



# COLEGIO DE POSTGRADUADOS

---

---

INSTITUCIÓN DE ENSEÑANZA E INVESTIGACIÓN EN CIENCIAS AGRÍCOLAS

## CAMPUS MONTECILLO

POSTGRADO DE SOCIOECONOMÍA, ESTADÍSTICA E INFORMÁTICA

ESTADÍSTICA

### Modelo de Espacio de Estados con Observaciones Censuradas

Francisco Julian Ariza Hernández

T E S I S

PRESENTADA COMO REQUISITO PARCIAL  
PARA OBTENER EL GRADO DE:

**DOCTOR EN CIENCIAS**

MONTECILLO, TEXCOCO, EDO. DE MÉXICO

2010

---

---

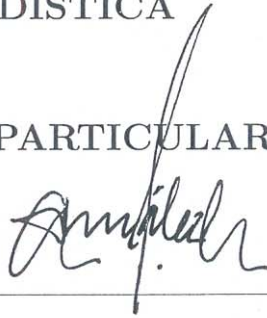
La presente tesis titulada: **Modelo de Espacio de Estados con Observaciones Censuradas**, realizada por el alumno: **Francisco Julian Ariza Hernandez**, bajo la dirección del Consejo Particular indicado ha sido aprobada por el mismo y aceptada como requisito parcial para obtener el grado de:

**DOCTOR EN CIENCIAS**

**SOCIOECONOMÍA, ESTADÍSTICA E INFORMÁTICA  
ESTADÍSTICA**

**CONSEJO PARTICULAR**

CONSEJERO



Dr. Félix V. González Cossio

DIRECTOR



Dr. Gabriel A. Rodríguez Yam

ASESOR



Dr. José A. Villaseñor Alva

ASESOR



Dr. Sergio Pérez Elizalde

ASESOR



Dr. Barry C. Arnold

Montecillo, Texcoco, México, Noviembre de 2010

# Modelo de Espacio de Estados con Observaciones Censuradas

Francisco Julian Ariza Hernández

## Resumen

En este trabajo, presentamos algunas alternativas para estimar los parámetros de un Modelo de Espacio de Estados (SSM, por sus siglas en inglés) cuando se tiene el problema de datos incompletos, los algoritmos de Esperanza-Maximización (EM), Monte Carlo EM y EM Estocástico son implementados. También, se presenta una aproximación a la función de verosimilitud utilizando Muestreo de Importancia. Se realizó un estudio de simulación para estudiar el desempeño de estos procedimientos para un modelo de espacio de estados con diferentes porcentajes de censura en las observaciones. Los algoritmos son implementados a dos conjuntos de datos reales; el primero, a datos sobre contaminación del aire con observaciones sujetas a límites inferiores de detección y con datos perdidos; el segundo, a datos sobre contaminación de agua sujetos también a límites inferiores de detección.

**Palabras clave:** algoritmo EM, algoritmo EM estocástico, algoritmo EM Monte Carlo, Recursiones de Kalman, límites de detección, datos perdidos.

# State-Space Model with Censored Observations

Francisco Julian Ariza Hernández

## Abstract

In this work, to estimate the parameters of a state-space models (SSM) with incomplete data, the Expectation-Maximization (EM) algorithm, the Monte Carlo EM (MCEM) algorithm and Stochastic EM (SEM) algorithm are implemented. Also we present an approximation to the likelihood function via importance sampling (IS). To study the performance of these procedures a simulation study for a state-space model with different rates of censoring is conducted. The algorithms are implemented to an air pollution data subject to lower limits of detection with missing observations and to a water pollution data, also subject to lower limits of detection.

**Keywords:** EM algorithm, Stochastic EM algorithm, Monte Carlo EM algorithm, Kalman recursions, limits of detection, missing data.

## AGRADECIMIENTOS

Al Consejo Nacional de Ciencia y Tecnología (CONACYT) por su apoyo económico brindado durante todos mis estudios de posgrado.

Al Colegio de Postgraduados, por haberme brindado la oportunidad de seguir mi formación académica en sus aulas.

Quiero expresar mi más profundo y mayor agradecimiento a los integrantes de mi Consejo Particular:

Dr. Félix V. González Cossio, por su gran disposición que tuvo siempre durante todo el proceso y culminación de este proyecto.

Dr. Gabriel A. Rodríguez Yam, por su excelente dirección, consejos, sugerencias, recomendaciones y su disposición, sin los cuales me hubiera sido mucho más difícil la culminación del presente trabajo.

A los doctores Dr. José A. Villaseñor Alva, Dr. Sergio Pérez Elizalde y Barry C. Arnold, por sus observaciones, consejos y ayuda para la realización del trabajo.

Al Dr. Humberto Vaquera Huerta por sus consejos y ayuda desinteresada durante toda mi estancia en esta institución.

A mis profesores, compañeros de clases y todos aquellos que de alguna u otra manera fueron copartícipes de esta tarea.

A la Unidad Académica de Matemáticas de la Universidad Autónoma de Guerrero, en la cual he encontrado excelentes compañeros de trabajo y amigos. Gracias por todo el apoyo que me brindaron tanto moral como de ambiente laboral para poder culminar este trabajo.

Finalmente quiero hacer patente mi agradecimiento a mi familia, mi padre Sr. Francisco Ariza Bahena, mis hermanos, mi esposa y mis hijas, por su constante amor, apoyo, paciencia y entendimiento que hicieron posible la realización y culminación de este proyecto.

-

*A la memoria de mi madre.*

*A mi padre y hermanos.*

*Con amor para Esmeralda, Barbara y la pequeña Julieta.*

# Índice

<b>1. Introducción</b>	<b>1</b>
<b>2. Objetivos</b>	<b>5</b>
2.1. Objetivos generales . . . . .	5
2.2. Objetivos particulares . . . . .	5
<b>3. Materiales y Métodos</b>	<b>6</b>
3.1. Datos Censurados . . . . .	6
3.1.1. Tipos de Censura . . . . .	7
3.1.2. Clasificación de datos censurados . . . . .	8
3.2. Conceptos básicos de series de tiempo . . . . .	9
3.2.1. Ejemplos de series de tiempo . . . . .	9
3.2.2. Algunos modelos de series de tiempo . . . . .	12
3.3. Modelos de Espacio de Estados Generalizado . . . . .	15
3.3.1. Modelo de Espacio de Estados Lineal . . . . .	17
3.3.2. Recursiones de Kalman . . . . .	19

# ÍNDICE

---

3.3.3. Estimación en el modelo SSM Generalizado . . . . .	22
3.4. Algoritmo EM . . . . .	24
3.4.1. Algoritmo EM Monte Carlo (MCEM) . . . . .	26
3.4.2. Algoritmo EM Estocástico . . . . .	27
3.4.3. Varianza del estimador . . . . .	28
3.5. Muestreo de Importancia . . . . .	29
<b>4. Modelo de Espacio de Estados Censurado</b>	<b>32</b>
4.1. El modelo CSSM . . . . .	32
4.2. Estimación del SSMCen con el Algoritmo EM . . . . .	34
4.2.1. Ejemplo. Modelo Lineal Gaussiano . . . . .	35
4.2.2. Distribución predictiva . . . . .	39
4.3. Algoritmo EM Monte Carlo . . . . .	41
4.4. Algoritmo EM Estocástico . . . . .	43
4.5. Estimación del SSMCen usando IS . . . . .	43
<b>5. Estudio de Simulación</b>	<b>46</b>
<b>6. Aplicaciones a datos reales</b>	<b>52</b>
6.1. Datos de Cedar R Logan . . . . .	52
6.2. Datos Zeger . . . . .	54
<b>7. Conclusiones</b>	<b>60</b>



# ÍNDICE

---

<b>Referencias</b>	<b>61</b>
<b>Apéndice</b>	<b>62</b>
Apéndice A: Función de verosimilitud del modelo de espacio de estados . . .	62
Apéndice B: Distribución de suavizamiento . . . . .	66
Apéndice C: Funciones y procedimientos realizados en el software R . . . . .	68

# Índice de cuadros

5.1. Límites de censura. . . . .	47
5.2. Sesgo estimado y error cuadrático medio (en paréntesis) utilizando $S = 500$ repeticiones, con diferentes porcentajes de censura, $\theta = (30, 2, 0.9, 1.0)$ y series de tiempo de tamaño 100. . . . .	49
5.3. Sesgo estimado y error cuadrático medio (en paréntesis) utilizando $S = 500$ repeticiones, con diferentes porcentajes de censura, $\theta = (30, 2, 0.9, 1.0)$ y series de tiempo de longitud 500. . . . .	50
6.1. Estimadores EM, MCEM, SEM y IS de modelo SSM lineal Gaussiano para los datos de Cedar R Logan. . . . .	53
6.2. Longitud de las cadenas generadas para cada nivel de censura en los datos . . . . .	54
6.3. Concentraciones mensuales de $\text{NH}_4$ (mequiv/sq m) en Lawrence Livermore, California. . . . .	57
6.4. Estimadores EM, MCEM, SEM y IS de los parámetros del modelo SSM lineal Gaussiano para los datos presentados por ?. . . . .	57

# Índice de figuras

3.1. Serie de tiempo de fósforo disuelto (mg/L) . . . . .	10
3.2. Concentraciones de $\text{NH}_4$ (mequiv/sq m) en Lawrence Livermore, California, mayo de 1977 a noviembre de 1980. . . . .	11
3.3. Precipitación pluvial en Chilapa, Gro., en el periodo de enero de 1977 a septiembre de 1988. . . . .	12
3.4. Realización de un modelos $AR(1)$ con: $\phi = 0.9$ (gráfica superior) y $\phi = -0.9$ (gráfica inferior) . . . . .	15
3.5. Realización de un modelo $MA(1)$ con: $\theta = 0.5$ (gráfica superior) y $\theta = -0.5$ (gráfica inferior) . . . . .	16
5.1. Serie de tiempo simulada . . . . .	47
6.1. Monitoreo de las medias (running means) para $\mu, \sigma^2, \phi$ y $\tau^2$ . . . . .	55
6.2. Gráfica de autocorrelaciones de $\mu, \sigma^2, \phi$ y $\tau^2$ . . . . .	56
6.4. Ajuste de la serie de tiempo de Concentraciones de $\text{NH}_4$ . . . . .	59

# Capítulo 1

## Introducción

Una serie de tiempo es un conjunto de observaciones registradas de forma sucesiva en el tiempo. Con frecuencia, estas observaciones se realizan en intervalos de tiempo regulares; por ejemplo, un año, un mes o una semana. Las series de tiempo aparecen en casi todas las disciplinas del conocimiento, tales como las ciencias naturales, las ciencias sociales, economía, ingeniería, medicina, meteorología, etc. y su análisis se puede realizar a partir de diferentes enfoques; por ejemplo, el análisis espectral de series de tiempo, con modelos autorregresivos y de promedios móviles (ARMA) presentados por ?, y el análisis armónico, entre otros. Un enfoque más reciente para el análisis de series de tiempo, utilizado en esta tesis, son los modelos de espacio de estados (SSM, por sus siglas en inglés).

Con los SSM se establece una metodología unificada para el análisis de series de tiempo. Con este enfoque, se supone que la evolución en el tiempo del sistema bajo estudio es determinado por una serie de valores no observados,  $\alpha_1, \alpha_2, \dots, \alpha_n$ , los cuales están asociados a una serie de valores observados,  $y_1, y_2, \dots, y_n$ ; la relación entre las  $\alpha_t$ 's y las  $y_t$ 's se especifica mediante un SSM. Así, el propósito del análisis de espacio de estados es inferir propiedades relevantes de las  $\alpha_t$ 's a partir del conocimiento de las observaciones  $y_1, y_2, \dots, y_n$ .

Los SSM constituyen una amplia clase de modelos que proveen un marco flexible para describir, modelar y pronosticar valores futuros de series de tiempo en áreas diversas del conocimiento. Por ejemplo, una de las características atractivas de los modelos SSM es que varios de los modelos tradicionales para el análisis de series de

## 1. Introducción

---

tiempo, tales como los modelos ARMA, ARIMA y de composición clásica, pueden ser expresados como un SSM; estos modelos aparecieron en la literatura estadística en los años sesenta y setenta a través de los trabajos de ? y ? quienes desarrollaron un nuevo enfoque para los problemas de filtro y predicción lineales que fueron utilizados en su origen para resolver problemas, tanto teóricos como prácticos, en la teoría del control y comunicaciones; por ejemplo, la predicción de señales aleatorias, la detección de señales de formas conocidas en presencia de ruidos aleatorios, etc. A partir de entonces, los modelos SSM y sus relaciones con las recursiones de Kalman se han aplicado en el análisis de series de tiempo en diversas disciplinas, como la Biología (???), la Economía (????), la Medicina (??), la Ingeniería (???), etc. Los libros de ?, de ? y ? contienen una explicación extensa de estos modelos y sus aplicaciones. Las recursiones de Kalman juegan un papel importante en el análisis de los modelos SSM tanto lineales como Gaussianos. Estas recursiones se utilizan para la estimación y predicción de todo el proceso, el cual puede ser representado como un modelo de espacio de estado a través de relaciones recursivas de funciones de densidad. El filtro de Kalman también se ha extendido a SSM no lineales y no Gaussianos (ver por ejemplo ?, ?, ? y ?, entre otros). Sin embargo, cuando existe censura en algunas de las observaciones y/o datos perdidos, aún en el caso simple, las recursiones no son directamente aplicables.

La censura ocurre cuando no se conoce exactamente la información completa del fenómeno bajo estudio, pero si una proporción de ella y puede ocurrir de varias maneras. Los libros de ?, ? y ? contienen información sobre diferentes tipos de censura y ejemplos relacionados. El fenómeno de censura en series de tiempo ocurre muy frecuentemente en disciplinas como la física, negocios, economía y ciencias ambientales. Por ejemplo, cuando se monitorea un fenómeno y/o experimento aleatorio de interés durante un periodo de tiempo, tal como un contaminante en un punto de un río o en el aire, la precipitación pluvial en una determinada región, la altura de las nubes en un aeropuerto, por mencionar algunas. El registro de este tipo de datos puede presentar algunas irregularidades, tales como la censura en algunas observaciones, situación que puede deberse, entre otros factores, a que el equipo de medición utilizado tenga límites de detección o haya restricciones de tiempo y/o dinero. Es común, que los investigadores ignoren la censura en series de tiempo, considerando los límites de censura como observaciones sin censura o que eliminen tales datos, de esta forma, al implementar métodos clásicos de análisis de series de tiempo, se obtienen estimaciones ineficientes y sesgadas.

## 1. Introducción

---

Existen algunos métodos simples para el manejo de la censura, como los presentados por [?] que a la fecha se siguen utilizando ([?]). Se han propuesto algunos métodos y procedimientos mas formales para analizar series de tiempo con datos censurados o perdidos. [?] usan el algoritmo Esperanza-Maximización (EM) conjuntamente con las recursiones de Kalman para obtener los estimadores por máxima verosimilitud de los parámetros de un modelo SSM lineal cuando se presentan datos perdidos, pero no tratan el problema de censura en la serie. [?] sugieren una estimación de la función de verosimilitud “completa” y un método de aproximación para un modelo de regresión con errores autorregresivos cuando algunas observaciones son censuradas por la izquierda; sin embargo, los autores mencionan que el método puede ser no factible cuando la razón de censura es alta y proponen un enfoque de pseudo-verosimilitud. [?] obtienen estimadores espectrales y de mínimos cuadrados para estimar parámetros en series de tiempo Gaussianas y en modelos espacio-temporales con datos censurados y/o faltantes. [?] realiza un revisión sobre métodos de imputación múltiple para manejar datos perdidos y/o censurados y presenta tres modelos estadísticos para ajustar datos de contaminación del aire, [?] realiza estimación Kaplan-Meier de una función de distribución del tiempo de falla con distribución marginal común. [?] propusieron un método para la estimación recursiva de los estados Gaussianos parcialmente observados, que está basado en una marginalización, la cual se obtiene mediante el filtro de Kalman. [?] utilizan un método de imputación para ajustar modelos ARMA en presencia de datos censurados, además muestran la efectividad de su técnica en términos de sesgo, eficiencia y pérdida de información, y aplican su método a datos meteorológicos.

En este trabajo de investigación se proponen métodos de inferencia con datos censurados y perdidos en series de tiempo a través de un SSM, en particular, se presentan cuatro procedimientos para la estimación de los parámetros de un SSM lineal Gaussiano cuando se tienen observaciones censuradas por la izquierda. En el primero se implementa el algoritmo EM, en los siguientes dos procedimientos se utilizan versiones estocásticas para el cálculo del paso E del algoritmo, el algoritmo EM Monte Carlo (MCEM, por sus siglas en inglés) y el algoritmo EM Estocástico (SEM, por sus siglas en inglés). En el último caso se estima la verosimilitud del modelo a través de muestreo de importancia.

El contenido de esta tesis es como sigue: en el Capítulo 2 se presentan los objetivos generales y particulares en este trabajo. En el Capítulo 3 se presenta una revisión de

## 1. Introducción

---

los conceptos, métodos y algoritmos útiles para el desarrollo de la investigación. De particular importancia, se realiza una descripción general de los modelos de espacio de estados y el problema de inferencia con datos censurados. En el Capítulo 4 se presentan los métodos propuestos para estimar los parámetros en SSM censurados, dichos métodos se aplican a un modelo lineal Gaussiano con observaciones censuradas. En el Capítulo 5 se realiza un estudio de simulación para comparar los estimadores en términos de su sesgo y error cuadrático medio, y en el Capítulo 6 se dan dos ejemplos de aplicación a datos reales. El primero es sobre contaminantes en un río y en el segundo se analizan los datos presentados por ?. Finalmente, en el Capítulo 7, se presentan las conclusiones de este trabajo.

# Capítulo 2

## Objetivos

### 2.1. Objetivos generales

- Analizar series de tiempo con datos censurados y/o perdidos usando modelos de espacio de estados (SSM) lineales.
- Proponer métodos de estimación de los parámetros del SSM cuando se tienen observaciones censuradas.

### 2.2. Objetivos particulares

- Implementar los algoritmos EM, EM Monte Carlo y EM estocástico para estimar los parámetros del SSM lineal Gaussiano con censura.
- Implementar el muestro de importancia para calcular la verosimilitud del SSM lineal Gaussiano y usar este valor para estimar los parámetros de este modelo.
- Comparar los estimadores obtenidos en términos de sesgo y error cuadrático medio.
- Aplicar los procedimientos propuestos a ejemplos con datos reales.



# Capítulo 3

## Materiales y Métodos

En este capítulo se abordan las definiciones, métodos y procedimientos que se utilizan a lo largo del trabajo, principalmente conceptos de datos censurados, de series de tiempo como un SSM y la estimación de los parámetros involucrados en el modelo.

### 3.1. Datos Censurados

Para obtener o registrar los datos de cualquier análisis estadístico se cuentan con diferentes métodos o procedimientos según sea el área en estudio, el interés del investigador, o bien el tipo de experimento o fenómeno aleatorio en cuestión. Sin embargo, en muchas situaciones, algunas de las observaciones obtenidas no son completamente conocidas, tal situación puede deberse a múltiples factores, como por ejemplo,

1. Cuando la variable de interés es el tiempo de ocurrencia de un evento, como es el caso de medir el tiempo de vida de pacientes enfermos y que algunos de los sujetos en estudio sigan vivos o sin la enfermedad (aliviados) cuando dicho estudio finaliza, de esta forma, los tiempos de ocurrencia del evento son desconocidos, pero se sabe al menos que vivirán más que el periodo de tiempo del estudio.
2. Cuando el registro de las observaciones de un experimento o fenómeno aleatorio bajo estudio se realiza con la ayuda de un instrumento de medición finita o su

### 3.1. Datos Censurados

---

rango de medición es restringido o acotado, es muy posible que algunas de las observaciones no puedan ser registradas exactamente por el equipo, es decir, que estén fuera del rango de medición, de tal forma que dichos valores de los datos no los conoceremos.

Los casos anteriores, son ejemplos de *observaciones censuradas*. Los datos de sobrevivencia típicamente presentan las características del primer caso y se presentan en problemas de Ingeniería, Física, control de calidad, epidemiología, etc. Cuando se trata del segundo caso, también suelen llamarle datos no detectados (non-detect data en inglés, ?) y se presentan frecuentemente en investigaciones medioambientales, en la biología, la meteorología, la medicina, la astronomía, etc.

La censura se puede clasificar de acuerdo al tipo de mecanismo que la produce y de como se presente en los datos.

#### 3.1.1. Tipos de Censura

Existen tres tipos de censura, los cuales se describen a continuación.

**Censura tipo I.** Sucede cuando se determina un periodo de tiempo fijo para el experimento y se registran las observaciones. Las observaciones censuradas serán aquellas donde el tiempo de ocurrencia de evento sea mayor al periodo de tiempo fijado de antemano para el estudio. Por ejemplo, si el estudio consiste en observar el tiempo de falla de ciertos artículos electrónicos, y se fija de antemano un periodo de tiempo de observación,  $t_0$ , de acuerdo a nuestros intereses (limitaciones de tiempo, recursos, etc.), entonces las observaciones exactas serán aquellos tiempos de falla menores a  $t_0$ , es decir aquellos cuando el artículo falla antes del tiempo fijado. Si por el contrario sucede que el artículo falla después de  $t_0$ , se registra el tiempo hasta que finaliza el estudio, considerando que el artículo no ha fallado hasta ese momento, los cuales se denominan tiempos de falla censurados o simplemente datos censurados.

**Censura tipo II.** En este caso, se fija un número de ocurrencias de eventos en la muestra, los cuales indican las observaciones exactas (o completas) y el resto son tomadas como observaciones censuradas. Considerando el ejemplo anterior, el experimento termina cuando se observa un porcentaje de tiempos de falla fijado de

### 3.1. Datos Censurados

---

antemano y el resto se toman como observaciones censuradas.

