mitad
    Trayectoria<-c(Trayectoria, x2) #Actualizamos la
trayectoria y los tiempos
    Tiempos<-c(Tiempos, Salto)
    Ordenado<-OrdenarBis(Trayectoria, Tiempos)
    return(list(Traye=Ordenado$Est, Tiempos=Ordenado$
Tiempos))
  }
  if (Eleccion==2){ #Constante en la primera mitad y dos o
mas saltos en la otra mitad
    #Trayectoria<-c(Trayectoria, x1) #Creo que esto no es
necesario volvere despues
    #Tiempos<-c(Tiempos, t1+t)
    return(Bisec(x1, x2, t1+t, t2, Trayectoria, Tiempos, Q, 2))
  }
  if (Eleccion==3){#Salto en la segunda mitad y dos o mas
saltos en la primer mitad

```

```

Salto<-SimSalto(x1,x2,t1+t,t2,Q) #Simulamos el salto
en la segunda mitad
Trayectoria<-c(Trayectoria,x2) #Actualizamos la
trayectoria y los tiempos
Tiempos<-c(Tiempos,Salto)
Ordenado<-OrdenarBis(Trayectoria,Tiempos)
return(Bisec(x1,x1,t1,t1+t,Ordenado$Est,Ordenado$
Tiempos,Q,2))
}
if(Eleccion==4){ #Se le aplica de nuevo a ambas mitades
del intervalo, pasa por x1
PrimeraMitad<-Bisec(x1,x1,t1,t1+t,Trayectoria,Tiempos
,Q,2)
return(Bisec(x1,x2,t1+t,t2,PrimeraMitad$Traye,
PrimeraMitad$Tiempos,Q,2))
}
if(Eleccion==5){ #Salto en la primera mitad y constante
en la segunda
Salto<-SimSalto(x1,x2,t1,t1+t,Q) #Simulamos el salto
en la primera mitad
Trayectoria<-c(Trayectoria,x2)
Tiempos<-c(Tiempos,Salto)
Ordenado<-OrdenarBis(Trayectoria,Tiempos)
return(list(Traye=Ordenado$Est,Tiempos=Ordenado$
Tiempos))
}
if(Eleccion==6){#Salto en la primera mitad y dos o mas
saltos en la segunda mitad
Salto<-SimSalto(x1,x2,t1,t1+t,Q) #Simulamos el salto
en la primera mitad
Trayectoria<-c(Trayectoria,x2)
Tiempos<-c(Tiempos,Salto)
Ordenado<-OrdenarBis(Trayectoria,Tiempos)
return(Bisec(x2,x2,t1+t,t2,Ordenado$Est,Ordenado$
Tiempos,Q,2))
}
if(Eleccion==7){ #Dos o mas saltos en la primera mitad
y constante en la segunda
#Trayectoria<-c(Trayectoria,x2) #Creo que esto no es
necesario volvere mas tarde
#Tiempos<-c(Tiempos,t1+t)
return(Bisec(x1,x2,t1,t1+t,Trayectoria,Tiempos,Q,2))
}

```

```

}
if (Eleccion==8){#Se le aplica de nuevo a ambas mitades
  del intervalo , pasa por x2
  PrimeraMitad<-Bisec(x1,x2,t1,t1+t, Trayectoria , Tiempos
    ,Q,2)
  return ( Bisec(x2,x2,t1+t,t2,PrimeraMitad$Traye,
    PrimeraMitad$Tiempos,Q,2) )
}
}
else{#Si la eleccion no fue menor a 9 entonces debemos
  considerar los casos adicionales
  #por cada estado que se tenga en el proceso distinto a
  x1 y x2
  IndiceEstado<-floor((Eleccion-5)/4) #Obtenemos el
  indice del estado de salto
  Estado<-Otros[IndiceEstado] #Obtenemos el estado de
  salto
  Subcaso<-((Eleccion-1)%4)+1 #Obtenemos el caso elegido
  if (Subcaso==1){ #Salta de x1 a un estado elegido y
    despues salta a x2
    PrimerSalto<-SimSalto(x1, Estado, t1, t1+t, Q) #Tiempo de
    salto desde x1
    SegundoSalto<-SimSalto(Estado, x2, t1+t, t2, Q) #Tiempo
    de salto hacia x2
    Trayectoria<-c(Trayectoria, Estado, x2)
    Tiempos<-c(Tiempos, PrimerSalto, SegundoSalto)
    Ordenado<-OrdenarBis(Trayectoria, Tiempos)
    return(list(Traye=Ordenado$Est, Tiempos=Ordenado$
      Tiempos))
  }
  if (Subcaso==2){ #Salta de x1 a un estado elegido y
    reaplicamos para la otra mitad
    Salto<-SimSalto(x1, Estado, t1, t1+t, Q)
    Trayectoria<-c(Trayectoria, Estado)
    Tiempos<-c(Tiempos, Salto)
    Ordenado<-OrdenarBis(Trayectoria, Tiempos)
    return(Bisec(Estado, x2, t1+t, t2, Ordenado$Est, Ordenado$
      Tiempos, Q, 2) )
  }
  if (Subcaso==3){ #Salta de un estado elegido a x2 y
    reaplicamos para la otra mitad
    Salto<-SimSalto(Estado, x2, t1+t, t2, Q)

```

```

    Trayectoria<-c(Trayectoria ,x2)
    Tiempos<-c(Tiempos ,Salto)
    Ordenado<-OrdenarBis(Trayectoria ,Tiempos)
    return(Bisec(x1 ,Estado ,t1 ,t1+t ,Ordenado$Est ,Ordenado$
      Tiempos ,Q,2) )
  }
  if(Subcaso==4){ #Reaplicamos para ambas partes , pasa
    por el estado elegido
    PrimeraMitad<-Bisec(x1 ,Estado ,t1 ,t1+t ,Trayectoria ,
      Tiempos ,Q,2)
    return(Bisec(Estado ,x2 ,t1+t ,t2 ,PrimeraMitad$Traye ,
      PrimeraMitad$Tiempos ,Q,2) )
  }
}
}
}
}