**Censura tipo III.** También llamada censura aleatoria. Éste tipo de censura se puede ver como una generalización de censura tipo I, considerando a  $t_0$  como una variable aleatoria,  $T_j$ , en lugar de una constante, la cual representa aquellas posibles causas no consideradas por el experimento y que provocan la censura en la  $j$ -ésima observación.

#### 3.1.2. Clasificación de datos censurados

Para cada uno de los tipos de censura, los datos censurados se pueden presentar de la forma siguiente.

**Censura por la izquierda.** Una observación es censurada por la izquierda en  $L$ , si solo se sabe que su valor es menor o igual a  $L$ , pero no se conoce su valor exacto. Según el contexto,  $L$  puede representar tiempo o un límite de detección de un instrumento de medición. Por ejemplo, en la edades de jubilación, puede darse el caso que solo se sabe la edad de la persona y que está jubilada pero no se sabe a que edad se jubiló, en este caso  $L$  es la edad de la persona jubilada. En el caso de la medición de contaminantes en un lago, puede ocurrir que el equipo de medición no detecte concentraciones bajas de ciertos metales pesados y  $L$  representa el límite de detección inferior del aparato.

**Censura por la derecha.** Una observación es censurada por la derecha en  $L$ , si se sabe que su valor es mayor o igual a  $L$ , pero su valor exacto se desconoce. Por ejemplo, que un componente eléctrico dure mas de un tiempo determinado o bien, en una lluvia intensa, puede suceder que el pluviómetro no alcance a registrar la precipitación exacta.

**Censura por intervalo.** Sucede cuando se sabe que la observación o el evento de interés ocurre en un intervalo fijo  $[L_i, L_s]$ .

Existe una vasta literatura para modelar y analizar datos censurados independientes, ver los libros de ?, ? y ?, por mencionar sólo algunos, en ellos se puede ver la estimación de curvas de sobrevivencia mediante métodos no paramétricos como el estimador de Kaplan-Meier, comparación no paramétrica de curvas de sobrevivencia, así como procedimientos paramétricos para estimar distribuciones de sobrevivencia.

## 3.2. Conceptos básicos de series de tiempo

---

Cuando los datos obtenidos de un fenómeno o experimento aleatorio presentan censura y/o datos perdidos y además se registran sucesivamente en el tiempo, estos presentan una estructura de correlación mas que de independencia, situación que sucede con mayor frecuencia en investigaciones ambientales. Por ejemplo, cuando se monitorean contaminantes en un periodo de tiempo, en el registro secuencial de la precipitación pluvial, la medición de la altitud de nubes densas en un aeropuerto, etc.; los valores exactos (o verdaderos) de las observaciones pueden estar por debajo o por encima de un límite de detección<sup>1</sup>. Para modelar este tipo de datos es común usar series de tiempo. En esta tesis, el análisis de datos censurados y/o perdidos se realiza a través de un modelo de espacio de estados.

## 3.2. Conceptos básicos de series de tiempo

Una **serie de tiempo** es un conjunto de observaciones  $y_t$ , cada una registrada en un tiempo específico  $t$ .

Las *series de tiempo discretas* son aquellas en las cuales el conjunto de tiempos  $T_0$ , en donde las observaciones se obtienen, es discreto, como en el caso cuando las observaciones se realizan en intervalos de tiempo fijos. Las *series de tiempo continuas* son aquellas cuando las observaciones son registradas continuamente sobre un intervalo de tiempo, por ejemplo,  $T_0 = [0, 1]$ .

### 3.2.1. Ejemplos de series de tiempo

En esta sección presentan algunos ejemplos de series de tiempo con observaciones censuradas y/o perdidas. Primero se presenta la serie de tiempo  $\{y_t, t = 1, 2, \dots, 154\}$  de fósforo en solución reactiva medida en miligramos por litro, mg/L, en un río que fue monitoreado de forma mensual de diciembre de 1994 a septiembre de 2007 por el Departamento de Ecología del estado de Washington, E.U., en la estación 08C070 Cedar R Logan St Renton.

En la Figura 3.1 se muestra la serie  $\{y_t, t = 1, 2, \dots, 154\}$  con círculos rellenos y los

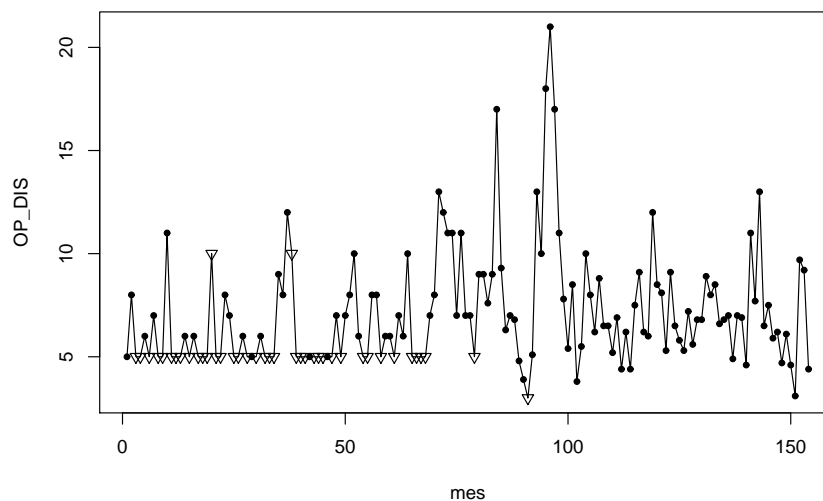
---

<sup>1</sup>El límite de detección es una cota superior o inferior en el rango de medición del instrumento.

### 3.2. Conceptos básicos de series de tiempo

---

valores no detectados (censurados) por el instrumento de medición se representan con un triángulo con pico hacia abajo, los cuales representan 26.62% en la serie. Estos datos fueron tomados del sitio de internet <http://www.ecy.wa.gov/apps/watersheds/riv/regions/state.asp>.



**Figura 3.1:** Serie de tiempo de fósforo disuelto (mg/L)

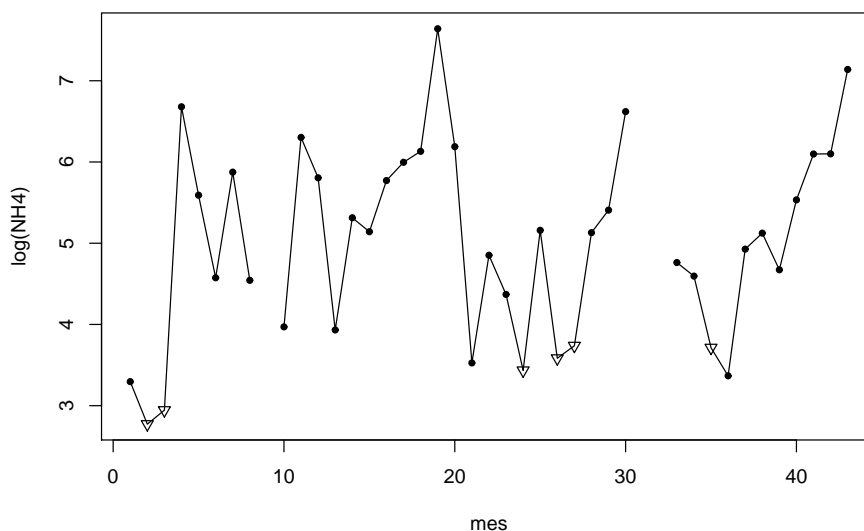
En este caso, el conjunto  $T_0$  contiene 154 observaciones. Generalmente, cuando se tienen  $n$  observaciones registradas en intervalos de tiempo igualmente espaciados, suele ser más conveniente cambiar el eje de las abscisas de tal manera que  $T_0$  sea el conjunto  $T_0 := \{1, 2, \dots, n\}$ . Para el presente ejemplo, el mes de diciembre de 1994 representa  $t = 1$ , el mes de enero de 1995 representa  $t = 2$ , y así sucesivamente hasta completar la serie.

En la Figura 3.2 se muestra gráficamente otra serie de tiempo sobre composición química de la deposición o degradación atmosférica. Particularmente las observaciones representan concentraciones mensuales de amonio ( $\text{NH}_4$ ) coleccionadas entre 1977 y 1980 en Lawrence Livermore, California, US, por The Environmental Measurements Laboratory. Estos datos fueron tomados del artículo de ?.

Existen límites de detección inferiores en las pruebas lo cuales dependen de la cantidad total de precipitación de la composición química coleccionada en cada mes, volúmenes pequeños recolectados provocan un aumento en los límites de detección. La serie de

### 3.2. Conceptos básicos de series de tiempo

tiempo consiste de 43 observaciones de los cuales 6 valores están censurados y 3 datos están perdidos, en total la serie tiene un 20.93% de información faltante. En la Figura 3.2 se muestra gráficamente los valores de la serie con círculos rellenos y los valores censurados con un triángulo con pico hacia abajo.



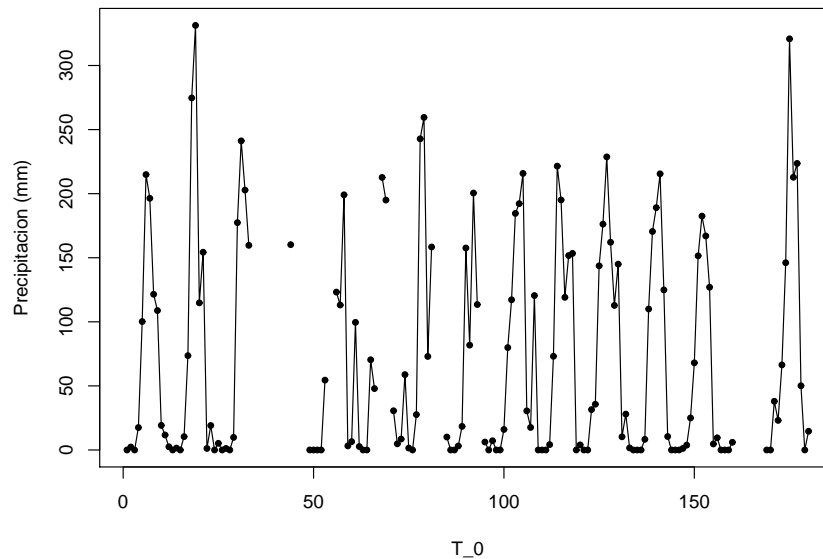
**Figura 3.2:** Concentraciones de  $\text{NH}_4$  (mequiv/sq m) en Lawrence Livermore, California, mayo de 1977 a noviembre de 1980.

En la Figura 3.3 se muestra una serie de tiempo que consiste en el promedio mensual de precipitación pluvial (en mm) en Chilapa, Gro. Los datos presentados en esta gráfica comprenden el periodo de enero de 1986 a diciembre de 2001. En este caso, el conjunto  $T_0$  contiene 180 valores. Para nuestro ejemplo, el mes de enero de 1986 representa el mes 1 ( $t=1$ ), febrero de 1986 el mes 2, y así sucesivamente. En la figura se observan algunos “huecos” en la serie los cuales representan valores perdidos y constituyen el 16.67% de las observaciones. Se puede observar también que la serie presenta un comportamiento estacional con picos en los meses de julio y agosto, y depresiones en los meses de enero y febrero.

Los ejemplos considerados en esta sección representan una muestra muy pequeña del gran número de series de tiempo que se pueden encontrar en muchos campos de las ciencias. Sin embargo, además de una descripción gráfica de los datos, el interés se centra en hacer inferencias sobre tales series, para lo cual es necesario considerar un modelo (o familia de modelos) probabilístico apropiado para los datos, ajustar el

## 3.2. Conceptos básicos de series de tiempo

---



**Figura 3.3:** Precipitación pluvial en Chilapa, Gro., en el periodo de enero de 1977 a septiembre de 1988.

modelo y checar su bondad de ajuste con el propósito de utilizarlo para entender mejor el comportamiento o mecanismo que genera la serie. El modelo desarrollado puede ser usado de diferentes maneras dependiendo del campo de aplicación. Se pueden consultar los libros de ?, ? y ? para mayores detalles.

### 3.2.2. Algunos modelos de series de tiempo

Un modelo de series de tiempo para un conjunto de datos observados  $\{y_t\}$  es la especificación de las distribuciones conjuntas (o posiblemente solo de sus medias y varianzas) de una secuencia de variables aleatorias  $\{Y_t\}$  en las cuales  $\{y_t\}$  representan una realización.

## 3.2. Conceptos básicos de series de tiempo

---

### Modelos con tendencia y estacionalidad

En varios ejemplos de series de tiempo se manifiesta una tendencia en los datos, cuando este es el caso, se sugiere tratar un modelo de la forma

$$Y_t = m_t + \varepsilon_t$$

donde  $m_t$  es el componente de tendencia del modelo y  $\varepsilon_t$  es el término de error con media cero. La tendencia puede ser lineal o cuadrática, i.e.,  $m_t = \beta_0 + \beta_1 t + \beta_2 t^2$ .

Muchas series de tiempo son influenciadas por factores de variación estacional, tales como el clima. Por ejemplo la serie de datos de lluvia en un lugar de Guerrero (Figura 3.3) muestra un patrón estacional anual con picos en los meses de julio y agosto y caídas en enero y febrero. Para el efecto estacional, sin suponer tendencia en la serie de tiempo, se puede usar el modelo

$$Y_t = s_t + \varepsilon_t$$

donde  $s_t$  es una función periódica de  $t$  con periodo  $d$ .

Sin embargo, existen series que presentan tanto componente *estacional* como de *tendencia*. Para representar datos con estas características se puede utilizar el modelo

$$Y_t = m_t + s_t + \varepsilon_t \tag{3.1}$$

donde  $m_t$  es el componente de tendencia,  $s_t$  es la componente estacional y  $\varepsilon_t$  es el término de error del modelo. El modelo en (3.1) es llamado modelo de descomposición clásica.

### Procesos estacionarios

**Definición 3.1**  $\{Y_t\}$  es una serie de tiempo estrictamente estacionaria si

$$(Y_1, \dots, Y_n)' \stackrel{d}{=} (Y_{1+h}, \dots, Y_{n+h})'$$

para todos los enteros  $h$  y  $n \geq 1$ , y “ $\stackrel{d}{=}$ ” indica que dos vectores aleatorios tienen la misma función distribución conjunta.



## 3.2. Conceptos básicos de series de tiempo

---

**Definición 3.2**  $\{Y_t\}$  es una serie de tiempo débilmente estacionaria si  $E(X_t)$  es independiente de  $t$  y  $Cov(X_t, X_{t+h})$  es independiente de  $t$  para cada  $h$ .

Algunas propiedades de una serie de tiempo estrictamente estacionaria son:

- Las variables aleatorias  $Y_t$  son idénticamente distribuidas
- $(Y_t, Y_{t+h})' \stackrel{d}{=} (Y_1, Y_{1+h})'$  para todos los enteros  $t$  y  $h$
- $\{Y_t\}$  es una serie de tiempo débilmente estacionaria si  $E(X_t^2) < \infty$  para todo  $t$
- La estacionaridad débil no implica estacionaridad estricta
- Una sucesión de variables aleatorias independientes e idénticamente distribuidas es estrictamente estacionaria.

### Modelos ARMA

Una clase importante de modelos de series de tiempo estacionarias o procesos lineales son conocidos como modelos autoregresivos de promedios móviles (ARMA, por sus siglas en inglés), que se define a continuación.

**Definición 3.3** Una serie de tiempo  $\{Y_t, t = 0, \pm 1, \pm 2, \dots\}$  es un modelo autoregresivo de promedios móviles de orden  $p$  y  $q$ , denotado por  $ARMA(p, q)$ , si  $\{Y_t\}$  es estacionario y si para todo  $t$ ,

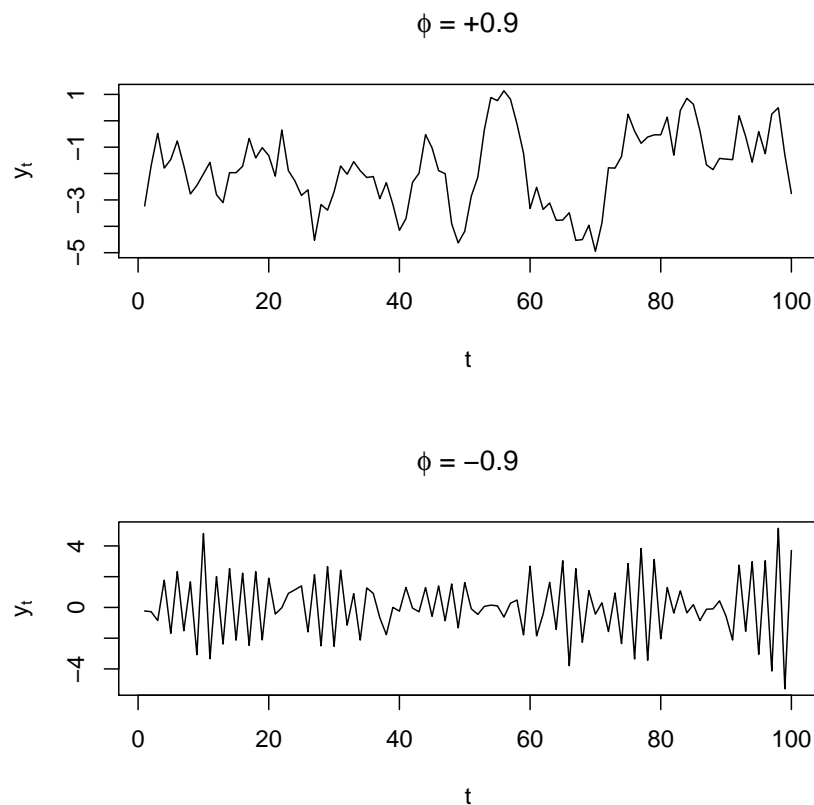
$$Y_t - \phi_1 Y_{t-1} - \dots - \phi_p Y_{t-p} = Z_t + \theta_1 Z_{t-1} + \dots + \theta_q Z_{t-q}, \quad (3.2)$$

donde  $Z_t \sim iidN(0, \sigma^2)$  y los polinomios  $(1 - \phi_1 z - \dots - \phi_p z^p)$  y  $(1 + \theta_1 z + \dots + \theta_q z^q)$  no tienen factores en común.

Si  $q = 0$  en (3.2) entonces el modelo se reduce a un modelo autoregresivo de orden  $p$ , denotado como  $AR(p)$ , estos modelos están basados en la idea de que el valor actual de una serie,  $y_t$ , es una función de los  $p$  valores anteriores,  $y_{t-1}, y_{t-2}, \dots, y_{t-p}$ . En la Figura 3.4 se muestran dos series de tiempo generadas mediante un proceso  $AR(1)$ , una con  $\phi = 0.9$  y la otra  $\phi = -0.9$ .

### 3.3. Modelos de Espacio de Estados Generalizado

---



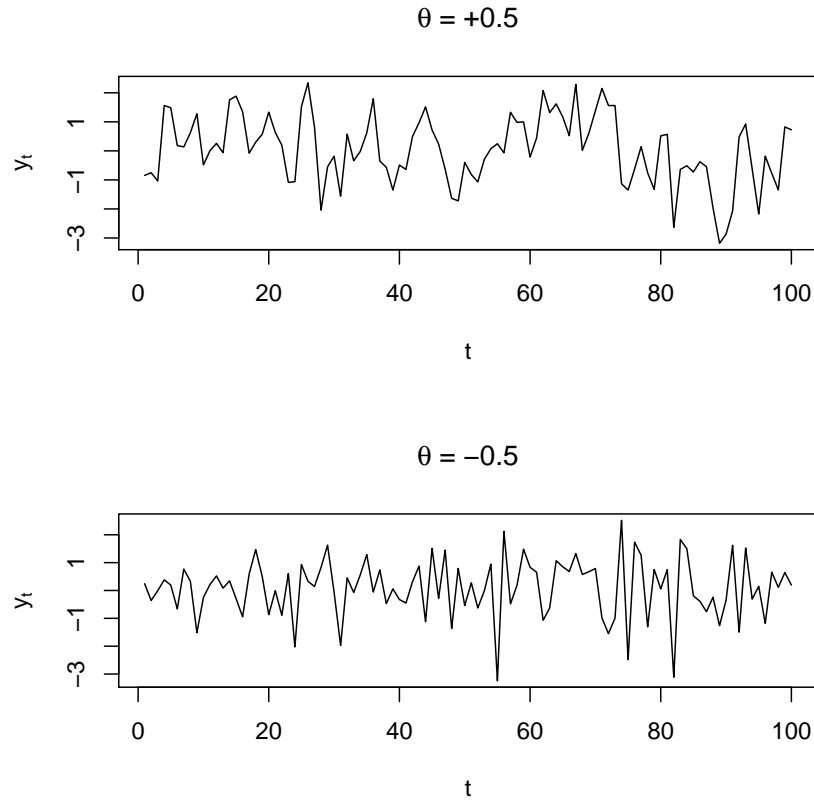
**Figura 3.4:** Realización de un modelos  $AR(1)$  con:  $\phi = 0.9$  (gráfica superior) y  $\phi = -0.9$  (gráfica inferior)

Si  $p = 0$  en (3.2) entonces se reduce a un modelo de promedios móviles de orden  $q$ , denotado como  $MA(q)$ , los cuales suponen que el valor actual,  $Y_t$ , puede ser explicado mediante una combinación lineal de los  $q$  valores anteriores de la serie  $Z_t$ . En la Figura 3.5 se presentan dos realizaciones de un modelo  $MA(1)$ , la primera, que se muestra en la gráfica superior, se realizó con  $\theta = 0.5$  y  $\sigma^2 = 1$ , en tanto que la gráfica inferior se realizó con  $\theta = -0.5$  y  $\sigma^2 = 1$ .

### 3.3. Modelos de Espacio de Estados Generalizado

Los modelos SSM y las recursiones de Kalman asociados a ellos han tenido un gran impacto en el análisis de series de tiempo en muchas áreas de interés como la física, ingeniería, economía, biología, etc. Son una clase de modelos muy rica ya que muchos

### 3.3. Modelos de Espacio de Estados Generalizado



**Figura 3.5:** realización de un modelo  $MA(1)$  con:  $\theta = 0.5$  (gráfica superior) y  $\theta = -0.5$  (gráfica inferior)

de los modelos tradicionales como los modelos ARIMA, y los modelos estructurales de series de tiempo para econometría, se pueden formular como casos especiales de un modelo SSM generalizado. Un modelo SSM para una serie de tiempo  $\{Y_t, t = 1, 2, \dots\}$  consta de dos ecuaciones: *Ecuación de Observación* y *Ecuación de Estados*. En este trabajo, se denota como  $\mathbf{y}_{1:t}$ , a el vector columna  $t$ -dimensional  $\mathbf{y}_{1:t} = (y_1, \dots, y_t)$ .

Un SSM generalizado supone que  $Y_t$  dado  $(\alpha_t, \boldsymbol{\alpha}_{1:t-1}, \mathbf{y}_{1:t-1})$  es independiente de  $(\boldsymbol{\alpha}_{1:t-1}, \mathbf{y}_{1:t-1})$ , donde,

$$f(y_t; \boldsymbol{\xi} | \alpha_t, \alpha_{t-1}, \dots, \alpha_1, y_{t-1}, \dots, y_1) = f(y_t; \boldsymbol{\xi} | \alpha_t), \quad t = 1, 2, \dots, \quad (3.3)$$

y que pertenecen a una cierta familia de distribuciones conocidas. Además, las variables de estado  $\alpha_t$ 's cumplen

$$f(\alpha_{t+1}; \boldsymbol{\psi} | \alpha_t, \alpha_{t-1}, \dots, \alpha_1, y_{t-1}, \dots, y_1) = f(\alpha_{t+1}; \boldsymbol{\psi} | \alpha_t), \quad t = 1, 2, \dots, \quad (3.4)$$

### 3.3. Modelos de Espacio de Estados Generalizado

---

donde el estado inicial  $\alpha_1$  tiene una función de densidad de probabilidad  $f_1$ .

Suponga que  $y_1, y_2, \dots, y_n$  es una realización del modelo de espacio de estados en (3.3) y (3.4), de estas ecuaciones, se sigue que la distribución conjunta de las observaciones y las variables de estado  $(\mathbf{y}, \boldsymbol{\alpha}) := (y_1, y_2, \dots, y_n, \alpha_1, \alpha_2, \dots, \alpha_n)$  está dada por

$$\begin{aligned}
 f(y_1, \dots, y_n, \alpha_1, \dots, \alpha_n) &= f(y_n | \boldsymbol{\alpha}_n, \boldsymbol{\alpha}_{1:n-1}, \mathbf{y}_{1:n-1}) f(\boldsymbol{\alpha}_n, \boldsymbol{\alpha}_{1:n-1}, \mathbf{y}_{1:n-1}) \\
 &= f(y_n | \alpha_n) f(\alpha_n | \boldsymbol{\alpha}_{1:n-1}, \mathbf{y}_{1:n-1}) f(\boldsymbol{\alpha}_{1:n-1}, \mathbf{y}_{1:n-1}) \\
 &= f(y_n | \alpha_n) f(\alpha_n | \alpha_{n-1}) f(\boldsymbol{\alpha}_{1:n-1}, \mathbf{y}_{1:n-1}) \\
 &\quad \vdots \\
 &= \left( \prod_{j=1}^n f(y_j | \alpha_j) \right) \left( \prod_{j=2}^n f(\alpha_j | \alpha_{j-1}) \right) f_1(\alpha_1)
 \end{aligned}$$

Por otro lado, de (3.4) implica que  $\{\alpha_t\}$  cumple con la propiedad de Markov, por tanto,

$$f(y_1, \dots, y_n | \alpha_1, \dots, \alpha_n) = \prod_{j=1}^n f(y_j | \alpha_j). \quad (3.5)$$

De la expresión anterior se puede ver que  $Y_1, Y_2, \dots, Y_n$  son condicionalmente independientes dadas las variables de estado  $\alpha_1, \alpha_2, \dots, \alpha_n$ , de tal manera que la estructura de dependencia de las  $Y$ 's es inducida por el proceso de estados  $\{\alpha_t\}$ , (?). La sucesión de variables de estados  $\{\alpha_t\}$  es usualmente llamada proceso “oculto” o “latente” asociado con el proceso observado.

#### 3.3.1. Modelo de Espacio de Estados Lineal

En este caso, la ecuación de observación expresa a la observación  $Y_t$  como una función lineal de la variable de estados mas un error aleatorio; es decir,

$$Y_t = \mathbf{g}_t^T \boldsymbol{\alpha}_t + w_t; \quad t = 1, 2, \dots \quad (3.6)$$

donde  $\mathbf{g}_t$  es un vector (posiblemente desconocido) de tamaño  $m \times 1$ , donde  $m$  es un entero positivo y  $w_t$  es un error aleatorio con media cero y varianza  $\sigma_{w_t}^2$ .

El vector de estados  $\boldsymbol{\alpha}_t$  es un vector no observable; sin embargo, se supone que se

### 3.3. Modelos de Espacio de Estados Generalizado

---

conoce como cambia  $\alpha_t$  a través del tiempo mediante una expresión que expresa el estado futuro  $\alpha_{t+1}$  en el tiempo  $t + 1$  en función del estado anterior  $\alpha_t$  y un término de error, como sigue

$$\alpha_{t+1} = F_t \alpha_t + \mathbf{v}_t; \quad t = 1, 2, \dots \quad (3.7)$$

donde  $F_t$  es una matriz de  $(m \times m)$  y  $\mathbf{v}_t$  es el vector de errores que sigue una distribución multivariada (en muchos casos normal) con vector de medias cero y matriz de varianzas y covarianzas  $V_t$ . Las secuencias  $\{w_t\}$  y  $\{v_t\}$  no están correlacionadas y en muchos casos especiales, por ejemplo en los modelos estacionarios,  $\mathbf{g}_t$ ,  $F_t$ ,  $\sigma_{w_t}^2$  y  $V_t$  no dependen de  $t$ . Es claro que de la definición, ni  $\{Y_t\}$  ni  $\{\alpha_t\}$  deben ser necesariamente estacionarios. Además, si la secuencia  $\alpha_1, \mathbf{v}_1, \mathbf{v}_2, \dots$  es independiente, entonces  $\{\alpha_t\}$  cumple la propiedad de Markov (?).

Las dos ecuaciones (3.6) y (3.7) constituyen la forma general de un modelo de espacio de estados lineal univariado. Sin embargo, el SSM se puede generalizar al caso en donde  $Y_t$  en (3.6) sea un vector, haciendo a  $\mathbf{g}_t$  una matriz de tamaño apropiada y a  $w_t$  un vector de longitud apropiado. Es posible agregar alguna combinación lineal de variables explicatorias en el lado derecho de (3.6).

Es posible encontrar representaciones de espacio de estados para un gran número de modelos de series de tiempo. Por ejemplo, una representación de espacio de estados para un modelo  $AR(1)$  dado por

$$Y_t = \phi Y_{t-1} + Z_t; \quad Z_t \sim iid N(0, \sigma^2),$$

se da, definiendo la secuencia de variables de estado  $\alpha_t$  como

$$\alpha_{t+1} = \phi \alpha_t + v_t; \quad t = 1, 2, \dots,$$

donde  $\alpha_1 = Y_1 = \sum_{j=0}^{\infty} \phi^j Z_{1-j}$  y  $v_t = Z_{t+1}$ . La ecuación de observación para la serie de tiempo  $\{Y_t\}$  es

$$Y_t = \alpha_t.$$

Para ver mas ejemplos sobre diferentes modelos de series de tiempo en forma de SSM se puede consultar a ?, ?, ?, ?, por mencionar algunos.

### 3.3. Modelos de Espacio de Estados Generalizado

---

#### 3.3.2. Recursiones de Kalman

En 1960, R. E. Kalman (?) publicó un artículo donde presentó un algoritmo recursivo para solucionar el problema del filtro lineal de datos discretos. Desde ese momento y debido al avance computacional, las recursiones de Kalman han sido objeto de investigación y aplicación en diferentes disciplinas, principalmente en navegación. El filtro de Kalman es un conjunto de ecuaciones matemáticas que proporcionan un recurso computacional eficiente para estimar el estado de un proceso de una manera que minimiza el error cuadrático medio.

En el análisis de un SSM, el principal objetivo es encontrar al mejor estimador lineal (en el sentido de menor error cuadrático medio) del vector de estados  $\alpha_t$  en términos de las observaciones  $Y_1, Y_2, \dots, Y_n$ , y la recursiones de Kalman proveen de un método general para realizar esto. La estimación de  $\alpha_t$  en términos de:

1.  $Y_0, Y_1, \dots, Y_{t-1}$  se le conoce como *predicción*.
2.  $Y_0, Y_1, \dots, Y_t$  se le conoce como *filtro*.
3.  $Y_0, Y_1, \dots, Y_n$  se le conoce como *suavizamiento*.

cada uno de estos problemas puede ser resuelto usando un procedimiento recursivo apropiado para calcular estimadores óptimos del vector de estados  $\alpha_t$  que consiste básicamente en ir actualizando el estimador de  $\alpha_t$  cuando se tiene una nueva observación  $y_t$ . A continuación se describen los procedimientos para los problemas de filtro, predicción y suavizamiento de Kalman.

#### Filtro de Kalman

Bajo un SSM generalizado definido por (3.3) y (3.4), para el filtro de Kalman es necesario determinar la densidad condicional  $f(\alpha_t | \mathbf{y}_{1:t})$ . Usando el teorema de Bayes

### 3.3. Modelos de Espacio de Estados Generalizado

---

y (3.3) se obtiene que

$$\begin{aligned}
 f(\alpha_t | \mathbf{y}_{1:t}) &= \frac{f(\alpha_t, \mathbf{y}_{1:t})}{f(\mathbf{y}_{1:t})} \\
 &= \frac{f(\alpha_t, y_t, \mathbf{y}_{1:t-1})}{f(\mathbf{y}_{1:t})} \\
 &= \frac{f(y_t | \alpha_t, \mathbf{y}_{1:t-1}) f(\alpha_t, \mathbf{y}_{1:t-1})}{f(\mathbf{y}_{1:t})} \\
 &= \frac{f(y_t | \alpha_t) f(\alpha_t | \mathbf{y}_{1:t-1}) f(\mathbf{y}_{1:t-1})}{f(\mathbf{y}_{1:t})} \\
 &= \frac{f(y_t | \alpha_t) f(\alpha_t | \mathbf{y}_{1:t-1})}{f(y_t | \mathbf{y}_{1:t-1})} \\
 &\propto f(y_t | \alpha_t) f(\alpha_t | \mathbf{y}_{1:t-1}), \tag{3.8}
 \end{aligned}$$

El término  $f(y_t | \mathbf{y}_{1:t-1})$  que aparece en (3.8) es justamente un factor de escala determinado por la condición de que  $f(\alpha_t | \mathbf{y}_{1:t})$  es una densidad, es decir,  $\int_{-\infty}^{\infty} f(\alpha_t | \mathbf{y}_{1:t}) d\alpha_t = 1$ .

Para el caso de un SSM lineal, el estimador del filtro de Kalman  $\alpha_{t|t} := P_t(\boldsymbol{\theta}_t)$ , y la matriz de varianzas y covarianzas  $\Omega_{t|t} = E[(\boldsymbol{\alpha}_t - \alpha_{t|t})(\boldsymbol{\alpha}_t - \alpha_{t|t})']$  del error, son determinadas por las relaciones:

$$P_t(\boldsymbol{\alpha}_t) = \alpha_{t|t} = \hat{\boldsymbol{\alpha}}_t + \Omega_t \mathbf{g}_t \delta_t^{-1} (Y_t - \mathbf{g}_t' \hat{\boldsymbol{\alpha}}_t), \tag{3.9}$$

$$\Omega_{t|t} = \Omega_t - \Omega_t \mathbf{g}_t \delta_t^{-1} \mathbf{g}_t' \Omega_t'. \tag{3.10}$$

### 3.3. Modelos de Espacio de Estados Generalizado

---

#### Predicciones de Kalman

Las predicciones de Kalman de paso para el modelo (3.3)-(3.4), se obtienen determinando la densidad condicional  $f(\alpha_{t+1}|\mathbf{y}_{1:t})$  como sigue

$$\begin{aligned}
 f(\alpha_{t+1}|\mathbf{y}_{1:t}) &= \frac{f(\alpha_{t+1}, \mathbf{y}_{1:t})}{f(\mathbf{y}_{1:t})} \\
 &= \left\{ \int_{-\infty}^{\infty} f(\alpha_{t+1}, \alpha_t, \mathbf{y}_{1:t}) d\alpha_t \right\} / f(\mathbf{y}_{1:t}) \\
 &= \left\{ \int_{-\infty}^{\infty} f(\alpha_{t+1}|\alpha_t, \mathbf{y}_{1:t}) f(\alpha_t, \mathbf{y}_{1:t}) d\alpha_t \right\} / f(\mathbf{y}_{1:t}) \\
 &= \int_{-\infty}^{\infty} f(\alpha_{t+1}|\alpha_t) f(\alpha_t|\mathbf{y}_{1:t}) d\alpha_t. \tag{3.11}
 \end{aligned}$$

La condición inicial para resolver estas recursiones es  $f(\alpha_1|\mathbf{y}_{1:0}) := f_1(\alpha_1)$ . Para el modelo SSM definido por (3.6) y (3.7), las predicciones de un paso,  $\hat{\boldsymbol{\alpha}}_t := P_{t-1}(\boldsymbol{\alpha}_t)$  y la matriz de varianzas y covarianzas  $\Omega_t = E[(\boldsymbol{\alpha}_t - \hat{\boldsymbol{\alpha}}_t)(\boldsymbol{\alpha}_t - \hat{\boldsymbol{\alpha}}_t)']$  de los errores  $t=1,2,\dots$ , están determinadas por las recursiones

$$\hat{\boldsymbol{\alpha}}_{t+1} = F_t \hat{\boldsymbol{\alpha}}_t + \boldsymbol{\theta}_t \delta_t^{-1} (Y_t - \mathbf{g}_t' \hat{\boldsymbol{\alpha}}_t), \tag{3.12}$$

$$\Omega_{t+1} = F_t \Omega_t F_t' + V_t - \boldsymbol{\theta}_t \delta_t^{-1} \boldsymbol{\theta}_t', \tag{3.13}$$

donde

$$\begin{aligned}
 \delta_t &= \mathbf{g}_t' \Omega_t \mathbf{g}_t + \sigma_{w_t}^2, \\
 \boldsymbol{\theta}_t &= F_t \Omega_t \mathbf{g}_t.
 \end{aligned}$$



### 3.3. Modelos de Espacio de Estados Generalizado

---

#### Suavizamiento de Kalman

La distribución de suavizamiento para  $\alpha_t$  se obtiene como sigue:

$$\begin{aligned}
 f(\alpha_t | \mathbf{y}_{1:n}) &= \int f(\alpha_{t+1}, \alpha_t | \mathbf{y}_{1:n}) d\alpha_{t+1} \\
 &= \int f(\alpha_{t+1} | \mathbf{y}_{1:n}) f(\alpha_t | \alpha_{t+1}, \mathbf{y}_{1:n}) d\alpha_{t+1} \\
 &= \int f(\alpha_{t+1} | \mathbf{y}_{1:n}) f(\alpha_t | \alpha_{t+1}, \mathbf{y}_{1:t}) d\alpha_{t+1} \\
 &= f(\alpha_t | \mathbf{y}_{1:t}) \int \frac{f(\alpha_{t+1} | \mathbf{y}_{1:n}) f(\alpha_{t+1} | \alpha_t)}{f(\alpha_{t+1} | \mathbf{y}_{1:t})} d\alpha_{t+1} \quad (3.14)
 \end{aligned}$$

Note que para calcular (3.14), es necesario calcular la distribuciones de predicción y de filtro en (3.11) y (3.8) respectivamente, para  $t = 1, 2, \dots, n$ , una vez obtenidas, se realizan las recursiones hacia atrás para calcular las distribuciones de suavizamiento  $f(\alpha_t | \mathbf{y}_{1:n})$  para  $t = n - 1, n - 2, \dots, 1$ .

Para el caso lineal, el estimador suavizado de Kalman  $\alpha_{t|n} := P_n(\theta_t)$ , y su matriz de varianzas y covarianzas del error  $\Omega_{t|n} = E[(\alpha_t - \alpha_{t|n})(\alpha_t - \alpha_{t|n})']$  son determinadas, para un valor fijo  $t$  ( $n < t$ ), por las siguientes recursiones:

$$P_n(\alpha_t) = \alpha_{t|n} = P_{n-1}(\alpha_t) + \Omega_{t,n} \mathbf{g}_n \delta_n^{-1} (Y_n - \mathbf{g}_n' \hat{\alpha}_n), \quad (3.15)$$

$$\Omega_{t,n+1} = \Omega_{t,n} [F_n - \theta_n \delta_n^{-1} \mathbf{g}_n]', \quad (3.16)$$

$$\Omega_{t|n} = \Omega_{t|n-1} - \Omega_{t,n} \mathbf{g}_n \delta_n^{-1} \mathbf{g}_n' \Omega_{t,n}', \quad (3.17)$$

con las condiciones iniciales  $P_{t-1}(\alpha_t) = \hat{\alpha}_t$  y  $\Omega_{t,t} = \Omega_{t|t-1} = \Omega_t$ .

#### 3.3.3. Estimación en el modelo SSM Generalizado

La distribución conjunta de  $Y_1, \dots, Y_n$  o función de verosimilitud para una realización  $y_1, y_2, \dots, y_n$  de del modelo SSM Generalizado en (3.3)-(3.4) está dada por

$$L(\theta; y_1, y_2, \dots, y_n) = f(y_1, \dots, y_n) = f_{Y_1}(y_1) \prod_{t=2}^n f(y_t | \mathbf{y}_{1:t-1}). \quad (3.18)$$

### 3.3. Modelos de Espacio de Estados Generalizado

---

donde

$$f(y_t|\mathbf{y}_{1:t-1}) = \int_{-\infty}^{\infty} f(y_t|\alpha_t)f(\alpha_t|\mathbf{y}_{1:t-1})d\alpha_t.$$

Entonces, el estimador de máxima verosimilitud (EMV) de  $\boldsymbol{\theta}$  se obtiene como

$$\hat{\boldsymbol{\theta}} = \arg \max_{\boldsymbol{\theta} \in \Theta} L(\boldsymbol{\theta}; y_1, y_2, \dots, y_n).$$

#### Estimación de Modelos de Espacio de Estados lineal

Suponga que el modelo de espacio de estados definido por la ecuaciones (3.6) y (3.7), está parametrizado por el vector  $\boldsymbol{\theta}$ . La función de verosimilitud para una realización  $y_1, y_2, \dots, y_n$  de este modelo con respecto a  $\boldsymbol{\theta}$ , está dada por

$$\begin{aligned} L(\boldsymbol{\theta}; y_1, y_2, \dots, y_n) &= f_{Y_1, \dots, Y_n}(y_1, \dots, y_n; \boldsymbol{\theta}) \\ &= \prod_{t=1}^n f_{Y_t|Y_{t-1}, \dots, Y_0}(y_t|y_{t-1}, \dots, y_0) \end{aligned} \quad (3.19)$$

donde  $f_{Y_t|Y_{t-1}, \dots, Y_0}(y_t|y_{t-1}, \dots, y_0)$  es la función de densidad condicional de  $Y_t$  dado  $Y_{t-1} = y_{t-1}, \dots, Y_0 = y_0$ .

Asumiendo que los errores  $w_t$  y  $v_t$ ,  $t = 1, 2, \dots$ , de las ecuaciones de observación y de estados respectivamente, tienen distribución normal, la tarea de calcular la verosimilitud se simplifica de manera significativa. Cuando este es el caso, se tiene un modelo de espacio de estados Gaussiano.

Si  $Y_0$ ,  $\boldsymbol{\alpha}_1$  y  $w_t$ ,  $\mathbf{v}_t$ ,  $t = 1, 2, \dots$ , tienen distribución normal, entonces

$$f_{Y_t|Y_{t-1}, \dots, Y_0}(y_t|y_{t-1}, \dots, y_0) = \frac{1}{\sqrt{2\pi}} \delta_t^{-1/2} \exp \left\{ -\frac{1}{2} \frac{(y_t - \mathbf{g}'_t \hat{\boldsymbol{\alpha}}_t)^2}{\delta_t} \right\}$$

donde  $\hat{\boldsymbol{\alpha}}_t$ ,  $\delta_t$ ,  $t = 1, 2, \dots, n$  son las predicciones de un paso y las varianzas del error de predicción, respectivamente, las cuales se obtienen de las recursiones de Kalman. Así,  $L(\boldsymbol{\theta}; y_1, y_2, \dots, y_n)$  se puede expresar como

$$L(\boldsymbol{\theta}; y_1, y_2, \dots, y_n) = \left( \frac{1}{\sqrt{2\pi}} \right)^n \left( \prod_{t=1}^n \delta_t \right)^{-1/2} \exp \left\{ \frac{1}{2} \sum_{t=1}^n \frac{(y_t - \mathbf{g}'_t \hat{\boldsymbol{\alpha}}_t)^2}{\delta_t} \right\}. \quad (3.20)$$

### 3.4. Algoritmo EM

---

Entonces, el estimador de máxima verosimilitud (EMV) de  $\theta$  se obtiene como

$$\hat{\theta} = \arg \max_{\theta \in \Theta} L(\theta; y_1, y_2, \dots, y_n).$$

Para encontrar el EMV de  $\theta$  se utiliza un algoritmo de optimización no lineal.