Reconstruccion<-function(Trayectoria ,Tiempos ,Q){ #La funcion
  reconstruye la trayectoria a tiempo continuo mediante
  observaciones a tiempo discreto
  ModificacionTray<-c() #Contendra la trayectoria con el
    algoritmo aplicado
  ModificacionTime<-c() #Contendra los tiempos con el
    algoritmo aplicadp
  for(i in 1:(length(Trayectoria)-1)){ #Aplicamos el
    algoritmo a cada subintervalo
    Completado<-Bisec(Trayectoria[i] ,Trayectoria[i+1] ,Tiempos
      [i] ,Tiempos[i+1] ,c(Trayectoria[i] ,Trayectoria[i+1]) ,c(
        Tiempos[i] ,Tiempos[i+1]) ,Q,1)
    ModificacionTray<-c(ModificacionTray ,Completado$Traye[-
      length(Completado$Traye)])
    ModificacionTime<-c(ModificacionTime ,Completado$Tiempos[-
      length(Completado$Tiempos)])
  }
  ModificacionTray<-c(ModificacionTray ,Trayectoria[length(
    Trayectoria)])
  ModificacionTime<-c(ModificacionTime ,Tiempos[length(Tiempos
    )])
  return(list(Est=ModificacionTray ,Tiempos=ModificacionTime))
}

```

## A.9. Método de Aceptación-Rechazo

El Código A.9 reconstruye una trayectoria de un PSM observada a tiempo discreto.

### Parámetros de entrada:

**Trayectoria:** Es un vector que contiene los estados visitados por el proceso (en orden) en la observación a tiempo discreto.

**Tiempos:** Es un vector que contiene los tiempos de observación del PSM.

**Q:** Matriz que contiene el generador infinitesimal del PSM.

### Salida:

**Est:** Contiene la trayectoria reconstruida del PSM (en orden)

**Tiempos:** Contiene los tiempos de salto del PSM en la trayectoria reconstruida.

### Código A.9: Método de Aceptación-Rechazo

```
AR<-function (Trayectoria , Tiempos ,Q) {#Completa una obs por el
  metodo de aceptacion rechazo
  ModificacionTray<-c() #Contendra la trayectoria con el
    algoritmo aplicado
  ModificacionTime<-c() #Contendra los tiempos con el
    algoritmo aplicado
  i<-1
  while (i<=(length(Trayectoria)-1)){ #Aplicamos el algoritmo
    a cada intervalo
    Inicial<-Trayectoria[i] #Obtenemos el extremo izquierdo
      del intervalo
    Final<-Trayectoria[i+1] #Obtenemos el extremo derecho del
      intervalo
    Longitud<-abs(Tiempos[i+1]-Tiempos[i]) #Obtenemos la
      longitud del intervalo
    DI<-c(1:length(Q[1,]))*0 #Esto servira para poner la
      distribucion inicial
    DI[Inicial]<-1
    Simulacion<-Tray(Q,DI,Longitud) #Simulo una trayectoria
      de la misma longitud
    EstadosSimulados<-Simulacion$Est[-length(Simulacion$Est)]
    TiemposSimulados<-Simulacion$Time+Tiempos[i]
    TiemposSimulados<-TiemposSimulados[-length(
      TiemposSimulados)]
    if (EstadosSimulados[length(EstadosSimulados)]==Final) {#
      Condicion de aceptacion
      i<-i+1 #Solamente si se cumple la condicion de
        aceptacion: avanzamos
      ModificacionTray<-c(ModificacionTray , EstadosSimulados)
```

```

        ModificacionTime<-c( ModificacionTime , TiemposSimulados)
    }
}
return( list( Est=c( ModificacionTray , Trayectoria [ length(
    Trayectoria) ] ) , Tiempos=c( ModificacionTime , Tiempos [ length
    (Tiempos) ] ) ) )
}

```

## A.10. Método de Uniformización

El Código A.10 reconstruye la trayectoria de un PSM observada a tiempo discreto.

### Parámetros de entrada:

**Trayectoria:** Es un vector que contiene los estados visitados por el proceso (en orden) en la observación a tiempo discreto.

**Tiempos:** Es un vector que contiene los tiempos de observación del PSM.

**Q:** Matriz que contiene el generador infinitesimal del PSM.

### Salida:

**Est:** Contiene la trayectoria reconstruida del proceso.

**Tiempos:** Contiene los tiempos de salto de la trayectoria reconstruida.

### Código A.10: Método de Uniformización

```

SaltosUnif<-function(x1,x2,t1,t2,Q){ #Simula los saltos de un
    estado a otro para el algoritmo de Uniformizacion
    U<-runif(1)
    i<-0 #Numero de saltos
    nu<-0 #El maximo de los elementos de la diagonal del
        generador
    Delta<-(t2-t1)
    for (j in 1:length(Q[1,])) { #Obtenemos el maximo de los
        elementos de la diag de Q
        if(nu<(-1)*Q[j,j]){
            nu<-(-1)*Q[j,j]
        }
    }
    Sigma<-diag(1,length(Q[1,]))+(Q/nu) #Calculamos Sigma
    p<-exp(-nu*Delta)*diag(1,length(Q[1,])) [x1,x2]/SemGT(Q,
        Delta)[x1,x2] #Probabilidad inicial (Cero saltos)
    S<-p
    while (S<U) { #Mientras la distribucion no rebase el valor
        de la Uniforme
        i<-i+1 #Aumentamos el salto
    }
}

```