### 3.4. Algoritmo EM

El algoritmo Esperanza Maximización (EM) es un procedimiento matemático iterativo para calcular el estimador de máxima verosimilitud (EMV) en problemas donde la función de verosimilitud es muy compleja y no se pueden utilizar métodos estándar de cálculo para encontrar el valor del parámetro que haga máxima la función de verosimilitud. Tal situación, se presenta más comúnmente cuando en la muestra hay datos censurados y/o perdidos. Dicho algoritmo, propuesto por ?, consiguió formalizar y justificar una idea que habían explorado otros investigadores y ha sido utilizado y mejorado por muchos más autores.

La idea del algoritmo es simple, dada una primera estimación de los parámetros, se predicen los datos faltantes, se re-estiman los parámetros y continúa iterando hasta que el procedimiento converja. Este procedimiento es llamado *principio de datos aumentados* (?), que se establece de la forma siguiente: Aumentar los datos observados,  $y$ , con datos latentes,  $z$ , de tal manera que la distribución final aumentada,  $f(\theta|y, z)$ , sea “simple”. Hacer uso de esta simplicidad para maximizar, calcular o muestrear de la distribución final observada,  $f(\theta|y)$ .

En general, en la situación de datos aumentados, se puede utilizar la notación siguiente. Sea  $\mathbf{y}$  el vector de datos observados con función de densidad  $f(\mathbf{y}; \theta)$  y  $\Omega_Y$  su correspondiente espacio muestral. Sea  $\mathbf{x}$  el vector de datos completos, no observado directamente, perteneciente al espacio muestral  $\Omega_X$  y cuya función de densidad está dada por  $f(\mathbf{x}; \theta)$ . Considere una función sobreyectiva  $\psi : \Omega_X \rightarrow \Omega_Y$  que relaciona los datos observados con los completos y sea  $\Omega_X(\mathbf{y}) \subset \Omega_X$  el conjunto de puntos cuya imagen según  $\psi$  son  $\mathbf{y}$ . Entonces la función de densidad de los datos observados  $\mathbf{y}$  es:

$$f(\mathbf{y}; \theta) = \int_{\Omega_X(\mathbf{y})} f(\mathbf{x}; \theta) d\mathbf{x}.$$

### 3.4. Algoritmo EM

---

Más específicamente, el algoritmo EM consiste de dos pasos que se realizan en forma iterativa: el **Paso E** (de Esperanza) y el **paso M** (de Maximización). Para iniciar, considere  $\theta^{(i)}$  como el valor de una estimación de  $\theta$  que maximiza a  $f(\mathbf{y}; \theta)$  en la  $i$ -ésima iteración del algoritmo, ( $i = 1, 2, \dots$ ) y sea  $f(\mathbf{x}; \theta)$  la verosimilitud de los datos completos, la cual corresponde a la observación de los datos completos  $\mathbf{x} = (\mathbf{y}, \mathbf{z})$ . Entonces los pasos son:

**Paso E.** Calcular

$$Q(\theta; \theta^{(i)}) = E(\ln f(\mathbf{x}; \theta) | \mathbf{y}, \theta^{(i)}) \quad (3.21)$$

donde la esperanza es con respecto a la densidad condicional,  $f(\mathbf{z} | \mathbf{y}; \theta)$ , de los datos perdidos dados los datos observados.

**Paso M.** Maximizar  $Q(\theta; \theta^{(i)})$  con respecto a  $\theta$  para obtener  $\theta^{(i+1)}$ , *i.e.*,

$$\theta^{(i+1)} = \arg \max_{\theta} Q(\theta; \theta^{(i)}).$$

La regla de parada de este proceso iterativo puede basarse en una distancia lo suficientemente pequeña entre  $\theta^{(i+1)}$  y  $\theta^{(i)}$  o bien entre  $Q(\theta^{(i+1)}; \theta^{(i)})$  y  $Q(\theta^{(i)}; \theta^{(i)})$ .

Esta técnica proporciona buenos resultados cuando la distribución de los datos completos,  $f(\mathbf{x}; \theta)$  es más simple que la de los datos observados  $f(\mathbf{y}; \theta)$ . Sin embargo, en ocasiones la implementación del algoritmo puede presentar dificultades en el cálculo tanto de la esperanza, que incluso puede no existir, como la maximización de  $Q(\theta; \theta^{(i)})$  con respecto a  $\theta$ , ya que puede no existir una expresión explícita para el estimador. Incluso tal maximización puede evitarse haciendo uso del algoritmo EM Generalizado, ver ? para mayores detalles.

En cada iteración del algoritmo el valor de  $Q(\theta; \theta^{(i)})$  se incrementa, *i.e.*,  $Q(\theta^{(i+1)}; \theta^{(i)}) \geq Q(\theta^{(i)}; \theta^{(i)})$ ,  $i = 1, 2, \dots$ ; ? menciona que las iteraciones  $\theta^{(i)}$  convergen a un punto estacionario de  $f(\mathbf{y}; \theta)$ . Si  $f(\mathbf{y}; \theta)$  es unimodal, entonces el algoritmo converge al máximo global; sin embargo, cuando existen varios máximos locales o puntos silla en la función, el algoritmo puede no converger al máximo global. En ? se presenta el siguiente teorema sobre la convergencia del algoritmo EM a un punto estacionario.

**Teorema 3.1** *Si la esperanza de la verosimilitud de los datos completos  $Q(\theta; \theta_0)$  es continua tanto en  $\theta$  como en  $\theta_0$ , entonces el límite de una secuencia EM  $\{\hat{\theta}^{(j)}\}$  es*

### 3.4. Algoritmo EM

---

un punto estacionario de  $L(\mathbf{y}; \theta)$ , y  $L(\mathbf{y}; \hat{\theta}^{(j)})$  converge monótonamente a  $L(\mathbf{y}; \hat{\theta})$  para algún punto estacionario  $\hat{\theta}$ .

En el teorema anterior, se menciona que la convergencia se garantiza a un punto estacionario, el cual puede ser un máximo global, máximos locales o un punto silla en la función. En la práctica es muy usual utilizar algunos métodos para tratar de asegurar que se ha encontrado un máximo global, por ejemplo, métodos gráficos, correr el algoritmo utilizando diferentes puntos iniciales seleccionados aleatoriamente o técnicas mas elaboradas como *simulated annealing*.

#### 3.4.1. Algoritmo EM Monte Carlo (MCEM)

Cuando el paso E es difícil de obtener, se puede realizar una implementación Monte Carlo para calcular la función  $Q(\theta; \theta^{(j)})$  en el algoritmo EM. Este procedimiento se realiza simulando muestras de la distribución condicional  $f(\mathbf{z}|\mathbf{y}; \theta)$  de modo que la esperanza en (3.21) se puede obtener usando integración monte carlo. Se denominan esta implementación algoritmo EM Monte Carlo (MCEM, por su siglas en inglés). Brevemente este procedimiento es como sigue:

Dada una estimación  $\theta^{(i)}$  de  $\theta$ ,

1. Generar  $\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_m$  i.i.d.  $\sim f(\mathbf{z}|\mathbf{y}, \theta^{(i)})$
2. Calcular

$$\hat{Q}(\theta, \theta^{(i)}) = \frac{1}{m} \sum_{j=1}^m \ln f(\mathbf{z}_j, \mathbf{y}; \theta). \quad (3.22)$$

El paso M consiste en maximizar (3.22) para generar un nuevo estimador y el procedimiento se itera hasta alcanzar un criterio de convergencia establecido.

Cuando  $m$  tiende a infinito,  $\hat{Q}_{i+1}(\theta, \theta^{(i)})$  converge a  $Q_{i+1}(\theta, \theta^{(i)})$  con probabilidad 1 (?). Se establecen la convergencia casi segura de este algoritmo y estudian la razón de convergencia.

Se deben tomar en cuenta dos aspectos importantes en la implementación del algoritmo MCEM, la especificación de  $m$ , y el monitoreo de la convergencia de  $\{\theta^{(i)}, i =$

### 3.4. Algoritmo EM

---

$1, 2, \dots\}$ . ? mencionan que es ineficiente iniciar con valores grandes de  $m$  cuando  $\theta^{(i)}$  está lejos del valor verdadero  $\theta^*$  y recomiendan iniciar el algoritmo con valores pequeños de  $m$  e incrementarlo conforme las aproximaciones se acercan al valor verdadero. La convergencia del algoritmo puede ser monitoreada graficando la pareja de valores  $(\theta^{(i)}, i), i = 1, 2, 3, \dots$ . Después de una cantidad de iteraciones los valores de  $\theta^{(i)}$  se estabilizarán alrededor del valor verdadero,  $\theta^*$ . En ese momento se puede detener el algoritmo o bien continuar con un valor de  $m$  mas grande y reducir la variabilidad del proceso.

#### 3.4.2. Algoritmo EM Estocástico

En la sección anterior, se presentó el algoritmo MCEM, el cual se implementa cuando el paso E del algoritmo EM es intratable; sin embargo, este procedimiento requiere de cálculos computacionales altamente demandantes. ? sugieren el algoritmo EM Estocástico (SEM, por sus siglas en inglés), el cual ha sido estudiado por ?, ?, entre otros. La idea principal del algoritmo SEM es sustituir el paso E, con una sola simulación de la distribución condicional de los valores perdidos dados los observados usando el valor del parámetro de la iteración anterior, de esta forma se tiene una muestra *pseudo*-completa de los datos y entonces se puede maximizar la función de verosimilitud de los datos completos para encontrar el EMV del parámetro. El resultado de este proceso es una cadena de Markov, indexada por el número de iteraciones, la cual puede ser promediada para producir una estimación puntual del parámetro.

Brevemente, este algoritmo consta de dos pasos que se realizan de forma iterativa: el Paso-S, donde los valores censurados y/o perdidos son reemplazados por valores “simulados”, dados los valores observados y el valor  $\theta^{(i-1)}$ , el paso M, donde  $\theta^{(i)}$  es el EMV del modelo completo obtenido. Este proceso alternado del paso-S y el paso-M genera una cadena de Markov,  $\{\theta^{(i)}, i = 1, 2, \dots\}$ , la cual converge a una distribución estacionaria  $\pi(\cdot)$  bajo condiciones de regularidad. ? mencionan que esta distribución está (aproximadamente) centrada en el EMV de  $\theta$ , que su varianza depende de la razón de cambio de  $\theta^{(i)}$  en las iteraciones del algoritmo EM y en muchas situaciones en donde se ha usado este algoritmo, la convergencia de  $\theta^{(i)}$  ha sido razonablemente rápida. En la práctica, es recomendable un periodo de calentamiento “burn-in” para que  $\{\theta^{(i)}\}$  pueda alcanzar un régimen estacionario.

### 3.4. Algoritmo EM

---

Se considera la media de la distribución estacionaria  $\pi(\cdot)$  como un estimador de  $\theta$ , a esta media, se le llama *estimador EM estocástico* y se denota por  $\tilde{\theta}_n$ . Para algunos ejemplos simples  $\tilde{\theta}_n$  coincide con el EMV. ? presentan algunos ejemplos en donde presentan algunos resultados sobre consistencia y normalidad asintótica del estimador  $\tilde{\theta}_n$ .

Además de  $\tilde{\theta}_n$ , se puede obtener otro estimador de  $\theta$  derivado de las iteraciones del SEM, este estimador es el punto en donde la función de log verosimilitud es mas grande en una región dada. ? mencionan que este valor es usualmente cercano al EMV en muchos propósitos prácticos sobre todo cuando la región de interés muestra un solo cluster. Sin embargo, para obtener este valor se requiere evaluar a la función de log verosimilitud en cada iteración.

#### 3.4.3. Varianza del estimador

El algoritmo EM no genera estimadores para la matriz de varianzas y covarianzas de los estimadores; sin embargo varios autores han trabajado tratando de dar soluciones para obtener esta matriz. Una de tales soluciones es la hecha por ?, la cual es simple y muy útil (?).

Si  $\hat{\theta}$  es un estimador de  $\theta$ , entonces la varianza de  $\hat{\theta}$  puede ser estimada como la inversa de la matriz de información de Fisher observada evaluada en  $\theta = \hat{\theta}$ , *i.e.*,

$$Var(\hat{\theta}) \approx \left[ \frac{\partial^2 \log L(\mathbf{y}; \theta)}{\partial \theta^2} \right]^{-1} \quad (3.23)$$

Calcular directamente (3.23) suele ser no conveniente ya que involucra integrales múltiples sobre  $f(\mathbf{z}|\mathbf{y}, \theta)$ . Sin embargo, el calculo se puede obtener usando la identidad de Louis (?), la cual relaciona la log verosimilitud de los datos observados y la log verosimilitud de los datos completos, como se muestra a continuación,

$$\frac{\partial^2 \log L(\mathbf{y}; \theta)}{\partial \theta^2} = E \left( \frac{\partial^2 \log L(\mathbf{y}, \mathbf{z}; \theta)}{\partial \theta^2} \right) + Var \left( \frac{\partial \log L(\mathbf{y}, \mathbf{z}; \theta)}{\partial \theta} \right) \quad (3.24)$$

dicha expresión se obtiene usando el *principio de información perdida* (Louis, 1982; citado por ?):

### 3.5. Muestreo de Importancia

---

Información Observada = Información completa - Información perdida.

La ventaja de la expresión en (3.24) es que solo involucra a la distribución de los datos completos, la cual es frecuentemente una distribución razonablemente fácil de trabajar. La desventaja es que las derivadas pueden ser difíciles de calcular.

En el contexto del algoritmo MCEM y el algoritmo SEM, dada una muestra de  $f(\mathbf{z}|\mathbf{y}; \hat{\theta})$  una buena aproximación a la matriz de información de Fisher observada en (3.24), está dada por una evaluación Monte Carlo,

$$\begin{aligned} \frac{\partial^2 \log L(\mathbf{y}; \theta)}{\partial \theta^2} &= \frac{1}{m} \sum_{j=1}^m \frac{\partial^2 \log L(\mathbf{y}, \mathbf{z}^{(j)}; \theta)}{\partial \theta^2} \Big|_{\theta=\hat{\theta}} \\ &\quad + \frac{1}{m} \sum_{j=1}^m \left( \frac{\partial \log L(\mathbf{y}, \mathbf{z}^{(j)}; \theta)}{\partial \theta} \Big|_{\theta=\hat{\theta}} \right)^2 \\ &\quad - \left[ \frac{1}{m} \sum_{j=1}^m \frac{\partial \log L(\mathbf{y}, \mathbf{z}^{(j)}; \theta)}{\partial \theta} \Big|_{\theta=\hat{\theta}} \right]^2 \end{aligned} \quad (3.25)$$

donde  $\{\mathbf{z}^{(j)}\}$ ,  $j = 1, 2, \dots, m$  son generadas de la distribución condicional de los datos perdidos dado los observados,  $f(\mathbf{z}|\mathbf{y}; \hat{\theta})$ .

### 3.5. Muestreo de Importancia

El método de *muestreo de importancia* es una técnica general para estimar propiedades importantes de alguna distribución en particular y está basado en funciones de importancia; es decir, tiene que ver con la determinación y el uso de una función de densidad alterna.

Considere el problema de calcular la integral

$$J(y) = \int f(y|x)g(x) dx \quad (3.26)$$

donde  $y$  y  $x$  pueden ser vectores. Sea  $h(x)$  una densidad de la cual se puede simular una muestra fácilmente y que se “parece” a  $g(x)$ . Entonces, el método de muestreo de importancia aproxima a (3.26) generando una muestra  $x_1, x_2, \dots, x_m$  de  $h(x)$  y se



### 3.5. Muestreo de Importancia

---

estima a (3.26) como

$$\tilde{J}(y) = \frac{\sum_{i=1}^m w_i f(y|x_i)}{\sum_{i=1}^m w_i}; \text{ donde } w_i = \frac{g(x_i)}{h(x_i)}$$

Este método está basado en una representación alternativa de (3.26), expresada como

$$J(y) = \int f(y|x) \frac{g(x)}{h(x)} h(x) dx$$

la cual es llamada la *identidad fundamental de muestreo de importancia*.

Para calcular adecuadamente a  $J(y)$  dada una muestra de  $h(x)$ , el muestreo de importancia da mayor peso a regiones donde  $h(x) < g(x)$  y poco en donde  $h(x) > g(x)$ , (?).

Si se cumple que

- El soporte de  $h(x)$  incluye o abarca el soporte de  $g(x)$ ,
- Las  $x$ 's son una muestra *iid* de  $h(x)$ ,
- $J(y)$  existe y es finita,

entonces,

$$\tilde{J}(y) \xrightarrow{a.s} J(y)$$

La razón de convergencia depende de que tan parecida sea  $h(x)$  a  $g(x)$ . Es importante que las colas de la densidad  $h(x)$  no decaigan mas rápido que las de  $g(x)$  (?).

El error estándar Monte Carlo de  $\tilde{J}(y)$  es estimado por

$$\frac{\sqrt{\sum_{i=1}^m [f(y|x_i) - \tilde{J}(y)]^2 w_i^2}}{\sum_{i=1}^m w_i}$$

donde  $w_i = g(x_i)/h(x_i)$ . De esta manera, si  $h(x)$  aproxima “pobremente” a  $g(x)$ ,

### 3.5. Muestreo de Importancia

---

entonces el error estándar de  $\tilde{J}(y)$  se incrementa.

# Capítulo 4

## Modelo de Espacio de Estados Censurado

En este capítulo se proponen algunos métodos para la estimar los parámetros en un modelo SSM con observaciones censuradas (CSSM). Se ejemplifican los métodos usando un modelo SSM lineal Gaussiano con observaciones censuradas por la izquierda; sin embargo, con ligeras modificaciones, los procedimientos aquí propuestos pueden ser aplicados a otros tipos de censura.

### 4.1. El modelo CSSM

Sea  $y_1, y_2, \dots, y_n$  una realización del modelo SSM presentado en (3.3) y (3.4), y suponga que algunas de las observaciones están censuradas o perdidas. Denote a  $\mathcal{C}_t$  como la región de censura en el tiempo  $t$ ,  $t = 1, 2, \dots, n$ . En muchas situaciones prácticas, la región de censura  $\mathcal{C}_t$  está definida por el instrumento de medición, es decir, solo se observa completamente (sin censura) aquellas observaciones con valores dentro del intervalo de medición del instrumento. Por ejemplo, los instrumentos utilizados para medir las concentraciones de fósforo en solución reactiva en agua (medida en mg/L) tienen un *limite de detección* inferior de 0.01 mg/L, entonces este instrumento registra su valor límite (0.01 mg/L) cuando el valor verdadero del contaminante precede el límite de detección, por lo tanto, la región de censura  $\mathcal{C}_t$  queda definida como  $\mathcal{C}_t = (0, 0.01]$   $t = 1, \dots, n$ , es decir, aquellos valores que son menores que al límite de

## 4.1. El modelo CSSM

---

detección.

Para denotar que una observación se encuentra o no censurada, considere la variable “indicadora” de censura  $\delta_t$ , definida como

$$\delta_t := \begin{cases} 1, & \text{si } y_t \notin \mathcal{C}_t, \\ 0, & \text{de otra manera,} \end{cases} \quad t = 1, 2, \dots, n. \quad (4.1)$$

Bajo este marco, se define la variable  $W_t$  como

$$W_t := \begin{cases} Y_t, & \text{si } \delta_t = 1, \\ Z_t, & \text{si } \delta_t = 0, \end{cases}$$

donde  $Z_t$  es el valor no observado de  $Y_t$ , y

$$f_{Z_t}(z_t) = \frac{f_{Y_t}(z_t)}{\int_{\mathcal{C}_t} f_{Y_t}(u_t) du_t} 1_{\mathcal{C}_t}(z_t).$$

La presencia de censura en datos correlacionados provoca dificultades en el proceso de estimación de los parámetros del modelo y no considerarla crea estimaciones sesgadas y decisiones equivocadas. Por tal motivo, se desea encontrar un estimador  $\hat{\boldsymbol{\theta}}$  del vector de parámetros  $\boldsymbol{\theta} := (\boldsymbol{\xi}, \boldsymbol{\psi})'$  que maximice la función de verosimilitud, donde  $\boldsymbol{\xi}$  y  $\boldsymbol{\psi}$  son los vectores de parámetros de la densidad condicional de  $\mathbf{y}$  dado  $\boldsymbol{\alpha}$  y de la densidad de  $\boldsymbol{\alpha}$  respectivamente,  $\boldsymbol{\alpha} := (\alpha_1, \dots, \alpha_n)'$ ,  $\mathbf{y}_s = (y_t : \delta_t = 1)'$  es el vector de datos observados sin censura y  $\mathbf{y}_c = (y_t : \delta_t = 0)'$  es el vector de datos censurados y/o perdidos. Entonces, la función de verosimilitud del modelo SSM dado en (3.3)-(3.4) está dada por

$$L(\boldsymbol{\theta}; \mathbf{y}_s, \boldsymbol{\delta}) = \int_{\mathbb{R}^n} \int_{\mathcal{C}} L(\boldsymbol{\theta}; \mathbf{y}_s, \mathbf{y}_c, \boldsymbol{\alpha}) d\mathbf{y}_c d\boldsymbol{\alpha} \quad (4.2)$$

donde  $\mathcal{C} = \mathcal{C}_{t_1} \times \mathcal{C}_{t_2} \times \dots \times \mathcal{C}_{t_N}$ , donde  $t_1$  es el tiempo de la primera observación censurada,  $t_2$  el de la segunda observación censurada y así sucesivamente.