```

    p<-exp(-nu*Delta)*((nu*Delta)^i)*((Sigma %>% a)[x1,x2])/
      factorial(i)*(SemGT(Q,Delta)[x1,x2])
    S<-S+p
  }
  return(i)
}

Uniformizacion<-function(Trayectoria,Tiempos,Q){
  ModificacionTray<-c(Trayectoria[1])
  ModificacionTime<-c(0)
  i<-1
  nu<-0 #El maximo de los elementos de la diagonal del
    generador
  for(j in 1:length(Q[1,])){ #Obtenemos el maximo de los
    elementos de la diag de Q
    if(nu<(-1)*Q[j,j]){
      nu<-(-1)*Q[j,j]
    }
  }
  Sigma<-diag(1,length(Q[1,]))+(Q/nu) #Calculamos Sigma
  while(i<=(length(Trayectoria)-1)){ #Aplicamos el algoritmo
    a cada intervalo
    EInicial<-Trayectoria[i] #Obtenemos el extremo izquierdo
      del intervalo
    EFinal<-Trayectoria[i+1] #Obtenemos el extremo derecho
      del intervalo
    TInicial<-Tiempos[i] #Extremo izquierdo del intervalo
    TFinal<-Tiempos[i+1] #Extremo derecho del intvervalo
    SaltosSim<-SaltosUnif(EInicial,EFinal,TInicial,TFinal,Q)
      #Numero de saltos en el intervalo
    if(SaltosSim==0){ #En caso de que no haya saltos
      ModificacionTray<-c(ModificacionTray,Trayectoria[i+1])
      ModificacionTime<-c(ModificacionTime,Tiempos[i+1])
      i<-i+1
    }
    else{ #En caso de que haya saltos
      TiemposDeSalto<-sort(runif(SaltosSim,TInicial,TFinal))
        #Genero los tiempos de salto
      Actual<-EInicial #Guardo el estado actual (extremo
        izquierdo del intervalo)
      for(k in 1:SaltosSim){ #Generamos la trayectoria en
        los puntos de salto

```

```

    if(k==SaltosSim){
      ModificacionTray<-c(ModificacionTray , EFinal)
    }
    else{
      Actual<-sample(c(1:length(Q[1,])), size=1, prob =
        Sigma[Actual,]) #Siguiete estado
      ModificacionTray<-c(ModificacionTray , Actual) #
        Agrego el estado a la trayectoria
    }
  }
  ModificacionTime<-c(ModificacionTime , TiemposDeSalto) #
    Agregamos los tiempos
  i<-i+1
}
}
return(list(Est=ModificacionTray , Tiempos=ModificacionTime))
}

```

## A.11. Norma Matricial Modificada

El Código A.11 calcula la norma de una matriz de tal forma que no toma en cuenta la diagonal. Ésta servirá para calcular los errores de las estimaciones del generador infinitesimal de un PSM.

### Parámetros de entrada:

**Matriz:** Objeto tipo matriz

### Salida:

**Max:** Norma 1 de la matriz sin tomar en cuenta la diagonal.

#### Código A.11: Norma 1 Matricial

```

Norma<-function(Matriz){
  n<-length(Matriz[,1]) #Dimension de la matriz
  Max<-0 #Maximo en cero
  for (i in 1:n) {
    Col<-Matriz[,i]
    Col[i]<-0 #No se toma en cuenta la diagonal
    if (sum(abs(Col))>Max) { #Condicion de norma
      Max<-sum(abs(Col))
    }
  }
  return(Max)
}

```

## A.12. Algoritmo EM Estocástico para estimar el Generador Infinitesimal de un PSM con información incompleta

El Código A.12 implementa el Algoritmo EM Estocástico para obtener el EMV del generador infinitesimal de un PSM estándar cuando tenemos información incompleta y periodo de calentamiento igual a 1000. Dicho código hace uso del Código A.6.

### Parámetros de entrada:

**Trayectoria:** Es un vector que contiene los estados visitados por el proceso (en orden) en la observación a tiempo discreto.

**Tiempos:** Es un vector que contiene los tiempos de observación del PSM.

**n:** Es un entero positivo que indica el número de veces que se va a iterar el algoritmo.

**Metodo:** Es un número del conjunto: 1,2,3. Dichos números indican que el método de construcción de los Puentes Markovianos será Bisección, Aceptación-Rechazo y Uniformización respectivamente.

**Q:** Matriz que contiene el generador infinitesimal con la estimación inicial.

### Salida:

**Gen:** Matriz que contiene el EMV del generador infinitesimal del PSM.

**Código A.12:** Algoritmo EM Estocástico para Estimar el Generador Infinitesimal de un PSM Estándar

```
EME<-function ( Trayectoria , Tiempos , n , Metodo , Q) {
  #Algoritmo para estimar el Gen de un PSM con informacion
  discretamente observada mediante EME
  i<-0
  EE<-Estados( Trayectoria )
  if (Metodo==1){ #Metodo 1 quiere decir que el metodo sera el
    de Biseccion
    Q1<-diag ( 0 , length (EE) , length (EE) )
    while (TRUE) {
      i<-i+1
      #Paso E
      Biseccion<-Reconstruccion ( Trayectoria , Tiempos , Q) #
        Reconstruimos la trayectoria
      BisDepurada<-Depurar ( Biseccion$Est , Biseccion$Tiempos)
      #Paso M
      QE<-Estimacion ( BisDepurada$Est [ - 1 ] , c ( BisDepurada$
        Tiempos , Tiempos [ length ( Tiempos ) ] ) , EE)
      Q<-QE$Gen
      if ( i > 1000) { #Periodo de calentamiento
        Q1<-(Q1*(i-1001)+Q)/(i-1000) #Promedio
```