De (4.2) y del supuesto de que  $Y_1, Y_2, \dots, Y_n$  son variables aleatorias condicionalmente

## 4.2. Estimación del SSMCen con el Algoritmo EM

---

independientes dadas la variables de estado  $\alpha_1, \dots, \alpha_n$  se sigue que

$$\begin{aligned} L(\boldsymbol{\theta}; \mathbf{y}_s, \boldsymbol{\alpha}, \boldsymbol{\delta}) &= f(\mathbf{y}_s | \boldsymbol{\alpha}; \boldsymbol{\xi}) f(\boldsymbol{\alpha}; \boldsymbol{\psi}) \\ &= \left( \prod_{t=1}^n f_{Y_t | \alpha_t}(y_t | \alpha_t; \boldsymbol{\xi})^{\delta_t} \left[ \int_{\mathcal{C}_t} f_{Y_t | \alpha_t}(u_t | \alpha_t; \boldsymbol{\xi}) du_t \right]^{1-\delta_t} \right) f(\boldsymbol{\alpha}; \boldsymbol{\psi}), \end{aligned} \quad (4.3)$$

donde  $\delta_t$ ,  $t = 1, 2, \dots, n$  es la variable indicadora de censura definida en (4.1),  $\mathcal{C}$  es la región de censura y  $f(\boldsymbol{\alpha}; \boldsymbol{\psi})$  es la densidad de  $\boldsymbol{\alpha}$ . Un caso común del SSM es cuando la variable de estado en (3.4) es un modelo  $AR(p)$ , *i.e.*,

$$\alpha_t = \phi_1 \alpha_{t-1} + \dots + \phi_p \alpha_{t-p} + \eta_t \quad (4.4)$$

donde  $p$  es un entero conocido,  $\eta_t \sim \text{iid } N(0, \tau^2)$ ,  $t = 1, 2, \dots, n$  y el polinomio  $(1 - \phi_1 z - \dots - \phi_p z^p)$  no tiene factores comunes.

La función de verosimilitud para los datos observados está dada por la integral

$$L(\boldsymbol{\theta}; \mathbf{y}_s, \boldsymbol{\delta}) = \int L(\boldsymbol{\theta}; \mathbf{y}_s, \boldsymbol{\alpha}, \boldsymbol{\delta}) d\boldsymbol{\alpha}, \quad (4.5)$$

donde  $L(\boldsymbol{\theta}; \mathbf{y}_s, \boldsymbol{\alpha}, \boldsymbol{\delta})$  está dada en (4.3).

De esta forma, el EMV se obtiene encontrando un valor  $\hat{\boldsymbol{\theta}}$  de  $\boldsymbol{\theta}$  que maximice a  $L(\boldsymbol{\theta}; \mathbf{y}, \boldsymbol{\delta})$ . En general, no existe una forma cerrada para tal solución ya que las integrales en (4.2) o en (4.5) pueden ser imposibles de calcularlas explícitamente. En las secciones siguientes se presentan algunas propuestas para analizar este tipo de datos usando el algoritmo EM y una aproximación a la función de verosimilitud de los datos observados mediante muestreo de importancia.

## 4.2. Estimación del SSMCen con el Algoritmo EM

El algoritmo EM (?) es un procedimiento iterativo para encontrar el máximo de una función de verosimilitud en problemas de datos incompletos. En esta sección, se usa el algoritmo EM para calcular un estimador  $\hat{\boldsymbol{\theta}}$  de  $\boldsymbol{\theta}$  en el modelo (4.2) cuando se tienen observaciones censuradas.

## 4.2. Estimación del SSMCen con el Algoritmo EM

---

Brevemente el algoritmo EM se describe como sigue: dada una primera aproximación  $\boldsymbol{\theta}^{(0)}$  de  $\boldsymbol{\theta}$ , los pasos **E** y **M** del algoritmo EM, en la  $j$ -ésima iteración, son:

1. **Paso E** (Esperanza). Calcular

$$\begin{aligned} Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(j)}) &= E(\ln f(\mathbf{z}, \mathbf{y}; \boldsymbol{\theta}) | \mathbf{y}, \boldsymbol{\delta}, \boldsymbol{\theta}^{(j)}) \\ &= \int \ln f(\mathbf{z}, \mathbf{y}; \boldsymbol{\theta}) f(\mathbf{z} | \mathbf{y}, \boldsymbol{\delta}, \boldsymbol{\theta}^{(j)}) d\mathbf{z} \end{aligned} \quad (4.6)$$

donde la esperanza es con respecto a la densidad condicional de los datos censurados y/o perdidos dados los datos observados,  $f(\mathbf{z} | \mathbf{y}; \boldsymbol{\theta}, \boldsymbol{\delta})$ .

2. **Paso M** (Maximización). Maximizar a  $Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(j)})$  con respecto a  $\boldsymbol{\theta}$  para obtener  $\boldsymbol{\theta}^{(j+1)}$ .

los pasos 1 y 2 se repiten hasta que se cumpla un criterio de convergencia dado para  $\{\boldsymbol{\theta}^{(j)}; j = 1, 2, \dots\}$ . Así, una aproximación del EMV es el valor de  $\boldsymbol{\theta}^{(j)}$  obtenido de la última iteración.

### 4.2.1. Ejemplo. Modelo Lineal Gaussiano

Considere el siguiente modelo de espacio de estados lineal Gaussiano

$$Y_t = \mu + \alpha_t + \varepsilon_t \quad (4.7)$$

donde  $\mu$  es la media general,  $\varepsilon_t \sim \text{iid } N(0, \sigma^2)$ ,  $t = 1, 2, \dots, n$  representan los errores del modelo (4.7), y las variables de estados  $\alpha_t$ ,  $t = 1, 2, \dots, n$  se modelan con un proceso autoregresivo de orden 1, *i. e.*,

$$\alpha_t = \phi \alpha_{t-1} + \eta_t \quad (4.8)$$

donde  $\eta_t \sim \text{iid } N(0, \tau^2)$ ,  $t = 1, 2, \dots, n$ . Además  $\varepsilon_t$  y  $\eta_t$ ,  $t = 1, 2, \dots, n$  son independientes. El vector de parámetros del modelo (4.7)–(4.8) está dado por  $\boldsymbol{\theta} := (\mu, \sigma^2, \phi, \tau^2)$ , con  $-\infty < \mu < \infty$ ,  $-1 < \phi < 1$ ,  $\sigma^2 \geq 0$  y  $\tau^2 \geq 0$ .

Sea  $w_1, w_2, \dots, w_n$  una realización de la serie de tiempo bajo el modelo (4.7)–(4.8). Supongamos que una parte de ellas han sido observadas con censura por la izquierda;

## 4.2. Estimación del SSMCen con el Algoritmo EM

esto es, los datos observados son  $y_1, y_2, \dots, y_n$ , donde  $w_t = y_t$  si  $\delta_t = 0$  y  $w_t \leq y_t$  si  $\delta_t = 1$ ,  $t = 1, 2, \dots, n$  y la region de censura esta dada por  $\mathcal{C} = \mathcal{C}_{t_1} \times \mathcal{C}_{t_2} \times \dots \times \mathcal{C}_{t_N}$ , donde  $\mathcal{C}_{t_j} = (-\infty, y_{t_j}]$  donde  $t_j$  es el tiempo de la  $j$ -ésima observación censurada.

En el Apéndice A se muestra que la log-verosimilitud de los datos completos es

$$l(\boldsymbol{\theta}; \mathbf{w}) = -\frac{n}{2} \log(2\pi) - \frac{1}{2} \sum_{t=1}^n \log(\Omega_t + \sigma^2) - \frac{1}{2} \sum_{t=1}^n \frac{(w_t - \mu - \hat{\alpha}_t)^2}{(\Omega_t + \sigma^2)} \quad (4.9)$$

donde  $\hat{\alpha}_t, t = 1, \dots, n$  son la predicciones de  $\alpha_t$  de un paso y  $\Omega_t$  es la varianza del error, los cuales son obtenidos a partir de las recursiones de predicción de Kalman (?).

Para aplicar el algoritmo EM se requiere conocer la distribución predictiva condicional  $f(\mathbf{z}|\mathbf{y}, \boldsymbol{\delta}, \boldsymbol{\theta})$ , que para este ejemplo, es una distribución normal condicional en el sentido que el valor de  $Z_t$  es menor a  $y_t$  cuando  $\delta_t = 0$ . Entonces, para el paso E del algoritmo EM hay que calcular,

$$\begin{aligned} Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(j)}) &= E(\ln f(\mathbf{z}, \mathbf{y}; \boldsymbol{\theta}) | \mathbf{y}, \boldsymbol{\delta}, \boldsymbol{\theta}^{(j)}) \\ &= E \left( -\frac{n}{2} \log(2\pi) - \frac{1}{2} \sum_{t=1}^n \log(\Omega_t + \sigma^2) - \frac{1}{2} \sum_{t=1}^n \frac{(w_t - \mu - \hat{\alpha}_t)^2}{(\Omega_t + \sigma^2)} \right) \\ &= -\frac{n}{2} \log(2\pi) - \frac{1}{2} \sum_{t=1}^n \log(\Omega_t + \sigma^2) \\ &\quad - \frac{1}{2} \sum_{t=1}^n \frac{E((w_t - \mu - \hat{\alpha}_t)^2 | \mathbf{y}_{1:t}, \delta_t, \boldsymbol{\theta}^{(j)})}{(\Omega_t + \sigma^2)} \end{aligned} \quad (4.10)$$

donde

$$E[(w_t - \mu - \hat{\alpha}_t)^2 | \mathbf{y}_{1:t}, \delta_t, \boldsymbol{\theta}^{(j)}] = \begin{cases} (y_t - \mu - \hat{\alpha}_t)^2, & \text{si } \delta_t = 1, \\ E[(Z_t - \mu - \hat{\alpha}_t)^2 | \mathbf{y}_{1:t}, \delta_t, \boldsymbol{\theta}^{(j)}], & \text{si } \delta_t = 0. \end{cases}$$

## 4.2. Estimación del SSMCen con el Algoritmo EM

Note que,

$$\begin{aligned}
 E[(Z_t - \mu - \hat{\alpha}_t)^2 | \mathbf{y}_{1:t}, \delta_t = 0, \boldsymbol{\theta}^{(j)}] &= E[(Z_t - (\mu + \hat{\alpha}_t))^2 | \mathbf{y}_{1:t}, Z_t < y_t, \boldsymbol{\theta}^{(j)}] \\
 &= E[Z_t^2 - 2(\mu + \hat{\alpha}_t)Z_t + (\mu + \hat{\alpha}_t)^2 | \mathbf{y}_{1:t}, Z_t < y_t, \boldsymbol{\theta}^{(j)}] \\
 &= E[Z_t^2 | \mathbf{y}_{1:t}, Z_t < y_t, \boldsymbol{\theta}^{(j)}] \\
 &\quad - 2(\mu + \hat{\alpha}_t)E[Z_t | \mathbf{y}_{1:t}, Z_t < y_t, \boldsymbol{\theta}^{(j)}] \\
 &\quad + (\mu + \hat{\alpha}_t)^2.
 \end{aligned} \tag{4.11}$$

Para calcular (4.11) es necesario obtener  $E[Z_t | \mathbf{y}_{1:t}, Z_t < y_t, \boldsymbol{\theta}^{(j)}]$  y  $E[Z_t^2 | \mathbf{y}_{1:t}, Z_t < y_t, \boldsymbol{\theta}^{(j)}]$  como sigue,

$$\begin{aligned}
 E[Z_t | \mathbf{y}_{1:t}, Z_t < y_t, \boldsymbol{\theta}^{(j)}] &= E[\mu + \hat{\alpha}_t + \sqrt{\Omega_t + \sigma^2} \varepsilon_t | \mathbf{y}_{1:t}, \mu + \hat{\alpha}_t + \sqrt{\Omega_t + \sigma^2} \varepsilon_t < y_t, \boldsymbol{\theta}^{(j)}] \\
 &= \mu + \hat{\alpha}_t + \sqrt{\Omega_t + \sigma^2} E[\varepsilon_t | \mathbf{y}_{1:t}, \varepsilon_t < \frac{(y_t - \mu - \hat{\alpha}_t)}{\sqrt{\Omega_t + \sigma^2}}, \boldsymbol{\theta}^{(j)}] \\
 &= \mu + \hat{\alpha}_t + \sqrt{\Omega_t + \sigma^2} \int_{-\infty}^{c_t} x \frac{\phi(x)}{\Phi(c_t)} dx \\
 &= \mu + \hat{\alpha}_t - \sqrt{\Omega_t + \sigma^2} \frac{\phi(c_t)}{\Phi(c_t)},
 \end{aligned} \tag{4.12}$$

donde  $c_t = \frac{(y_t - \mu - \hat{\alpha}_t)}{\sqrt{\Omega_t + \sigma^2}}$ ,  $\phi(\cdot)$  y  $\Phi(\cdot)$  representan la función de densidad y de distribución acumulada de la distribución normal estándar, respectivamente. También,

$$\begin{aligned}
 E[Z_t^2 | \mathbf{y}_{1:t}, Z_t < y_t, \boldsymbol{\theta}^{(j)}] &= E[(\mu + \hat{\alpha}_t + \sqrt{\Omega_t + \sigma^2} \varepsilon_t)^2 | \mathbf{y}_{1:t}, Z_t < y_t, \boldsymbol{\theta}^{(j)}] \\
 &= (\mu + \hat{\alpha}_t)^2 + 2(\mu + \hat{\alpha}_t) \sqrt{\Omega_t + \sigma^2} E[\varepsilon_t | \mathbf{y}_{1:t}, \varepsilon_t < c_t, \boldsymbol{\theta}^{(j)}] \\
 &\quad + (\Omega_t + \sigma^2) E[\varepsilon_t^2 | \mathbf{y}_{1:t}, \varepsilon_t < c_t, \boldsymbol{\theta}^{(j)}]
 \end{aligned} \tag{4.13}$$

donde

$$\begin{aligned}
 E[\varepsilon_t^2 | \mathbf{y}_{1:t}, \varepsilon_t < c_t, \boldsymbol{\theta}^{(j)}] &= \int_{-\infty}^{c_t} x^2 \frac{\phi(x)}{\Phi(c_t)} dx \\
 &= \frac{1}{\Phi(c_t)} \int_{-\infty}^{c_t} x^2 \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2} dx
 \end{aligned}$$



## 4.2. Estimación del SSMCen con el Algoritmo EM

---

Integrando por partes esta última expresión se tiene que,

$$\begin{aligned}
 E[\varepsilon_t^2 | \mathbf{y}_{1:t}, \varepsilon_t < c_t, \boldsymbol{\theta}^{(j)}] &= -\frac{c_t e^{-\frac{1}{2}c_t^2}}{\Phi(c_t)\sqrt{2\pi}} + \frac{1}{\Phi(c_t)} \int_{-\infty}^{c_t} \frac{1}{\sqrt{2\pi}} e^{\frac{1}{2}x^2} dx \\
 &= -\frac{c_t \phi(c_t)}{\Phi(c_t)} + \frac{\Phi(c_t)}{\Phi(c_t)} \\
 &= 1 - \frac{c_t \phi(c_t)}{\Phi(c_t)}
 \end{aligned}$$

sustituyendo esta última expresión en (4.13) se tiene,

$$\begin{aligned}
 E[Z_t^2 | \mathbf{y}_{1:t}, Z_t < y_t, \boldsymbol{\theta}^{(j)}] &= (\mu + \hat{\alpha}_t)^2 + 2(\mu + \hat{\alpha}_t)\sqrt{\Omega_t + \sigma^2} \left[ -\frac{\phi(c_t)}{\Phi(c_t)} \right] \\
 &\quad + (\Omega_t + \sigma^2) \left[ 1 - \frac{c_t \phi(c_t)}{\Phi(c_t)} \right] \\
 &= (\mu + \hat{\alpha}_t)^2 + \Omega_t + \sigma^2 - \sqrt{\Omega_t + \sigma^2} (y_t + \mu + \hat{\alpha}_t) \frac{\phi(c_t)}{\Phi(c_t)}.
 \end{aligned}$$

Finalmente, sustituyendo (4.12) en (4.11) se obtiene,

$$\begin{aligned}
 E[(Z_t - \mu - \hat{\alpha}_t)^2 | \mathbf{y}_{1:t}, \delta_t = 0, \boldsymbol{\theta}^{(j)}] &= \Omega_t + \sigma^2 - \sqrt{\Omega_t + \sigma^2} (y_t + \mu + \hat{\alpha}_t) \frac{\phi(c_t)}{\Phi(c_t)} \\
 &\quad - 2(\mu + \hat{\alpha}_t) \left[ (\mu + \hat{\alpha}_t) - \sqrt{\Omega_t + \sigma^2} \frac{\phi(c_t)}{\Phi(c_t)} \right] \\
 &\quad + 2(\mu + \hat{\alpha}_t)^2 \\
 &= \Omega_t + \sigma^2 - (y_t + \mu + \hat{\alpha}_t) \sqrt{\Omega_t + \sigma^2} \frac{\phi(c_t)}{\Phi(c_t)} \\
 &\quad + 2(\mu + \hat{\alpha}_t) \sqrt{\Omega_t + \sigma^2} \frac{\phi(c_t)}{\Phi(c_t)} \\
 &= \Omega_t + \sigma^2 + \sqrt{\Omega_t + \sigma^2} \frac{\phi(c_t)}{\Phi(c_t)} (\mu + \hat{\alpha}_t - y_t) \quad (4.14)
 \end{aligned}$$

Para realizar el paso M, se maximiza la función en (4.10) mediante un proceso de optimización no lineal. En los Capítulos 5 y 6 se presentan estimaciones de los parámetros de este modelo usando el algoritmo EM.

Es posible calcular a la función  $Q$  en (4.6) de forma numérica a través de versiones estocásticas en el paso E del el algoritmo EM. Una mediante el algoritmo MCEM (véase

## 4.2. Estimación del SSMCen con el Algoritmo EM

---

la Sección 3.4.1) y la otra mediante el algoritmo SEM (Sección 3.4.2). En ambos casos, es necesario calcular la distribución conjunta condicional de los datos censurados y/o perdidos dados los datos observados,  $f(\mathbf{z}|\mathbf{y}, \boldsymbol{\delta}, \boldsymbol{\theta})$ , y obtener muestras de esta distribución. En la siguiente sección se presenta una propuesta de cálculo de la distribución condicional  $f(\mathbf{z}|\mathbf{y}, \boldsymbol{\delta}, \boldsymbol{\theta})$ .

### 4.2.2. Distribución predictiva

La distribución predictiva puede verse como la distribución condicional de los datos censurados y/o perdidos dado los valores observados. Para facilidad de manejo de tal distribución, se incluye una variable auxiliar,  $\delta_t$ , definida en (4.1) que indica si la observación está o no censurada en el tiempo  $t$ . El cálculo de  $f(\mathbf{z}|\mathbf{y}, \boldsymbol{\delta}, \boldsymbol{\theta})$  implica conocer las distribuciones condicionales  $f(\alpha_t|\mathbf{z}_{1:t}, \mathbf{y}_{1:t}, \boldsymbol{\delta}_{1:t})$  y  $f(\alpha_t|\mathbf{z}_{1:t-1}, \mathbf{y}_{1:t}, \boldsymbol{\delta}_{1:t})$ , suponiendo que la distribución de  $Z_t$  dado  $(\alpha_t, \boldsymbol{\alpha}_{1:t-1}, \mathbf{z}_{1:t-1}, \mathbf{y}_{1:t})$  no depende de  $(\mathbf{z}_{1:t-1}, \mathbf{y}_{1:t-1}, \boldsymbol{\delta}_{1:t-1})$ , es decir

$$f(z_t|\boldsymbol{\alpha}_{1:t}, \mathbf{z}_{1:t-1}, \mathbf{y}_{1:t}, \boldsymbol{\delta}_{1:t}) := f(z_t|\alpha_t, y_t, \delta_t)$$

donde

$$f(z_t|\alpha_t, y_t, \delta_t) = \begin{cases} f_{Y_t|\alpha_t}(y_t|\alpha_t), & \text{si } \delta_t = 1, \\ \frac{f_{Y_t|\alpha_t}(z_t|\alpha_t)}{\int_{\mathcal{C}} f_{Y_t|\alpha_t}(x_t|\alpha_t) dx_t} 1_{\mathcal{C}}(z_t), & \text{si } \delta_t = 0. \end{cases}$$

donde  $\mathcal{C}$  es una región de censura conocida, en la cual  $Z_t$  toma valores. Es decir, cuando la observación  $y_t$  está censurada por la izquierda en  $c_t$ , entonces  $f(z_t|\alpha_t, y_t, \delta_t)$  es la versión truncada por la derecha en  $c_t$  de  $f_{Y_t|\alpha_t}(y_t|\alpha_t)$ . Por lo contrario, si la observación es censurada por la derecha, entonces usamos la distribución truncada por la izquierda.