```

    }
    if ( i==n+1000){
        return ( list ( Gen=Q1 ) )
    }
}
}
if (Metodo==2){ #Metodo 2 quiere decir que el metodo sera el
    de Aceptacion-Rechazo
    Q1<-diag ( 0 , length ( EE ) , length ( EE ) )
    while ( TRUE ) { #Repetimos el ciclo
        i<-i+1
        #Paso E
        AceptacRech<-AR ( Trayectoria , Tiempos , Q ) #Reconstruimos
            la trayectoria
        ARDepurado<-Depurar ( AceptacRech$Est , AceptacRech$Tiempos
            )
        #Paso M
        QE<-Estimacion ( ARDepurado$Est [ - 1 ] , c ( ARDepurado$Tiempos ,
            Tiempos [ length ( Tiempos ) ] ) , EE )
        Q<-QE$Gen
        if ( i > 1000 ) { #Periodo de calentamiento
            Q1<-( Q1*( i - 1001 ) + Q ) / ( i - 1000 ) #Promedio
        }
        if ( i==n+1000 ) {
            return ( list ( Gen=Q1 ) )
        }
    }
}
}
if (Metodo==3){ #Metodo 3 quiere decir que el metodo sera el
    de uniformizacion
    Q1<-diag ( 0 , length ( EE ) , length ( EE ) )
    while ( TRUE ) {
        i<-i+1
        #Paso E
        Uniform<-Uniformizacion ( Trayectoria , Tiempos , Q ) #
            Reconstruimos la trayectoria
        UnifDepurado<-Depurar ( Uniform$Est , Uniform$Tiempos )
        #Paso M
        QE<-Estimacion ( UnifDepurado$Est [ - 1 ] , c ( UnifDepurado$
            Tiempos , Tiempos [ length ( Tiempos ) ] ) , EE )
        Q<-QE$Gen
        if ( i > 1000 ) { #Periodo de calentamiento

```

```

        Q1<-(Q1*(i-1001)+Q)/(i-1000) #Promedio
    }
    if (i==n+1000){
        return (list (Gen=Q1))
    }
}
}
}

```

### A.13. Algoritmo EM Monte Carlo para estimar el Generador Infinitesimal de un PSM con información incompleta

El Código A.13 implementa el Algoritmo EM Estocástico para obtener el EMV del generador infinitesimal de un PSM estándar cuando tenemos información incompleta. Dicho código hace uso del Código A.6.

#### Parámetros de entrada:

**Trayectoria:** Es un vector que contiene los estados visitados por el proceso (en orden) en la observación a tiempo discreto.

**Tiempos:** Es un vector que contiene los tiempos de observación del PSM.

**n:** Es un entero positivo que indica el número de veces que se va a iterar el algoritmo.

**m:** Es un entero positivo que indica el número de veces que se repetirá la construcción de puentes para estimar la función de verosimilitud por MonteCarlo.

**Método:** Es un número del conjunto: 1,2,3. Dichos números indican que el método de construcción de los Puentes Markovianos será Bisección, Aceptación-Rechazo y Uniformización respectivamente.

**Q:** Matriz que contiene el generador infinitesimal con la estimación inicial.

#### Salida:

**Gen:** Matriz que contiene el EMV del generador infinitesimal del PSM.

**Código A.13:** Algoritmo EM Monte Carlo para Estimar el Generador Infinitesimal de un PSM Estándar

```

EMMC<-function ( Trayectoria , Tiempos , n , m , Metodo , Q ) {
    #Algoritmo para estimar el generador de un Proceso de
    Saltos de Markov con informacion discretamente observada
    mediante EMMC
    EE<-Estados ( Trayectoria )
    i<-0
    if (Metodo==1){ #Metodo 1 quiere decir que el metodo sera el
        de Biseccion
    }
}

```

```

while (TRUE) {
  i<-i+1
  S<-diag(0,nrow = length(EE))
  for (j in 1:m) {#MonteCarlo
    #Paso E
    Biseccion<-Reconstruccion (Trayectoria ,Tiempos ,Q) #
      Reconstruimos la trayectoria
    BisDepurada<-Depurar ( Biseccion$Est , Biseccion$Tiempos)
    #Paso M
    S<-S+Estimacion ( BisDepurada$Est [-1] ,c ( BisDepurada$
      Tiempos , Tiempos [ length ( Tiempos ) ] ) ,EE)
  }
  Q<-S/m
  if (i==n) {
    return(Q)
  }
}
}
if (Metodo==2){ #Metodo 2 quiere decir que el metodo sera el
  de Aceptacion-Rechazo
  while (TRUE) {
    i<-i+1
    S<-diag(0,nrow = length ( EstadosVisitados ))
    for (j in 1:m) { #MonteCarlo
      #Paso E
      AceptacRech<-AR( Trayectoria ,Tiempos ,Q) #Reconstruimos
        la trayectoria
      ARDepurado<-Depurar ( AceptacRech$Est , AceptacRech$
        Tiempos)
      #Paso M
      S<-S+Estimacion ( ARDepurado$Est [-1] ,c ( ARDepurado$
        Tiempos , Tiempos [ length ( Tiempos ) ] ) ,EE)
    }
    Q<-S/m
    if (i==n) {
      return(Q)
    }
  }
}
}
if (Metodo==3){ #Metodo 3 quiere decir que el metodo sera el
  de uniformizacion
  while (TRUE) {

```

```

i<-i+1
S<-diag(0,nrow=length( EstadosVisitados ))
for (j in 1:m) {
  #Paso E
  Uniform<-Uniformizacion( Trayectoria , Tiempos ,Q) #
  Reconstruimos la trayectoria
  UnifDepurado<-Depurar( Uniform$Est , Uniform$Tiempos )
  #Paso M
  S<-S+Estimacion( UnifDepurado$Est [ -1] ,c( UnifDepurado$
    Tiempos , Tiempos [ length( Tiempos ) ] ) ,EE)
}
Q<-S/m
if (i==n) {
  return(Q)
}
}
}
}

```

## A.14. Simulación de un MB

El Código A.14 simula una trayectoria de un MB.

### Parámetros de entrada:

**N:** Es un entero positivo que indica el número de saltos del proceso.

**delta:** Es un número positivo que indica el espaciamiento entre las observaciones.

### Salida:

**MB:** Un objeto tipo timeseries que contiene la trayectoria del proceso.

**Código A.14:** Simulación de un MB

```

MB<-function(N, delta) {#Funcion para simular un MB
  MovB<-c(0)
  for (i in 1:N) {
    MovB<-c(MovB, MovB[ length( MovB ) ] + rnorm(1, mean = 0, sd=sqrt(
      delta)))
  }
  return( ts( MovB, start = 0, deltat = delta ) )
}

```

## A.15. Simulación de un MB con Deriva vía su solución analítica

El Código [A.15](#) simula una trayectoria de un MB con Deriva usando la solución analítica de la EDE.

### Parámetros de entrada:

**param:** Es un vector de dos entradas. En la primer entrada se encuentra la media del MB con Deriva y en la segunda entrada se encuentra la desviación estándar del MB con Deriva.

**N:** Es un entero positivo que indica el número de saltos del proceso.

**delta:** Es un número positivo que indica el espaciamento entre observaciones.

### Salida:

**MBD:** Un objeto tipo timeseries que contiene la trayectoria del proceso.

**Código A.15:** Simulación de un MB con Deriva Vía su Solución Analítica

```
MBD<-function (param ,N, delta ) {#Funcion para simular MBD
  analiticamente
  t<-seq (0 ,N*delta , length .out = N+1)
  t<-t*param [1]
  MovB<-MB(N, delta )
  MovB<-param [2] *MovB
  MovBD<-t+MovB
  return ( ts (MovBD, start = 0 ,deltat = delta ))
}
```

## A.16. Simulación de un MBG vía su solución analítica

El Código [A.16](#) simula una trayectoria de un MBG usando la solución analítica de la EDE.

### Parámetros de entrada:

**param:** Es un vector de dos entradas. En la primer entrada se encuentra la media del MBG y en la segunda entrada se encuentra la volatilidad del MBG.

**X0:** Es un número positivo que indica el valor inicial del proceso.

**t0:** Es un número positivo que indica el tiempo inicial del proceso.

**N:** Es un entero positivo que indica el número de saltos del proceso.

**delta:** Es un número positivo que indica el espaciamento entre observaciones.

### Salida:

**MBG:** Un objeto tipo timeseries que contiene la trayectoria del proceso.

**Código A.16:** Simulación de un MBG Vía su Solución Analítica

```

MBG<-function (param ,X0 ,t0 ,N, delta) {#Funcion para simular MBG
  analiticamente
  MovBD<-MBD(c (param [1] -0.5*param [2]^2 ,param [2]) ,N, delta)
  MovBG<-exp (MovBD)
  MovBG<-X0*MovBG
  return (ts (MovBG, start = t0 ,deltat = delta))
}

```

## A.17. Estimación de Parámetros de un MBG

El Código A.17 encuentra los EMV de un MBG.

**Parámetros de entrada:**

**X:** Son los valores del proceso en cada tiempo observado.

**Delta:** Es un número positivo que indica la separación de tiempo entre observaciones.

**Diferenciado:** Es un argumento tipo booleano. Es TRUE si las observaciones del proceso están diferenciadas en log rendimientos y FALSE en caso contrario.

**Salida:**

**Mu:** EMV de la media del MBG.

**Sigma:** EMV de la volatilidad del MBG.

**Código A.17:** Estimación de Parámetros de un MBG

```

FitBS<-function (X, Delta , Diferenciado) {#Funcion para obtener
  los EMV de los parametros de un MBG
  if (!Diferenciado) { #En caso de que no se hayan
    proporcionado los logrendimientos se obtienen
    X<-diff(log(X)) #calculo de los logrendimientos
  }
  n<-length(X)
  SIGMA<-sqrt (var(X)*((n-1)/n)/Delta) #EMV de sigma
  MU<-(mean(X)+0.5*var(X)*((n-1)/n))/Delta #EMV de mu
  return (list (Mu=MU, Sigma=SIGMA))
}

```

## A.18. Simulación del Modelo de Difusión Markov-Modulado con Saltos

El Código A.18 simula una trayectoria proveniente del Modelo de Difusión Markov-Modulado con Saltos.

### Parámetros de entrada:

**Trayectoria:** Es un vector que contiene los estados visitados de un PSM (en orden).

**Tiempos:** Es un vector que contiene los tiempos de salto del PSM.

**Mu:** Es un vector que contiene las medias de la difusión.

**Volatilidad:** Es un vector que contiene las volatilidades de la difusión.

**Delta:** Es un número positivo que contiene el espaciado entre tiempos de la difusión.

**ParamSalto:** Es un número positivo que contiene el parámetro de la variable exponencial de los saltos.

**x0:** Es un número que representa el valor inicial del proceso.

### Salida:

**Difusion:** Es un objeto de tipo timeseries que contiene la trayectoria del proceso dada por la simulación.

#### Código A.18: Simulación del Modelo de Difusión Markov-Modulado con Saltos