## 4.2. Estimación del SSMCen con el Algoritmo EM

---

Aplicando el teorema de Bayes se tiene que

$$\begin{aligned}
 f(\alpha_t | \mathbf{z}_{1:t}, \mathbf{y}_{1:t}, \boldsymbol{\delta}_{1:t}) &= \frac{f(\alpha_t, \mathbf{z}_{1:t}, \mathbf{y}_{1:t}, \boldsymbol{\delta}_{1:t})}{f(\mathbf{z}_{1:t}, \mathbf{y}_{1:t}, \boldsymbol{\delta}_{1:t})} \\
 &= \frac{f(z_t | \alpha_t, \mathbf{z}_{1:t-1}, \mathbf{y}_{1:t}, \boldsymbol{\delta}_{1:t}) f(\alpha_t, \mathbf{z}_{1:t-1}, \mathbf{y}_{1:t}, \boldsymbol{\delta}_{1:t})}{f(\mathbf{z}_{1:t}, \mathbf{y}_{1:t}, \boldsymbol{\delta}_{1:t})} \\
 &= \frac{f(z_t | \alpha_t, y_t, \delta_t) f(\alpha_t | \mathbf{z}_{1:t-1}, \mathbf{y}_{1:t}, \boldsymbol{\delta}_{1:t}) f(\mathbf{z}_{1:t-1}, \mathbf{y}_{1:t}, \boldsymbol{\delta}_{1:t})}{f(\mathbf{z}_{1:t}, \mathbf{y}_{1:t}, \boldsymbol{\delta}_{1:t})} \\
 &= \frac{f(z_t | \alpha_t, y_t, \delta_t) f(\alpha_t | \mathbf{z}_{1:t-1}, \mathbf{y}_{1:t}, \boldsymbol{\delta}_{1:t})}{f(z_t | \mathbf{z}_{1:t-1}, \mathbf{y}_{1:t}, \boldsymbol{\delta}_{1:t})} \\
 &\propto f(z_t | \alpha_t, y_t, \delta_t) f(\alpha_t | \mathbf{z}_{1:t-1}, \mathbf{y}_{1:t}, \boldsymbol{\delta}_{1:t}) \tag{4.15}
 \end{aligned}$$

y

$$\begin{aligned}
 f(\alpha_{t+1} | \mathbf{z}_{1:t}, \mathbf{y}_{1:t+1}, \boldsymbol{\delta}_{1:t+1}) &= \frac{f(\alpha_{t+1}, \mathbf{z}_{1:t}, \mathbf{y}_{1:t+1}, \boldsymbol{\delta}_{1:t+1})}{f(\mathbf{z}_{1:t}, \mathbf{y}_{1:t+1}, \boldsymbol{\delta}_{1:t+1})} \\
 &= \frac{\int f(\alpha_{t+1}, \alpha_t, \mathbf{z}_{1:t}, \mathbf{y}_{1:t+1}, \boldsymbol{\delta}_{1:t+1}) d\mu(\alpha_t)}{f(\mathbf{z}_{1:t}, \mathbf{y}_{1:t+1}, \boldsymbol{\delta}_{1:t+1})} \\
 &= \frac{1}{f(\mathbf{z}_{1:t}, \mathbf{y}_{1:t+1}, \boldsymbol{\delta}_{1:t+1})} \times \\
 &\quad \int \{f(\alpha_{t+1} | \alpha_t, \mathbf{z}_{1:t}, \mathbf{y}_{1:t+1}, \boldsymbol{\delta}_{1:t+1}) \times \\
 &\quad \quad f(\alpha_t, \mathbf{z}_{1:t}, \mathbf{y}_{1:t+1}, \boldsymbol{\delta}_{1:t+1})\} d\mu(\alpha_t) \\
 &= \frac{1}{f(\mathbf{z}_{1:t}, \mathbf{y}_{1:t+1}, \boldsymbol{\delta}_{1:t+1})} \int \{f(\alpha_{t+1} | \alpha_t) \times \\
 &\quad \quad f(\alpha_t | \mathbf{z}_{1:t}, \mathbf{y}_{1:t+1}, \boldsymbol{\delta}_{1:t+1}) f(\mathbf{z}_{1:t}, \mathbf{y}_{1:t+1}, \boldsymbol{\delta}_{1:t+1})\} d\mu(\alpha_t) \\
 &= \int f(\alpha_{t+1} | \alpha_t) f(\alpha_t | \mathbf{z}_{1:t}, \mathbf{y}_{1:t}, \boldsymbol{\delta}_{1:t}) d\mu(\alpha_t). \tag{4.16}
 \end{aligned}$$

El denominador en la penúltima expresión de (4.15) es un factor de escala determinado por la condición de que  $\int f(\alpha_t | \mathbf{z}_{1:t}, \mathbf{y}_{1:t}, \boldsymbol{\delta}_{1:t}) d\alpha_t = 1$ . La condición inicial para llevar a cabo estas recursiones es

$$f(\alpha_1 | z_0, y_1, \delta_1) := f(\alpha_1)$$

La densidad condicional de  $Z_{t+1}$  dado  $(\mathbf{z}_{1:t}, \mathbf{y}_{1:t+1}, \boldsymbol{\delta}_{1:t+1})$  puede ser calculada de (4.16)

### 4.3. Algoritmo EM Monte Carlo

---

como

$$\begin{aligned}
 f(z_{t+1} | \mathbf{z}_{1:t}, \mathbf{y}_{1:t+1}, \boldsymbol{\delta}_{1:t+1}) &= \frac{f(z_{t+1}, \mathbf{z}_{1:t}, \mathbf{y}_{1:t+1}, \boldsymbol{\delta}_{1:t+1})}{f(\mathbf{z}_{1:t}, \mathbf{y}_{1:t+1}, \boldsymbol{\delta}_{1:t+1})} \\
 &= \frac{\int f(z_{t+1}, \alpha_{t+1}, \mathbf{z}_{1:t}, \mathbf{y}_{1:t+1}, \boldsymbol{\delta}_{1:t+1}) d\mu(\alpha_{t+1})}{f(\mathbf{z}_{1:t}, \mathbf{y}_{1:t+1}, \boldsymbol{\delta}_{1:t+1})} \\
 &= \frac{1}{f(\mathbf{z}_{1:t}, \mathbf{y}_{1:t+1}, \boldsymbol{\delta}_{1:t+1})} \times \\
 &\quad \int f(z_{t+1} | \alpha_{t+1}, \mathbf{z}_{1:t}, \mathbf{y}_{1:t+1}, \boldsymbol{\delta}_{1:t+1}) \times \\
 &\quad \quad f(\alpha_{t+1}, \mathbf{z}_{1:t}, \mathbf{y}_{1:t+1}, \boldsymbol{\delta}_{1:t+1}) d\mu(\alpha_{t+1}) \\
 &= \int f(z_{t+1} | \alpha_{t+1}, y_{t+1}, \delta_{t+1}) \times \\
 &\quad \quad f(\alpha_{t+1} | \mathbf{z}_{1:t}, \mathbf{y}_{1:t+1}, \boldsymbol{\delta}_{1:t+1}) d\mu(\alpha_{t+1}) \quad (4.17)
 \end{aligned}$$

Finalmente, la densidad condicional de  $\mathbf{Z} = (Z_0, Z_1, \dots, Z_n)'$  dado  $(\mathbf{y}, \boldsymbol{\delta}, \boldsymbol{\theta})$  puede ser expresada como

$$f(\mathbf{z} | \mathbf{y}, \boldsymbol{\delta}, \boldsymbol{\theta}) = \prod_{t=1}^n f(z_t | \mathbf{z}_{1:t-1}, \mathbf{y}_{1:t}, \boldsymbol{\delta}_{1:t}) \quad (4.18)$$

donde

$$f(z_1 | z_0, y_1, \delta_1) = f(z_1 | y_1, \delta_1) = f(z_1 | y_1, \delta_1) = \begin{cases} f_{Y_1}(y_1), & \delta_1 = 1, \\ \frac{f_{Y_1}(z_1)}{\int_{\mathcal{C}} f_{Y_1}(u_1) du_1}, & \delta_1 = 0. \end{cases}$$

### 4.3. Algoritmo EM Monte Carlo

Como ya se ha mencionado en la Sección 3.4.1, la idea de éste método es facilitar el paso  $E$  del algoritmo EM estimando la función  $Q$  mediante el método Monte Carlo. Así el paso  $E$  en (4.6) en términos del algoritmo MCEM de la Sección 3.4.1 es como sigue:

Dado un estimador  $\boldsymbol{\theta}^{(i)}$  de  $\boldsymbol{\theta}$ ,

1. Generar  $\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_m$  i.i.d.  $\sim f(\mathbf{z} | \mathbf{y}, \boldsymbol{\delta}, \boldsymbol{\theta}^{(i)})$

### 4.3. Algoritmo EM Monte Carlo

---

2. Calcular

$$\hat{Q}_{i+1}(\boldsymbol{\theta}, \boldsymbol{\theta}^{(i)}) = \frac{1}{m} \sum_{j=1}^m \ln f(\mathbf{z}_j, \mathbf{y}; \boldsymbol{\theta}). \quad (4.19)$$

En el paso  $M$ ,  $\hat{Q}$  en (4.19) se maximiza para obtener  $\boldsymbol{\theta}^{(i+1)}$  y el procedimiento es iterado hasta que se cumpla un criterio de convergencia establecido.

Para muestrear de  $f(\mathbf{z}|\mathbf{y}, \boldsymbol{\delta}, \boldsymbol{\theta}^{(i)})$  se procede como sigue

Para  $t = 1$ ,

1. Obténgase  $\alpha_1$  de  $f(\alpha_1)$ . Por ejemplo, para el caso del modelo presentado en (4.8),  $\alpha_1 \sim N(0, \frac{\tau^2}{1-\phi^2})$
2. Usando  $\alpha_1$ , obténgase  $z_1$  de (4.17) mediante el método de composición.
3. Usando (4.15), generar  $\alpha_1^f$

Para  $t = 2, 3, \dots, n$  y usando el método de composición

- a. Úsese  $\alpha_{t-1}^f$  para generar  $\alpha_t^p$  de (4.16).
- b. Úsese  $\alpha_t^p$  para obtener  $z_t$  de (4.17).
- c. Con  $\alpha_t^p$  y  $z_t$  se genera  $\alpha_t^f$  de (4.15).

El procedimiento 1–3 y a-c se repiten  $m$ -veces para generar  $m$  muestras de  $f(\mathbf{z}|\mathbf{y}, \boldsymbol{\delta}, \boldsymbol{\theta}^{(i)})$ .

Para el caso del ejemplo presentado en la Sección 4.2.1, se tiene que  $Z_t | (\alpha_t, y_t, \delta_t = 1) \sim N(\mu + \alpha_t, \sigma^2)$ , y  $Z_t | (\alpha_t, y_t, \delta_t = 0)$  se distribuye como

$$\frac{\phi(z_t; \mu + \alpha_t, \sigma^2)}{\int_{-\infty}^{y_t} \Phi(u_t; \mu + \alpha_t, \sigma^2) du_t} 1_{(-\infty, y_t)}(z_t)$$

es decir, si  $\delta_t = 0$  se trata de la distribución normal truncada por la derecha en  $y_t$ , denotada por  $\phi_{TD}(y_t; \mu + \alpha_t, \sigma^2)$ , y

$$f(\alpha_{t+1} | \alpha_t) = \phi(\alpha_{t+1}; \phi\alpha_t, \tau^2).$$

## 4.4. Algoritmo EM Estocástico

Para el modelo SSMCen, el algoritmo SEM, descrito en la Sección 3.4.2 es como sigue:

1. Se inicia el algoritmo con una valor inicial  $\boldsymbol{\theta}^{(0)}$ , el EMV de la serie observada “pretendiendo” que no hay valores censurados.
2. Para llevar a cabo el Paso-S en la  $i$ -ésima iteración, se obtiene una muestra de  $f(\mathbf{z}|\mathbf{y}, \boldsymbol{\delta}, \boldsymbol{\theta}^{(i-1)})$ , usando el procedimiento descrito en la Sección 4.3 y se calcula la función Q en (4.19) con  $m = 1$ .
3. En el paso M, se actualiza el estimador calculando  $\boldsymbol{\theta}^{(i)}$  mediante un procedimiento de optimización no lineal de la función  $\hat{Q}_i(\boldsymbol{\theta}, \boldsymbol{\theta}^{(i-1)})$  en (4.19) con  $m = 1$ .

Este procedimiento es iterado hasta que la cadena generada  $\boldsymbol{\theta}^{(i)}$  cumple con un criterio de convergencia establecido de antemano. Después de un periodo de “burn-in” de  $m_0$  iteraciones, la sucesión  $\boldsymbol{\theta}^{(i)}$  podría acercarse a su régimen estacionario y entonces podemos parar el proceso iterativo después de un numero suficientemente grande de iteraciones  $T$ . De esta forma, el estimador SEM del vector de parámetros  $\boldsymbol{\theta}$  está dado

$$\tilde{\boldsymbol{\theta}} = (T - m_0)^{-1} \sum_{m=m_0+1}^T \boldsymbol{\theta}^{(m)}. \quad (4.20)$$

## 4.5. Estimación del SSMCen usando IS

En esta sección se aplica el algoritmo de muestreo de importancia para aproximar la integral en (4.5). La función de importancia que se propone, denotada por  $g_{ic}(\boldsymbol{\alpha}|\mathbf{y})$ , es la densidad condicional de  $\boldsymbol{\alpha}$  dado  $\mathbf{y}$  “ignorando” censura en las observaciones. Así, si  $\boldsymbol{\alpha}^{(1)}, \boldsymbol{\alpha}^{(2)}, \dots, \boldsymbol{\alpha}^{(M)}$  es una muestra aleatoria de  $g_{ic}(\boldsymbol{\alpha}|\mathbf{y})$ , por el muestreo de importancia se tiene que

$$\hat{L}_{MI}(\boldsymbol{\theta}; \mathbf{y}, \boldsymbol{\delta}) = \frac{1}{M} \sum_{j=1}^M \frac{L(\boldsymbol{\theta}; \mathbf{y}, \boldsymbol{\alpha}^{(j)}, \boldsymbol{\delta})}{g_{ic}(\boldsymbol{\alpha}^{(j)}|\mathbf{y})}, \quad (4.21)$$

## 4.5. Estimación del SSMCen usando IS

---

es un estimador consistente de (4.5) (?). O equivalentemente, la expresión (4.21) puede ser calculada como

$$\hat{L}_{MI}(\boldsymbol{\theta}; \mathbf{y}, \boldsymbol{\delta}) = \frac{\sum_{j=1}^M f(\mathbf{y} | \boldsymbol{\alpha}; \boldsymbol{\xi}) f(\boldsymbol{\alpha}; \boldsymbol{\psi}) / g_{ic}(\boldsymbol{\alpha}^{(j)} | \mathbf{y})}{\sum_{j=1}^M f(\boldsymbol{\alpha}; \boldsymbol{\psi}) / g_{ic}(\boldsymbol{\alpha}^{(j)} | \mathbf{y})}. \quad (4.22)$$

? mencionan que este estimador mas estable<sup>1</sup> que (4.21).

En ambos casos, se requiere obtener a  $g_{ic}(\boldsymbol{\alpha} | \mathbf{y})$ . Sea  $\mathbf{y}_{1:t} := (y_1, \dots, y_t)'$  y considere la factorización

$$g_{ic}(\boldsymbol{\alpha} | \mathbf{y}) = g_{ic}(\alpha_n | y_{1:n}) \prod_{t=1}^{n-1} g_{ic}(\alpha_t | \alpha_{t+1:n}, y_{1:n}), \quad (4.23)$$

donde

$$\begin{aligned} g_{ic}(\alpha_t | \alpha_{t+1:n}, y_{1:n}) &= g_{ic}(\alpha_t | \alpha_{t+1}, y_{1:t}) \\ &= g_{ic}(\alpha_t | y_{1:t}) f(\alpha_{t+1} | \alpha_t) / g_{ic}(\alpha_{t+1} | y_{1:t}) \\ &\propto g_{ic}(\alpha_t | y_{1:t}) f(\alpha_{t+1} | \alpha_t). \end{aligned}$$

Para obtener una realización  $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \dots, \alpha_n)'$  de  $g_{ic}(\boldsymbol{\alpha} | \mathbf{y})$ , considere que

$$g_{ic}(\boldsymbol{\alpha} | \mathbf{y}) = g_{ic}(\alpha_n | y_{1:n}) \prod_{t=1}^{n-1} g_{ic}(\alpha_t | \alpha_{t+1:n}, y_{1:n}). \quad (4.24)$$

Entonces, el estimador IS está dado por

$$\hat{\boldsymbol{\theta}} = \arg \max_{\boldsymbol{\theta} \in \Theta} \hat{L}_{MI}(\boldsymbol{\theta}; \mathbf{y}, \boldsymbol{\delta})$$

que aproxima al EMV de  $\boldsymbol{\theta}$ .

Considere el ejemplo presentado en la Sección 4.2.1 donde algunas de las observaciones están censuradas por la izquierda, entonces la función de verosimilitud “completa”

---

<sup>1</sup>Estable en el sentido que asegura una varianza finita para  $L(\boldsymbol{\theta}; \mathbf{y}, \boldsymbol{\delta})$ .

## 4.5. Estimación del SSMCen usando IS

puede escribirse como

$$\begin{aligned} L(\boldsymbol{\theta}; \mathbf{y}, \boldsymbol{\alpha}, \boldsymbol{\delta}) &= f(\mathbf{y} | \boldsymbol{\alpha}; \boldsymbol{\xi}) f(\boldsymbol{\alpha}; \boldsymbol{\psi}) \\ &= \left( \prod_{t=1}^n f_{Y_t | \alpha_t}(y_t | \alpha_t; \boldsymbol{\xi})^{\delta_t} F_{Y_t | \alpha_t}(y_t | \alpha_t; \boldsymbol{\xi})^{1-\delta_t} \right) \\ &\quad \times |\mathbf{V}|^{-1/2} e^{-\boldsymbol{\alpha}^T \mathbf{V}^{-1} \boldsymbol{\alpha} / 2} / (2\pi)^{n/2}, \end{aligned} \quad (4.25)$$

donde  $F_{Y_t | \alpha_t}(y_t | \alpha_t) = \int_{-\infty}^{y_t} f_{Y_t | \alpha_t}(u_t | \alpha_t) du_t$  es la función de distribución acumulada de  $Y_t | \alpha_t$  evaluada en  $y_t$ , y  $Y_t | \alpha_t \sim N(\mu + \alpha_t, \sigma^2)$ .

Sustituyendo (4.25) en (4.5) se obtiene la función de verosimilitud de los datos observados. Usando la expresión (4.24) se puede obtener una realización  $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \dots, \alpha_n)'$  de  $g_{ic}(\boldsymbol{\alpha} | \mathbf{y})$ . Como se puede ver en el Anexo B,

$$g_{ic}(\alpha_t | \alpha_{t+1:n}, y_{1:n}) = \phi(\alpha_t; \mu_t^*, \nu_t^*)$$

donde

$$\mu_t^* = \frac{\tau^2 \alpha_{t|t} + \phi \alpha_{t+1} \Omega_{t|t}}{\tau^2 + \phi^2 \Omega_{t|t}} \quad \text{y} \quad \nu_t^* = \frac{\tau^2 \Omega_{t|t}}{\tau^2 + \phi^2 \Omega_{t|t}}; \quad t = 1, \dots, n-1.$$

Además, del Apéndice A, se tiene que  $g_{ic}(\alpha_t | y_{1:t}) = \phi(\alpha_t; \alpha_{t|t}, \Omega_{t|t})$  y  $g_{ic}(\alpha_{t+1} | y_{1:t}) = \phi(\alpha_{t+1}; \hat{\alpha}_{t+1}, \Omega_{t+1})$ , donde  $\alpha_{t|t}$ ,  $\Omega_{t|t}$ ,  $\hat{\alpha}_{t+1}$  y  $\Omega_{t+1}$  se obtienen a partir de las recursiones de Kalman. En resumen, el procedimiento para obtener una realización  $\boldsymbol{\alpha}$  de  $g_{ic}(\boldsymbol{\alpha} | \mathbf{y})$ , es el siguiente

1. Se obtienen las estimaciones de predicción y del filtro de las  $\alpha$ 's (ver Apéndice A); es decir,

$$\alpha_{t|t}, \Omega_{t|t}, \hat{\alpha}_{t+1} \text{ y } \Omega_{t+1}; \quad t = 1, \dots, n.$$

2. Se genera  $\alpha_n$  de  $\phi(\alpha_n; \alpha_{n|n}, \Omega_{n|n})$ .
3. Se obtienen las  $\alpha_t$ 's de  $\phi(\alpha_t; \mu_t^*, \nu_t^*)$ ; para  $t = n-1, n-2, \dots, 1$ .

Se repiten los pasos 1 a 3 hasta obtener una muestra de tamaño  $M$ .



# Capítulo 5

## Estudio de Simulación

En este capítulo se estudia el desempeño de los estimadores propuestos en el capítulo anterior mediante un estudio de simulación en el que se consideran diferentes porcentajes de censura por la izquierda. El modelo que se considera en este estudio es el SSM lineal Gaussiano presentado en la Sección 4.2.1, es decir,

$$Y_t = \mu + \alpha_t + \varepsilon_t \quad (5.1)$$

donde  $\mu$  es la media general,  $\varepsilon_t \sim \text{iid } N(0, \sigma^2)$ ,  $t = 1, 2, \dots, n$  representan los errores del modelo (5.1), y las variables de estados  $\alpha_t$ ,  $t = 1, 2, \dots, n$  se modelan con un proceso autoregresivo de orden 1, *i. e.*,

$$\alpha_t = \phi\alpha_{t-1} + \eta_t \quad (5.2)$$

donde  $\eta_t \sim \text{iid } N(0, \tau^2)$ ,  $t = 1, 2, \dots, n$ . Además  $\varepsilon_t$  y  $\eta_t$ ,  $t = 1, 2, \dots, n$  son independientes. El vector de parámetros del modelo (5.1)–(5.2) está dado por  $\boldsymbol{\theta} := (\mu, \sigma^2, \phi, \tau^2)$ , con  $-\infty < \mu < \infty$ ,  $-1 < \phi < 1$ ,  $\sigma^2 > 0$  y  $\tau^2 > 0$ .

En este estudio se usará  $\boldsymbol{\theta} := (30, 2, 0.9, 1)$  y las regiones de censura  $\mathcal{C}_t$  están dadas por  $\mathcal{C}_t = (-\infty, c_j]$ ,  $j = 1, 2, 3$ . En la Tabla 5.1 se listan los valores de  $c_j$  que se usarán, y los niveles (promedio) de censura.