```
ToTS<-function ( Trayectoria , Tiempos , ElimT , Delta ) { #Ayuda a
  eliminar observaciones adicionales obtenidas por la
  funcion SimJumpDif
  for ( i in 1:length( ElimT ) ) {
    Indice<-match( ElimT [ i ] , Tiempos ) #Se busca la posicion en
    los tiempos
    Trayectoria<-Trayectoria[-Indice] #Se elimina tiempo y
    valor en la trayectoria del proceso
    Tiempos<-Tiempos[-Indice]
  }
  return ( ts ( Trayectoria , start = Tiempos [ 1 ] , deltat = Delta ) )
}

SimJumpDif<-function ( Trayectoria , Tiempos , Mu , Volatilidad , Delta
, ParamSalto , x0 ) {
  #Simula una difusion con saltos modulada
  Difusion<-c ( x0 ) #Inicio
  DifusionT<-seq ( Tiempos [ 1 ] , Tiempos [ length ( Tiempos ) ] , by =
    Delta )
  Ajuste<-0 #Nos servira para los saltos
  n<-0 #Cuenta las observaciones que hemos simulado
```

```

for (i in 1:length(Trayectoria)) { #Tenemos que cambiar de
  parametros este numero de veces
  Inicio<-n*Delta #Inicio de la ventana de tiempo que
    queremos simular
  Longitud<-Tiempos[i+1]-Inicio #Longitud de la ventana de
    tiempo donde simularemos
  #un tramo con los mismos parametros
  m<-floor(Longitud/Delta) #Numero entero de saltos antes
    de salir de la ventana de tiempo
  #Simulacion
  Difusion<-c(Difusion ,sde.sim(model="BS" ,theta=c(Mu[
    Trayectoria[i] , Volatilidad [ Trayectoria [ i ] ] ) , delta=
    Delta ,N=m,method="milstein" ,X0=Difusion [ length (
    Difusion ) ] [-1] )
  #Tiempos
  DifusionT<-c(DifusionT , Tiempos [ i +1] )
  n<-n+m #Total de observaciones hasta el momento
  Restante<-Tiempos [ i +1] -n*Delta #Tiempo restante antes de
    cambiar de parametro
  #Simulamos cuanto vale el proceso en el instante en el
    que cambia de parametros
  Difusion<-c(Difusion , Difusion [ length (Difusion ) ] *rlnorm(1 ,
    meanlog=(Mu[ Trayectoria [ i ] ] -0.5*( Volatilidad [
    Trayectoria [ i ] ] ^2 ) ) *Restante , sdlog=Volatilidad [
    Trayectoria [ i ] ] *sqrt ( Restante ) ) )
  if (!i==length(Trayectoria)) { #Si no es el ultimo cambio
    de parametros
    #Necesitamos simular un salto y cambiar los parametros
    n<-n+1 #Agregamos uno a las observaciones
    Ajuste<-n*Delta-Tiempos [ i +1] #Calculamos cuanto tiempo
      de la ventana de tiemposucede antes de observarlo
    #Simulamos un salto y una observacion con los nuevos
      parametros
    Difusion<-c(Difusion , Difusion [ length (Difusion ) ] *exp(
      rexp(n = 1 , rate = ParamSalto [ 1 ] ) ) *rlnorm(1 , meanlog=(
      Mu[ Trayectoria [ i +1] ] -0.5*( Volatilidad [ Trayectoria [ i
      +1] ] ^2 ) ) *Ajuste , sdlog=Volatilidad [ Trayectoria [ i +1] ] *
      sqrt ( Ajuste ) ) )
  }
}
Difusion<-Difusion[-length(Difusion)]
DifusionT<-sort(DifusionT[-length(DifusionT)])

```

```

return (ToTS(Difusion , DifusionT , Tiempos[-c(1 , length(Tiempos)
) ] , Delta))
}

```

## A.19. Identificación de los Saltos

El Código A.19 hace identificación de los saltos en un conjunto de datos y separa los rendimientos con salto de los que no tienen salto.

### Parámetros de entrada:

**Serie:** Es un vector con la trayectoria del proceso (valores).

**Cortes:** Son las posiciones de los saltos obtenidos mediante la función Separar.

### Salida:

**RendMBG:** Rendimientos sin salto.

**RendSalto:** Rendimientos con salto.

### Código A.19: Identificación de Saltos para Estimar Parámetros del Modelo

```

Separar<-function(Retornos , Tolerancia) { #Obtiene las
  observaciones que contienen salto de acuerdo con una
  tolerancia dada
  #Retornos es un vector con los logrendimientos cuadrados de
  la serie
  #Tolerancia es el umbral permitido para los rendimientos
  sin salto
  Cortes<-c()
  Indice<-0
  for (Rendimiento in Retornos) { #Recorre todos los
    rendimientos
    Indice<-Indice+1
    if (Rendimiento>=Tolerancia) { #En caso de sobrepasar el
      umbral, guarda la posicion del rendimiento
      Cortes<-c(Cortes , Indice)
    }
  }
  return(Cortes)
}

SepararRend<-function(Serie , Cortes) { #Separa rendimientos con
  salto y sin salto
  #Serie es un vector con la coleccion de valores del proceso
  #Cortes son las posiciones de los saltos obtenidas mediante
  Separar

```

```

Cortes<-c(Cortes ,length( Serie )) #Añamos un punto adicional
a los cortes
n<-length(Cortes) #Numero de cortes
Obs<-c() #Rendimientos sin salto
Saltos<-c() #Rendimientos con salto
Inicio<-1 #inicio
for ( i in 1:n) { #Tomamos los rendimientos sin salto por
cluster
Obs<-c(Obs, diff(log( Serie [ Inicio : Cortes [ i ] ] ) ) )
Inicio<-Cortes [ i ]+1
if (!i==n) { #rendimientos con salto
Saltos<-c( Saltos , diff(log( Serie [ Cortes [ i ] : Inicio ] ) ) )
}
}
return( list (RendMBG=Obs, RendSalto=Saltos) )
}

```

## A.20. Estimación de Parámetros de la Difusión del Modelo de Difusión Markov-Modulado con Saltos

El Código A.20 obtiene los EMV de los parámetros de la difusión del Modelo de Difusión Markov-Modulado con Saltos mediante el Algoritmo EM para mezclas.

### Parámetros de entrada:

**alpha:** Vector que contiene los valores iniciales para las ponderaciones de la mezcla.

**mu:** Vector que contiene los valores iniciales para las medias de la difusión.

**sigm:** Vector que contiene los valores iniciales para las volatilidades de la difusión.

**Observaciones:** Vector que contiene las observaciones de los rendimientos sin salto del proceso.

**Tolerancia:** Número positivo que indica el épsilon para la condición de paro del algoritmo EM.

### Salida:

**Alphas:** Vector que contiene los EMV para las ponderaciones de la mezcla.

**Medias:** Vector que contiene los EMV para las medias de la difusión.

**Volatilidades:** Vector que contiene los EMV para las volatilidades de la difusión.

**Código A.20:** Estimación de Parámetros la Difusión del Modelo de Difusión Markov-Modulado con Saltos

```

Gama<-function( alpha ,mu, sigm , k , y) { #Calcula la probabilidad
definida como gamma a una observacion y

```

```

m<-length(alpha)
S<-0
for (j in 1:m) { #Se obtiene la probabilidad total
  S<-S+alpha[j]*dnorm(y,mean=mu[j],sd=sqrt(sigm[j]))
}
return(alpha[k]*dnorm(y,mean=mu[k],sd=sqrt(sigm[k]))/S)
}

GammaObs<-function(k,alpha,mu,sigm,Observaciones){ #Funcion
  de apoyo, calcula la probabilidad gamma a un vector de
  Observaciones
  G<-c()
  for (y in Observaciones) { #Se aplica la funcion Gama a
    cada elemento del vector
    G<-c(G,Gama(alpha,mu,sigm,k,y))
  }
  return(G)
}

GammaSuma<-function(GammaObservaciones){ #Funcion de apoyo,
  suma las probabilidades provenientes de GammaObs
  return(sum(GammaObservaciones))
}

AlphaEstim<-function(N,Suma){ #Estimacion de la ponderacion
  return(Suma/N)
}

ParamEstim<-function(GammaObservaciones,Observaciones,Suma){
  #Estimacion de media y varianza
  Esp<-GammaObservaciones*Observaciones
  MU<-sum(Esp)/Suma
  V<-sum(GammaObservaciones*((Observaciones-MU)^2))
  return(list(Media=MU,Varianza=V/Suma))
}

MixLogVerosim<-function(alpha,mu,sigm,Observaciones){ #
  Obtiene la logverosimilitud de la muestra
  m<-length(alpha)
  S<-0
  GamObs1<-c()
  GamObs2<-c()

```

A.20. ESTIMACIÓN DE PARÁMETROS DE LA DIFUSIÓN DEL MODELO DE DIFUSIÓN MARKOV

```

GamObs3<-c()
for (k in 1:m) {
  X<-log(alpha[k])+dnorm(Observaciones,mean=mu[k],sd=sqrt(
    sigm[k]),log=T)
  Y<-GammaObs(k,alpha,mu,sigm,Observaciones)
  if (k==1) {
    GamObs1<-Y
  }
  if (k==2) {
    GamObs2<-Y
  }
  if (k==3) {
    GamObs3<-Y
  }
  S<-S+sum(X*Y)
}
return(list(Verosimilitud=S,G1=GamObs1,G2=GamObs2,G3=
  GamObs3))
}

EMMix<-function(alpha,mu,sigm,Observaciones,Tolerancia){#
  Algoritmo EM para mezclas
  N<-length(Observaciones)
  VV<-MixLogVerosim(alpha,mu,sigm,Observaciones)
  while (T) {
    GObs1<-VV$G1
    GObs2<-VV$G2
    GObs3<-VV$G3
    GS1<-GammaSuma(GObs1)
    GS2<-GammaSuma(GObs2)
    GS3<-GammaSuma(GObs3)
    alpha<-c(AlphaEstim(N,GS1),AlphaEstim(N,GS2),AlphaEstim(N
      ,GS3))
    #print(alpha)
    P1<-ParamEstim(GObs1,Observaciones,GS1)
    P2<-ParamEstim(GObs2,Observaciones,GS2)
    P3<-ParamEstim(GObs3,Observaciones,GS3)
    mu<-c(P1$Media,P2$Media,P3$Media)
    #print(mu)
    sigm<-c(P1$Varianza,P2$Varianza,P3$Varianza)
    #print(sigm)
    VN<-MixLogVerosim(alpha,mu,sigm,Observaciones)
  }
}

```

```

Diferencia<-abs(VN$Verosimilitud-VV$Verosimilitud)
#print(Diferencia)
if (Diferencia<Tolerancia) { #Si ya se dio la
  convergencia
  mu<-mu+sigm*0.5 #Se recupera la media de la difusion
  return(list(Alphas=alpha, Medias=mu, Volatilidades=sqrt(
    sigm)))
}
VV<-VN
}
}

```

## A.21. Clasificación de Bloques

El Código [A.21](#) clasifica los bloques de rendimientos obtenidos usando los EMV obtenidos del código [A.20](#).

### Parámetros de entrada:

**Serie:** Vector que contiene los valores del proceso en los tiempos de observación.

**Cortes:** Cortes obtenidos mediante la función Separar en el Código [A.19](#).

**Alphas:** Vector que contiene los EMV de las ponderaciones de la mezcla.

**Mu:** Vector que contiene los EMV de las medias de la difusión.

**Sigm:** Vector que contiene los EMV de las volatilidades de la difusión.

**Delta:** Número positivo que representa el espaciamiento de tiempo entre observaciones.

### Salida:

**Est:** Vector que contiene la trayectoria (estados visitados en orden) a tiempo discreto del PSM subyacente.

**Tiempos:** Vector que contiene los tiempos de observación de la trayectoria del PSM.

### Código A.21: Clasificación de Bloques