La Figura 5.1 muestra una realización del modelo (5.1)–(5.2) y  $c_2 = 27.77$ . La serie contiene un 20% de valores censurados por la izquierda, los cuales son representados

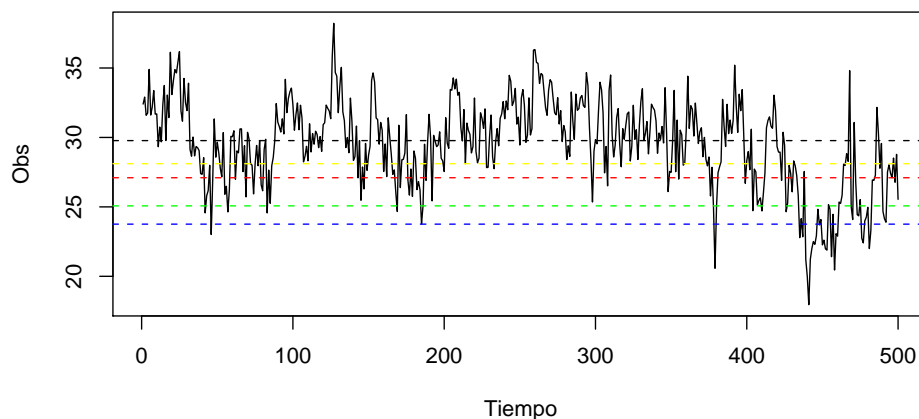
## 5. Estudio de Simulación

---

**Tabla 5.1:** Límites de censura.

$c_j$	% de censura
26.61	10 %
27.77	20 %
29.33	40 %

por triángulos con picos hacia abajo.



**Figura 5.1:** Serie de tiempo simulada

Todos los procedimientos propuestos se inician con un valor de  $\theta$ ,  $\theta^{(0)}$ , el cual se obtiene optimizando la función (4.9) considerando a la muestra observada como si fueran los datos completos, es decir, ignorando censura en la serie.

En la Tabla 5.2 se presenta el sesgo y error cuadrático medio estimados para los parámetros en el modelo (5.1)–(5.2), usando los estimadores EM, MCEM, SEM y IS, para los niveles de censura de 10 %, 20 % y 40 % respectivamente, con  $S = 500$  realizaciones de tamaños de muestra  $n = 500$  y en la Tabla 5.3 con tamaños de muestra de  $n = 100$ .

Los estimadores EM se obtuvieron de acuerdo con la descripción hecha del algoritmo EM en la Sección 4.2. Para implementar el algoritmo MCEM se sigue el procedimiento presentado en la Sección 4.3, que consiste en obtener  $m$  muestras de la distribución

## 5. Estudio de Simulación

---

condicional  $f(\mathbf{z}|\mathbf{y}, \boldsymbol{\delta}, \boldsymbol{\theta}^{(i)})$ . Para determinar el valor de  $m$ , en este estudio se utilizó una modificación al criterio utilizado por ?, la cual consiste en el siguiente procedimiento:

1. Se inicia el procedimiento con un valor inicial de  $\boldsymbol{\theta}, \boldsymbol{\theta}^0$ .
2. Se realiza un periodo de calentamiento con valores de  $m$  igual a 10, 100, 200, 500 ( $\times 2$ ) y 1000 ( $\times 3$ ), obteniendo  $\boldsymbol{\theta}^{(j)}, j = 1, \dots, 8$ , con  $\boldsymbol{\theta}^{(8)}$  se calcula una estimación de la desviación estándar de  $\hat{\boldsymbol{\theta}}$  y se determina  $s_1$  como la desviación estándar mayor de las desviaciones estimadas de los elementos de  $\hat{\boldsymbol{\theta}}$ .
3. Se inicia nuevamente el algoritmo MCEM con el ultimo valor obtenido de  $\boldsymbol{\theta}^{(j)}$  como valor inicial y  $m$  igual al entero mas pequeño q sea mayor que  $m_1 s_1 / \lambda$ , donde  $m_1$  es el ultimo valor de  $m$  utilizado en el periodo de calentamiento (del paso 2,  $m_1 = 1000$ ) y  $\lambda$  es un nivel predeterminado de tolerancia. En este trabajo se considera  $\lambda = 0.01$
4. Finalmente, se utiliza como criterio de parada que el valor de  $\Delta Q$ , definido como  $\Delta Q = |\hat{Q}(\boldsymbol{\theta}^{(j)}, \boldsymbol{\theta}^{(j-1)}) - \hat{Q}(\boldsymbol{\theta}^{(j-1)}, \boldsymbol{\theta}^{(j-1)})|$  sea menor a 0.001.

Para calcular los estimadores usando el algoritmo SEM se emplea el procedimiento descrito en la Sección 4.4. El número de iteraciones  $T$  que se realizaron para este ejemplo así como también el periodo de “burn-in” establecido en las cadenas se obtuvieron de acuerdo con el criterio propuesto por ?, y el estimador SEM se obtiene mediante la expresión (4.20).

De acuerdo con lo descrito en la Sección 4.5 se obtienen estimadores de maxima verosimilitud (EMV) via muestreo de importancia (estimadores IS) de los parámetros del modelo SSM presentado por (5.1)–(5.2). Para un  $M = 500$ , calculamos  $\hat{L}_{MI}(\boldsymbol{\theta}; \mathbf{y}, \boldsymbol{\delta})$  como en (4.22) y usamos este valor para calcular un estimador de  $\boldsymbol{\theta}$  mediante un algoritmo de optimización no lineal.

En esta tabla se puede observar que tanto  $\widehat{S}_{\mathbf{y}}(\boldsymbol{\theta})$  como  $\widehat{\text{ecm}}(\boldsymbol{\theta})$  aumentan conforme se incrementa el nivel de censura en los datos. Es decir, a mayor pérdida de información en los datos el sesgo y el error cuadrático medio se incrementan.

El comportamientos de las estimaciones es muy similar con respecto a las hechas con el algoritmo EM, es decir a medida que aumenta el nivel se censura en las observaciones el sesgo y el error cuadrático medio se incrementan.

## 5. Estudio de Simulación

---

**Tabla 5.2:** Sesgo estimado y error cuadrático medio (en paréntesis) utilizando  $S = 500$  repeticiones, con diferentes porcentajes de censura,  $\theta = (30, 2, 0.9, 1.0)$  y series de tiempo de tamaño 100.

% de censura	$\mu$	$\sigma^2$	$\phi$	$\tau^2$
0 %	0.035 (0.927)	-0.160 (0.374)	-0.057 (0.011)	0.194 (0.390)
Estimadores EM				
10 %	0.211 (0.822)	-0.338 (0.416)	-0.059 (0.012)	-0.041 (0.259)
20 %	0.330 (0.864)	-0.426 (0.525)	-0.068 (0.016)	-0.152 (0.275)
Estimadores MCEM				
10 %	0.140 (0.828)	-0.281 (0.395)	-0.059 (0.013)	0.062 (0.252)
20 %	0.231 (0.789)	-0.367 (0.477)	-0.063 (0.013)	-0.033 (0.222)
Estimadores SEM				
10 %	0.143 (0.825)	-0.284 (0.406)	-0.059 (0.012)	0.068 (0.266)
20 %	0.221 (0.810)	-0.388 (0.510)	-0.066 (0.015)	-0.018 (0.231)
Estimadores IS				
10 %	0.070 (0.877)	-0.054 (0.313)	-0.052 (0.011)	0.059 (0.229)
20 %	0.110 (0.866)	0.123 (0.522)	-0.045 (0.011)	-0.089 (0.210)

## 5. Estudio de Simulación

---

**Tabla 5.3:** Sesgo estimado y error cuadrático medio (en paréntesis) utilizando  $S = 500$  repeticiones, con diferentes porcentajes de censura,  $\theta = (30, 2, 0.9, 1.0)$  y series de tiempo de longitud 500.

% de censura	$\mu$	$\sigma^2$	$\phi$	$\tau^2$
0 %	0.010 (0.197)	-0.020 (0.059)	-0.011 (0.0009)	-0.032 (0.057)
Estimadores EM				
10 %	0.202 (0.205)	-0.242 (0.108)	-0.014 (0.0011)	-0.206 (0.080)
20 %	0.339 (0.271)	-0.342 (0.168)	-0.017 (0.0014)	-0.319 (0.133)
40 %	0.604 (0.506)	-0.484 (0.289)	-0.024 (0.0020)	-0.492 (0.266)
Estimadores MCEM				
10 %	0.085 (0.182)	-0.156 (0.079)	-0.0116 (0.0010)	-0.081 (0.048)
20 %	0.180 (0.190)	-0.246 (0.115)	-0.0138 (0.0011)	-0.172 (0.066)
40 %	0.427 (0.313)	-0.381 (0.203)	-0.0206 (0.0017)	-0.338 (0.1431)
Estimadores SEM				
10 %	0.087 (0.184)	-0.157 (0.081)	-0.012 (0.0010)	-0.079 (0.049)
20 %	0.181 (0.194)	-0.247 (0.117)	-0.014 (0.0012)	-0.169 (0.066)
40 %	0.425 (0.312)	-0.379 (0.202)	-0.021 (0.0017)	-0.334 (0.140)

## 5. Estudio de Simulación

---

Se puede observar que las estimaciones usando el algoritmo SEM tienen el mismo comportamiento con las estimaciones hechas con el EM y MCEM; sin embargo, se observa que el incremento tanto en sesgo como en error cuadrático medio es inferior a las dos anteriores.

Para el cálculo de los estimadores EM, MCEM, SEM y IS se realizaron subrutinas en Fortran las cuales fueron llamadas como DLL's (Dynamic Link Library, por sus siglas en inglés) mediante un código hecho en R. Las funciones y programas usados muestran en el Apéndice C.

# Capítulo 6

## Aplicaciones a datos reales

En este capítulo se presentan dos ejemplos de aplicación a datos reales usando el modelo SSM lineal Gaussiano definido por las ecuaciones (5.1) y (5.2). En el primero se analiza una serie de tiempo de fósforo en solución reactiva y en la segunda se analizan los datos presentados por ?.

### 6.1. Datos de Cedar R Logan

El Departamento de Ecología del estado de Washington registra datos de calidad del agua mensualmente en 118 estaciones ubicadas en los ríos del estado. El principal objetivo de este programa de monitoreo es la caracterización de la calidad del agua en los ríos del estado y evaluar los cambios de tendencia en la calidad del agua. Mas información sobre estos estudios se presentan en ?.

En esta sección solo nos limitamos a analizar la serie de tiempo  $\{y_t, t = 1, 2, \dots, 154\}$  de fósforo en solución reactiva en la estación 08C070 Cedar R Logan St Renton, como se mencionó en la Sección 3.2.1, el río fue monitoreado mensualmente de diciembre de 1994 a septiembre de 2007 por el Departamento de Ecología del estado de Washington, E.U. En la Figura 3.1 se muestra la serie  $\{y_t, t = 1, 2, \dots, 154\}$ , la cual presenta un 26.62% de censura por la izquierda.

## 6.1. Datos de Cedar R Logan

**Tabla 6.1:** Estimadores EM, MCEM, SEM y IS de modelo SSM lineal Gaussiano para los datos de Cedar R Logan.

Método	$\mu$	$\sigma^2$	$\phi$	$\tau^2$
EM	6.73 (0.038)	2.46 (0.089)	0.65 (0.008)	3.82 (0.115)
MCEM	6.74 (0.039)	2.12 (0.101)	0.636 (0.008)	4.35 (0.129)
SEM	6.74 (0.038)	2.11 (0.104)	0.635 (0.008)	4.36 (0.133)
IS	6.53 (0.044)	5.28 (0.096)	0.73 (0.006)	2.62 (0.071)

Para analizar esta serie, el modelo que se propone es el siguiente

$$Y_t = \mu + \alpha_t + \varepsilon_t \quad (6.1)$$

donde  $\mu$  es la media general,  $\varepsilon_t \sim \text{iid } N(0, \sigma^2)$ ,  $t = 1, 2, \dots, 154$  representan los errores del modelo (6.1), y las variables de estados  $\alpha_t$ ,  $t = 1, 2, \dots, n$  se modelan con un proceso autoregresivo de orden 1, *i. e.*,

$$\alpha_t = \phi\alpha_{t-1} + \eta_t \quad (6.2)$$

donde  $\eta_t \sim \text{iid } N(0, \tau^2)$ ,  $t = 1, 2, \dots, 154$ . Además  $\varepsilon_t$  y  $\eta_t$ ,  $t = 1, 2, \dots, 154$  son independientes. El vector de parámetros del modelo (6.1)–(6.2) está dado por  $\theta := (\mu, \sigma^2, \phi, \tau^2)$ , con  $-\infty < \mu < \infty$ ,  $-1 < \phi < 1$ ,  $\sigma^2 > 0$  y  $\tau^2 > 0$ .

En la Tabla 6.1 se presentan las estimaciones de este modelo usando los métodos propuestos y en paréntesis se muestran el error estándar estimado para cada una de las estimaciones.

Para iniciar los algoritmos EM, MCEM y SEM se usaron los valores iniciales para los parámetros  $\mu_0 = 7.04$ ,  $\sigma_0^2 = 1.41$ ,  $\phi_0 = 0.60$  y  $\tau_0^2 = 4.09$ . La diferencia  $|Q(\theta^{(j+1)}; \theta^{(j)}) - Q(\theta^{(j)}; \theta^{(j)})| < 0.001$  se usó como criterio de parada usado para el algoritmo EM. Los valores de  $m$  para usados para el algoritmo MCEM fueron obtenidos de acuerdo al criterio utilizado por ?.

La longitud de las cadenas, el periodo de “burn-in” y el factor de dependencia (I) para cada uno de los parámetros en el algoritmo SEM se obtuvieron mediante el criterio



## 6.2. Datos Zeger

---

**Tabla 6.2:** Longitud de las cadenas generadas para cada nivel de censura en los datos

Parámetros	$T$	$m_0$	I
$\mu$	2036	200	1.35
$\sigma^2$	2024	200	1.32
$\phi$	1894	100	1.23
$\tau^2$	1644	100	1.07

de convergencia de de Raftery y Lewis (?). El estimador SEM se obtiene mediante la expresión (4.20) y usando la longitud de la cadena mayor, la cual corresponde a 2036 iteraciones como se muestra en la Tabla 6.2.

En la Figura 6.1 se presentan las gráficas “running-mean” para un diagnóstico visual de convergencia para cada parámetros en las estimaciones del SEM. También, en la Figura 6.2 se presentan gráficas de autocorrelaciones de las cadenas generadas, en ellas se observa una convergencia rápida.

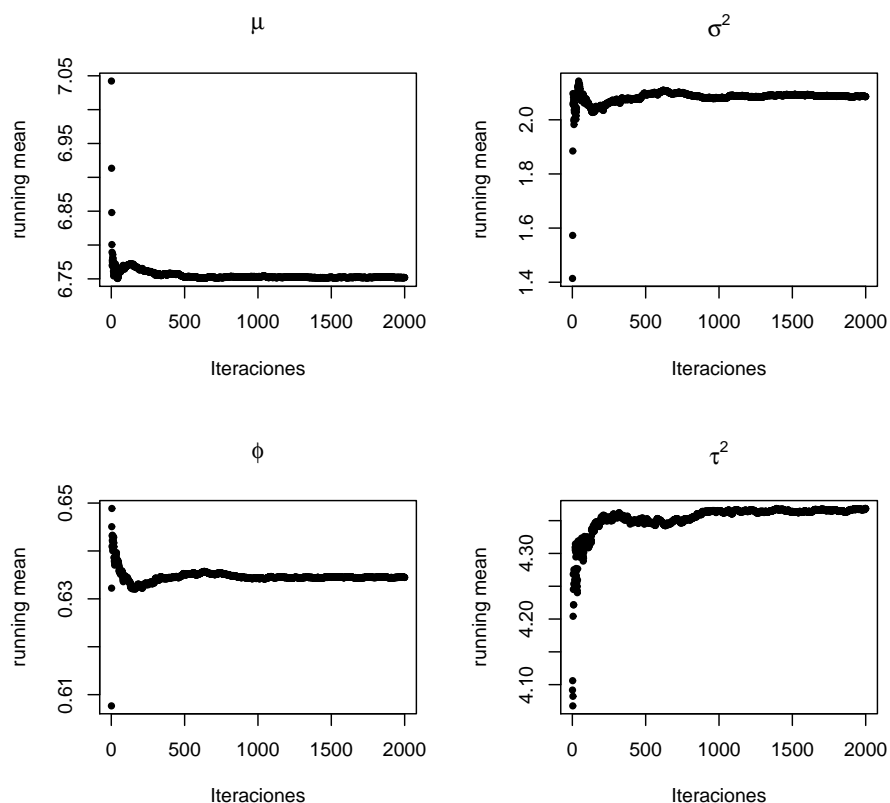
En la Figura 6.3, se muestra el ajuste de la serie mediante el método de muestreo de importancia (línea punteada).

## 6.2. Datos Zeger

En esta sección se aplican los métodos propuestos a una serie de tiempo sobre composición química de la deposición o degradación atmosférica. Particularmente las observaciones representan concentraciones mensuales de amoníaco ( $\text{NH}_4$ ) coleccionadas entre 1977 y 1980 en Lawrence Livermore, California, US, por The Environmental Measurements Laboratory. Estos datos fueron tomados del artículo de ? quienes ajustaron la serie de tiempo mediante un modelo de regresión para datos correlacionados usando un modelo autorregresivo de primer orden y considerando el tiempo (en meses) como covariable. ? mencionan que el objetivo principal fue estudiar diferencias geográficas y tendencias en el tiempo sobre la concentración de este contaminante.

Existen límites de detección inferiores en las pruebas lo cuales dependen de la cantidad total de precipitación de la composición química coleccionada en cada mes, volúmenes

## 6.2. Datos Zeger



**Figura 6.1:** Monitoreo de las medias (running means) para  $\mu$ ,  $\sigma^2$ ,  $\phi$  y  $\tau^2$

pequeños recolectados provocan un aumento en los límites de detección.

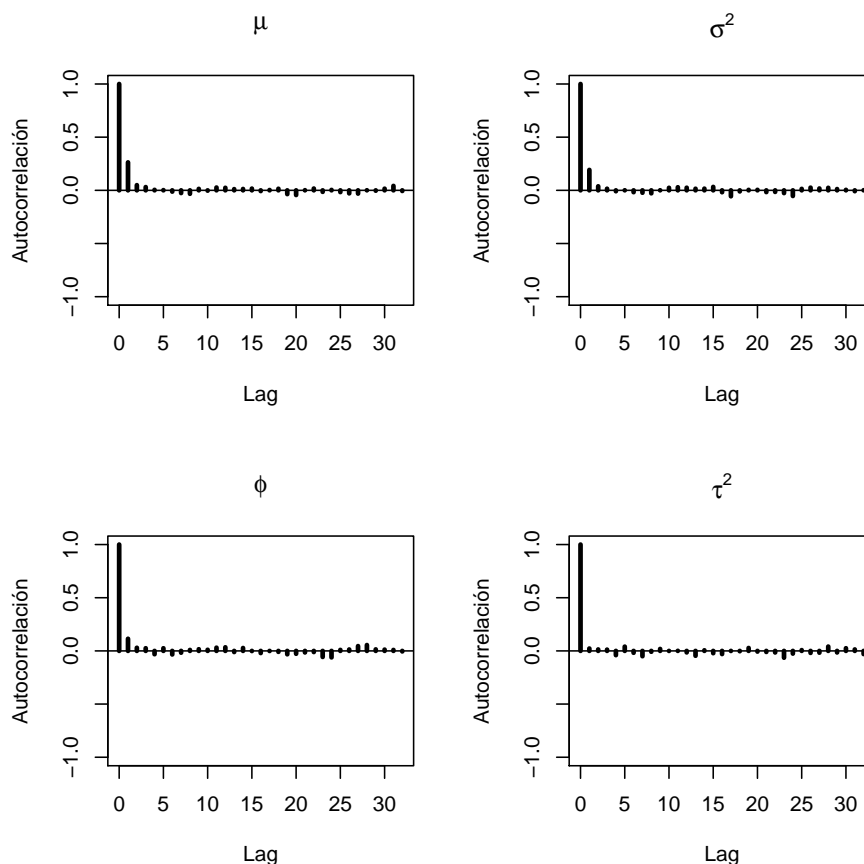
Los datos se listan en la Tabla 6.3 que consisten en 43 observaciones de los cuales 6 valores están censurados y 3 datos están perdidos, los cuales se denotan por NA, en total la serie tiene un 20.93% de información faltante. En la Figura 3.2 se muestra gráficamente la serie con círculos rellenos y los valores censurados con un triángulo con pico hacia abajo.

Para analizar los datos de la Tabla 6.3, ? propusieron el modelo siguiente

$$\lg(y_t) = \beta_0 + \beta_1 t + \varepsilon_t, \quad t = 1, 2, \dots, n, \quad (6.3)$$

donde  $t$  es la covariable que representa al tiempo en meses y  $\beta_0$ ,  $\beta_1$  son los coeficientes de regresión desconocidos. Este modelo supone que los errores  $\varepsilon_t$  provienen de un

## 6.2. Datos Zeger



**Figura 6.2:** Gráfica de autocorrelaciones de  $\mu, \sigma^2, \phi$  y  $\tau^2$

proceso estacionario autoregresivo de orden 1 que satisface

$$\varepsilon_t = \phi \varepsilon_{t-1} + a_t \quad (6.4)$$

donde las  $a_t$ 's  $\sim \text{iid } N(0, \tau^2)$ . No es difícil observar, que el modelo propuesto por ? en (6.3) se puede ver como un caso particular del modelo SSM lineal Gaussiano representado por las ecuaciones (5.1) y (5.2) con  $\mu_t = \beta_0 + \beta_1 t$  y  $\sigma^2 = 0$ .