```

Clasificacion<-function(Serie, Cortes, Alphas, Mu, Sigm, Delta) { #
  Algoritmo de clasificacion de bloques
  Cortes<-c(Cortes, length(Serie))
  n<-length(Cortes)
  EstadoObs<-c()
  TiempoObs<-seq(0, length.out = length(Serie), by = Delta)
  Inicio<-1
  Proba1<-1
  Proba2<-1
  Proba3<-1
}

```

```

for (i in 1:n) {
  Obs<-diff(log(Serie[Inicio:Cortes[i]]))
  for (x in Obs) {
    if (length(EstadoObs)==0) {
      Proba1<-Proba1*Gama(Alphas,Mu,Sigm,1,x)
      Proba2<-Proba2*Gama(Alphas,Mu,Sigm,2,x)
      Proba3<-Proba3*Gama(Alphas,Mu,Sigm,3,x)
    }
    else{
      if (EstadoObs[length(EstadoObs)]==1) {
        Proba1<-0
        Proba2<-Proba2*Gama(Alphas,Mu,Sigm,2,x)
        Proba3<-Proba3*Gama(Alphas,Mu,Sigm,3,x)
      }
      if (EstadoObs[length(EstadoObs)]==2) {
        Proba1<-Proba1*Gama(Alphas,Mu,Sigm,1,x)
        Proba2<-0
        Proba3<-Proba3*Gama(Alphas,Mu,Sigm,3,x)
      }
      if (EstadoObs[length(EstadoObs)]==3) {
        Proba1<-Proba1*Gama(Alphas,Mu,Sigm,1,x)
        Proba2<-Proba2*Gama(Alphas,Mu,Sigm,2,x)
        Proba3<-0
      }
    }
  }
  Eleccion<-sample(c(1,2,3),size=1,prob=c(Proba1,Proba2,
    Proba3))
  EstadoObs<-c(EstadoObs,rep(Eleccion,each=length(Obs)+1))
  Proba1<-1
  Proba2<-1
  Proba3<-1
  Inicio<-Cortes[i]+1
}
return(list(Est=EstadoObs,Tiempos=TiempoObs))
}

```



# Bibliografía

- [1] Howard Taylor and Samuel Karlin. *An Introduction to Stochastic Modelling*. Academic Press, 1998.
- [2] Mogens Bladt. *A REVIEW IN PHASE-TYPE DISTRIBUTIONS AND THEIR USE IN RISK THEORY*. ASTIN BULLETIN, 2005.
- [3] Jacobsen, M. *Statistical analysis of counting processes*. Lect. Notes Statist, 1982.
- [4] Søren Asmussen and Asger Hobolth. *Markov Bridges, Bisection and Variance Reduction*. THIELE CENTRE for applied mathematics in natural science, Research Report No. 01, 2011.
- [5] Luz Judith R. Esparza and Fernando Baltazar Larios. *Bayesian Estimation for a Mortality Model via the Aging Process*. March 9, 2020.
- [6] Cristina Guzmán Solis. (2018) *Algoritmo EM: Implementación y Aplicaciones* (Tesis de Licenciatura). Facultad de Ciencias UNAM, Ciudad de México.
- [7] Karl Sigman. *1 IEOR 6711: Notes on the Poisson Process*. 2006.
- [8] Luis Rincón. (2012) *Introducción a los procesos estocásticos*. Facultad de Ciencias UNAM, Ciudad de México.
- [9] Michael J. Steele. (2001) *Stochastic Calculus and Financial Applications*. Springer-Verlag New York Inc.
- [10] Rama Cont and Peter Tankov. (2004) *Financial Modelling with Jump Processes*. Chapman & Hall/CRC, Financial Mathematics Series.
- [11] Jodie Duncan, John Randal and Peter Thomson. (2009) *Fitting jump diffusion processes using the EM algorithm*.
- [12] Shih-Kuei Lin, Yu-Min Lian & Szu-Lang Liao. (2014) *Pricing gold options under Markov-modulated jump-diffusion processes*, Applied Financial Economics, 24:12, 825-836, DOI: 10.1080/09603107.2014.914142

- [13] Nicole Bäuerle & Mirko Kötter. (2007) *Markov-modulated diffusion risk models*, Scandinavian Actuarial Journal, 2007:1, 34-52, DOI: 10.1080/03461230601069528
- [14] S. G. Kou and Hui Wang. (2003) *Option pricing under a Double Exponential Jump Diffusion Model*, Department of IEOR, Columbia University, New York.
- [15] Stefano M. Iacus. (2008) *Simulation and Inference for Stochastic Differential Equations With R Examples*, Springer Series in Statistics, New York.
- [16] Michael Spivak. (2012) *Calculus*, Reverté, Barcelona.
- [17] Louis Bachelier. (1900) *Théorie de la Spéculation* (Tesis de Doctorado).
- [18] Fisher Black, Myron Scholes and Robert C. Merton. (1973) *Theory of Rational Option Pricing*
- [19] E.S. Page. (1957) *On problems in which a change in parameter occurs at an unknown point*
- [20] Michele Basseville and Igor V. Nikiforov. (1993) *Detection of Abrupt Changes: Theory and Application*