En la Tabla 6.4 se presentan los estimadores de los parámetros del modelo SSM lineal Gaussiano mediante los métodos propuestos y se compara con los estimados por ?. En paréntesis se presenta el error estándar estimado para los métodos propuestos y un intervalo de confianza aproximado del 95 % para la pendiente en el caso de la estimación hecha por ?.

## 6.2. Datos Zeger

**Tabla 6.3:** Concentraciones mensuales de  $\text{NH}_4$  (mequiv/sq m) en Lawrence Livermore, California.

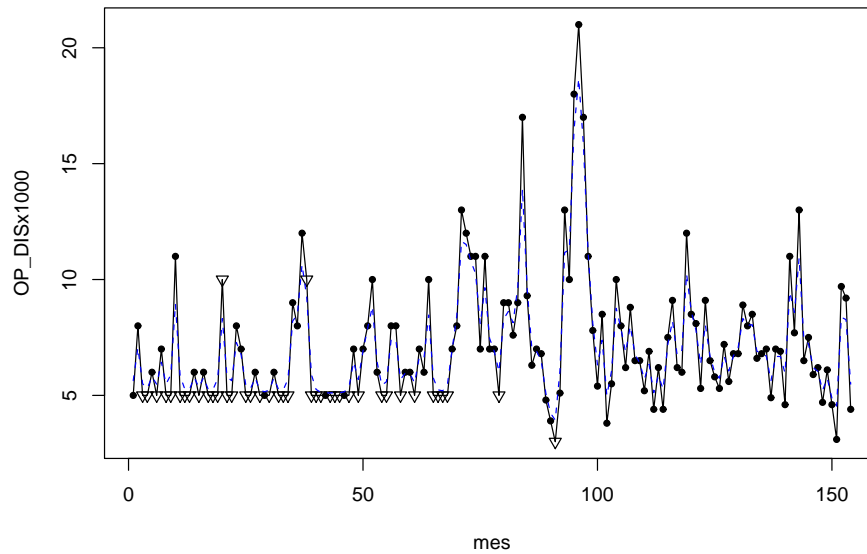
Mes	1977	1978	1979	1980
Enero		NA	34	117
Febrero		53	128	99
Marzo		546	79	<41
Abril		332	<31	29
Mayo	27	51	174	138
Junio	<16	203	<36	168
Julio	<19	171	<42	107
Agosto	796	321	169	253
Septiembre	268	402	223	445
Octubre	97	460	750	446
Noviembre	356	2080	NA	1260
Diciembre	95	487	NA	

**Tabla 6.4:** Estimadores EM, MCEM, SEM y IS de los parámetros del modelo SSM lineal Gaussiano para los datos presentados por ?.

Estimación	$\beta_1$	$\beta_2$	$\phi$	$\tau^2$
EM	4.38 (0.595)	0.026 (0.023)	0.42 (0.142)	1.33 (0.288)
MCEM	4.52 (0.083)	0.022 (0.0034)	0.38 (0.024)	1.20 (0.044)
SEM	4.52 (0.082)	0.022 (0.0033)	0.378 (0.023)	1.19 (0.043)
IS	5.03 (0.089)	0.025 (0.0035)	0.41 (0.022)	1.29 (0.042)
Z&B	5.02	0.015 (-.042,.066)	0.38	1.47

## 6.2. Datos Zeger

---

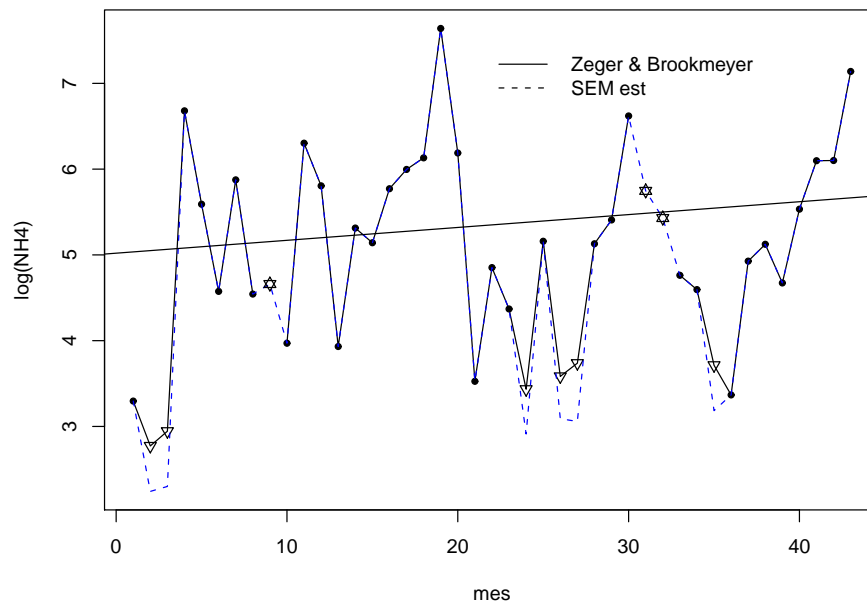


**Figura 6.3:** Serie de tiempo observada (línea sólida), serie de tiempo ajustada (línea punteada)

En la Figura 6.4 se presenta la serie de tiempo ajustada al modelo (línea punteada) y los valores perdidos estimados, así como también el ajuste presentado por ?.

## 6.2. Datos Zeger

---



**Figura 6.4:** Ajuste de la serie de tiempo de Concentraciones de  $\text{NH}_4$ .

# Capítulo 7

## Conclusiones

En este trabajo se ha estudiado el modelo de espacio de estados con observaciones censuradas (CenSSM). La función de verosimilitud de este modelo es una integral en  $\Re^{n+N}$ , donde  $n$  es el tamaño de la serie de tiempo y  $N$  el número de observaciones censuradas y/o perdidas. Excepto en casos triviales, aún en el caso sin censura, esta integral es difícil de obtener. En este trabajo para estimar los parámetros de este modelo se han considerado los algoritmos EM, MCEM, SEM y el muestreo de importancia. En el estudio de simulación que se realizó en un modelo de espacio de estados lineal Gaussiano con observaciones censuradas por la izquierda con niveles de censura bajo (10 %), medio (20 %) y alto (40 %), el sesgo que se obtiene de estos estimadores es razonable, aún para el caso de porcentaje alto de cesura. Como es de esperar, para el modelo CenSSM lineal Gaussiano, los métodos propuestos son computacionalmente intensivos; aunque el más rápido es el algoritmo EM, éste no se puede implementar para todos los modelos CenSSM tales como el caso no Guassianos o no lineales. El siguiente en rapidez es el algoritmo SEM. Los algoritmos MCEM y el muestreo de importancia son los más demandantes de tiempo.

El primer conjunto de datos reales que se estudió en el Capítulo 6 son 154 observaciones mensuales de fósforo en solución reactiva en un río tomadas en la estación 08C070 por el Departamento de Ecología del estado de Washington. El 26.62 % de las observaciones de esta serie de tiempo están censuradas. La segunda serie que se incluyó en éste capítulo de aplicaciones a datos reales, tomada en ?, son concentraciones mensuales de amoníaco ( $NH_4$ ) de la composición atmosférica para estudiar la composición química de la deposición o degradación atmosférica. Esta serie de tiem-

## 7. Conclusiones

---

po, además de observaciones censuradas, tiene datos perdidos. El modelo de regresión con errores autorregresivos de primer orden, considerado en ?, se puede representar como un modelo de espacio de estados con observaciones censuradas. Como se puede notar de las Figuras mostradas en el capítulo de aplicaciones, los procedimientos que se propusieron en este trabajo proporcionan buenos resultados.



# Apéndice

## Apéndice A: Función log verosimilitud del modelo de espacio de estados lineal Gaussiano

En esta sección describimos como se obtuvo la función log-verosimilitud del modelo SSM lineal Gaussiano en (4.7)–(4.8), usando las recursiones de Kalman.

Bajo las ecuaciones (4.7)–(4.8) se tiene que

$$f_{Y_t|\alpha_t}(y_t|\alpha_t) = \phi(y_t; \mu + \alpha_t, \sigma^2) \quad (\text{A.1})$$

$$f_{\alpha_t|\alpha_{t-1}}(\alpha_t|\alpha_{t-1}) = \phi(\alpha_t; \phi\alpha_{t-1}, \tau^2)$$

$$f(\alpha_1) = \phi(\alpha_1; 0, \text{Var}(\alpha_1)), \text{ donde } \text{Var}(\alpha_1) = \frac{\tau^2}{1 - \phi^2}$$

Donde  $\phi(y_t; \mu, \sigma^2)$  se utiliza para denotar a la función de densidad normal de la variables aleatoria  $Y_t$  con media  $\mu$  y varianza  $\sigma^2$ .

Entonces la función de verosimilitud está dada como en (3.18), es decir

$$L(\boldsymbol{\theta}; \mathbf{y}) = f_{Y_1}(y_1) \prod_{t=2}^n f(y_t|\mathbf{y}_{1:t-1}). \quad (\text{A.2})$$

donde

$$f(y_t|\mathbf{y}_{1:t-1}) = \int_{-\infty}^{\infty} f_{Y_t|\alpha_t}(y_t|\alpha_t) f(\alpha_t|\mathbf{y}_{1:t-1}) d\alpha_t \quad (\text{A.3})$$

Se sabe que  $f_{Y_t|\alpha_t}(y_t|\alpha_t) = \phi(y_t; \mu + \alpha_t, \sigma^2)$  y las densidades condicionales  $f(\alpha_t|\mathbf{y}_{1:t-1})$  y

$f(\alpha_t|\mathbf{y}_{1:t})$  se obtienen como en (3.11) y (3.8), *i.e.*,

$$f(\alpha_{t+1}|\mathbf{y}_{1:t}) = \int_{-\infty}^{\infty} f(\alpha_t|\mathbf{y}_{1:t})f(\alpha_{t+1}|\alpha_t) d\alpha_t \quad (\text{A.4})$$

donde  $f(\alpha_1|y_0) = f(\alpha_1)$ , y

$$f(\alpha_t|\mathbf{y}_{1:t}) = \frac{f(y_t|\alpha_t) f(\alpha_t|\mathbf{y}_{1:t-1})}{f(y_t|\mathbf{y}_{1:t-1})} \quad (\text{A.5})$$

En nuestro ejemplo (A.4) y (A.5), ambas tienen distribución normal, es decir,

$$f(\alpha_{t+1}|\mathbf{y}_{1:t}) = \phi(\alpha_{t+1}; \hat{\alpha}_{t+1}, \Omega_{t+1}) \quad (\text{A.6})$$

$$f(\alpha_t|\mathbf{y}_{1:t}) = \phi(\alpha_t; \alpha_{t|t}, \Omega_{t|t}) \quad (\text{A.7})$$

Calculando los argumentos  $\hat{\alpha}_{t+1}$ ,  $\Omega_{t+1}$ ,  $\alpha_{t|t}$  y  $\Omega_{t|t}$  se tiene

$$\begin{aligned} \hat{\alpha}_{t+1} &= E[\alpha_{t+1}|\mathbf{y}_{1:t}] = \int_{-\infty}^{\infty} \alpha_{t+1} f(\alpha_{t+1}|\mathbf{y}_{1:t}) d\alpha_{t+1} \\ &= \int_{-\infty}^{\infty} \alpha_{t+1} \int_{-\infty}^{\infty} f(\alpha_t|\mathbf{y}_{1:t}) f(\alpha_{t+1}|\alpha_t) d\alpha_t d\alpha_{t+1} \\ &= \int_{-\infty}^{\infty} f(\alpha_t|\mathbf{y}_{1:t}) \left[ \int_{-\infty}^{\infty} \alpha_{t+1} f(\alpha_{t+1}|\alpha_t) d\alpha_{t+1} \right] d\alpha_t \\ &= \int_{-\infty}^{\infty} f(\alpha_t|\mathbf{y}_{1:t}) \phi \alpha_t d\alpha_t \\ &= \phi \int_{-\infty}^{\infty} \alpha_t f(\alpha_t|\mathbf{y}_{1:t}) d\alpha_t \\ &= \phi E[\alpha_t|\mathbf{y}_{1:t}] \\ &= \phi \alpha_{t|t}. \end{aligned}$$

y

$$\Omega_{t+1} = \text{Var}(\alpha_{t+1}|\mathbf{y}_{1:t}) = E[\alpha_{t+1}^2|\mathbf{y}_{1:t}] - E[\alpha_{t+1}|\mathbf{y}_{1:t}]^2 = E[\alpha_{t+1}^2|\mathbf{y}_{1:t}] - \phi^2 \alpha_{t|t}^2 \quad (\text{A.8})$$

donde

$$\begin{aligned}
 E[\alpha_{t+1}^2 | \mathbf{y}_{1:t}] &= \int_{-\infty}^{\infty} \alpha_{t+1}^2 f(\alpha_{t+1} | \mathbf{y}_{1:t}) d\alpha_{t+1} \\
 &= \int_{-\infty}^{\infty} \alpha_{t+1}^2 \int_{-\infty}^{\infty} f(\alpha_t | \mathbf{y}_{1:t}) f(\alpha_{t+1} | \alpha_t) d\alpha_t d\alpha_{t+1} \\
 &= \int_{-\infty}^{\infty} f(\alpha_t | \mathbf{y}_{1:t}) \left[ \int_{-\infty}^{\infty} \alpha_{t+1}^2 f(\alpha_{t+1} | \alpha_t) d\alpha_{t+1} \right] d\alpha_t \\
 &= \int_{-\infty}^{\infty} f(\alpha_t | \mathbf{y}_{1:t}) E[\alpha_{t+1}^2 | \alpha_t] d\alpha_t \\
 &\quad \parallel \text{ya que } \text{Var}(\alpha_{t+1} | \alpha_t) = \tau^2, \text{ entonces } E[\alpha_{t+1}^2 | \alpha_t] = \tau^2 + \phi^2 \alpha_t^2 \parallel \\
 &= \tau^2 + \phi^2 \int_{-\infty}^{\infty} \alpha_t^2 f(\alpha_t | \mathbf{y}_{1:t}) d\alpha_t = \tau^2 + \phi^2 E[\alpha_t^2 | \mathbf{y}_{1:t}] \\
 &= \tau^2 + \phi^2 \{ \text{Var}(\alpha_t | \mathbf{y}_{1:t}) + E[\alpha_t | \mathbf{y}_{1:t}] \} = \tau^2 + \phi^2 \Omega_{t|t} + \phi^2 \alpha_{t|t}.
 \end{aligned}$$

Sustituyendo la expresión anterior en (A.8) se tiene

$$\begin{aligned}
 \Omega_{t+1} &= \tau^2 + \phi^2 \Omega_{t|t} + \phi^2 \alpha_{t|t} - \phi^2 \alpha_{t|t}^2 \\
 &= \tau^2 + \phi^2 \Omega_{t|t}.
 \end{aligned}$$

Sustituyendo las correspondientes densidades (A.1) y (A.7) en (A.5) e igualando el coeficiente de  $\alpha_t^2$  en ambos lados de (A.5) se obtiene que

$$\begin{aligned}
 \frac{\alpha_t^2}{\Omega_{t|t}} &= \frac{\alpha_t^2}{\sigma^2} + \frac{\alpha_t^2}{\Omega_t} \\
 \Omega_{t|t}^{-1} &= \frac{1}{\sigma^2} + \Omega_t^{-1},
 \end{aligned}$$

Análogamente para el término no cuadrático de  $\alpha_t$  y dividiendo ambos lados de la ecuación por  $-2\alpha_t$ , se tiene

$$\begin{aligned}
 -\frac{2\alpha_t \alpha_{t|t}}{\Omega_{t|t}} &= -\frac{2\alpha_t y_t}{\sigma^2} + \frac{2\alpha_t \mu}{\sigma^2} - \frac{2\alpha_t \hat{\alpha}_t}{\Omega_t} \\
 \alpha_{t|t} \Omega_{t|t}^{-1} &= \frac{y_t - \mu}{\sigma^2} + \frac{\hat{\alpha}_t}{\Omega_t}
 \end{aligned}$$

pero se sabe que  $\Omega_{t|t}^{-1} = \frac{1}{\sigma^2} + \Omega_t^{-1}$ , entonces

$$\begin{aligned}
 \alpha_{t|t} \Omega_{t|t}^{-1} &= \frac{y_t - \mu}{\sigma^2} + \hat{\alpha}_t (\Omega_{t|t}^{-1} - \frac{1}{\sigma^2}) \\
 &= \frac{y_t - \mu - \hat{\alpha}_t}{\sigma^2} + \hat{\alpha}_t \Omega_{t|t}^{-1}
 \end{aligned}$$

por lo tanto,

$$\alpha_{t|t} = \hat{\alpha}_t + \frac{\Omega_{t|t}}{\sigma^2}(y_t - \mu - \hat{\alpha}_t)$$

Sustituyendo (A.1) y (A.6) en (A.3), se tiene que

$$f(y_t|\mathbf{y}_{1:t-1}) = \mathcal{N}(\mu + \hat{\alpha}_t, \Omega_t + \sigma^2)$$

y la log-verosimilitud del modelo es

$$\begin{aligned} l(\boldsymbol{\theta}; \mathbf{y}) &= \log L(\boldsymbol{\theta}; \mathbf{y}) \\ &= \sum_{t=1}^n \log f(y_t|\mathbf{y}_{1:t-1}) \\ &= -\frac{n}{2} \log(2\pi) - \frac{1}{2} \sum_{t=1}^n \log(\Omega_t + \sigma^2) - \frac{1}{2} \sum_{t=1}^n \frac{(w_t - \mu - \hat{\alpha}_t)^2}{(\Omega_t + \sigma^2)}. \end{aligned}$$

## Apéndice B: Distribución de la función de importancia

En este apartado se desea conocer la densidad condicional conjunta de  $g_{ic}(\boldsymbol{\alpha}|\mathbf{y})$  considerando el modelo de espacio de estados lineal Gaussiano dado en (4.7)–(4.8).

Es posible deducir  $g_{ic}(\boldsymbol{\alpha}|\mathbf{y})$  a partir de (4.24), donde  $g_{ic}(\alpha_t|\alpha_{t+1:n}, \mathbf{y}_{1:n})$  se obtiene como

$$\begin{aligned} g_{ic}(\alpha_t|\alpha_{t+1:n}, \mathbf{y}_{1:n}) &= f(\alpha_t|\alpha_{t+1}, \mathbf{y}_{1:t}) \\ &= \frac{f(\alpha_t|\mathbf{y}_{1:t})f(\alpha_{t+1}|\alpha_t)}{f(\alpha_{t+1}|\mathbf{y}_{1:t})} \\ g_{ic}(\alpha_t|\alpha_{t+1:n}, \mathbf{y}_{1:n}) &\propto f(\alpha_t|\mathbf{y}_{1:t})f(\alpha_{t+1}|\alpha_t) \end{aligned} \quad (\text{B.9})$$

Note que

$$\begin{aligned} f(\alpha_{t+1}|\alpha_t) &= \mathcal{N}(\phi\alpha_t, \tau^2) \\ f(\alpha_t|\mathbf{y}_{1:t}) &= \mathcal{N}(\alpha_{t|t}, \Omega_{t|t}) \end{aligned}$$

por lo que se puede ver que (B.9) se distribuye también como una normal, i.e.  $\mathcal{N}(\mu_t^*, \nu_t^*)$ .

Ahora de calcular los argumentos  $\mu_t^*$  y  $\nu_t^*$ . Resolviendo para  $\alpha_t$ , la ecuación  $\frac{\partial}{\partial \alpha_t} \log g_{ic}(\alpha_t|\alpha_{t+1:n}, \mathbf{y}_{1:n}) = 0$ , se obtiene  $\mu_t^*$ , como sigue

$$\begin{aligned} 0 &= \frac{\partial}{\partial \alpha_t} \log g_{ic}(\alpha_t|\alpha_{t+1:n}, \mathbf{y}_{1:n}) \\ &= \frac{\partial}{\partial \alpha_t} \left\{ -\frac{1}{2} \log(2\pi) - \frac{1}{2} \log(\Omega_{t|t}) - \frac{1}{2} \frac{(\alpha_t - \alpha_{t|t})^2}{\Omega_{t|t}} \right. \\ &\quad \left. - \frac{1}{2} \log(2\pi) - \frac{1}{2} \log(\tau^2) - \frac{1}{2} \frac{(\alpha_{t+1} - \phi\alpha_t)^2}{\tau^2} \right\} \\ &= -\frac{(\alpha_t - \alpha_{t|t})}{\Omega_{t|t}} - \frac{(\alpha_{t+1} - \phi\alpha_t)}{\tau^2} (-\phi) \\ &= -\frac{(\alpha_t - \alpha_{t|t})}{\Omega_{t|t}} + \frac{\phi(\alpha_{t+1} - \phi\alpha_t)}{\tau^2} \\ &= -\frac{\alpha_t}{\Omega_{t|t}} - \frac{\phi\alpha_t}{\tau^2} + \frac{\phi\alpha_{t+1}}{\tau^2} + \frac{\alpha_{t|t}}{\Omega_{t|t}} \\ &= -\alpha_t \left( \frac{\tau^2 + \phi^2\Omega_{t|t}}{\Omega_{t|t}\tau^2} \right) + \frac{\phi\alpha_{t+1}\Omega_{t|t} + \tau^2\alpha_{t|t}}{\tau^2\Omega_{t|t}} \end{aligned}$$

por lo tanto,

$$\alpha_t = \frac{\Omega_{t|t}\tau^2}{\tau^2 + \phi^2\Omega_{t|t}} \left( \frac{\phi\alpha_{t+1}\Omega_{t|t} + \tau^2\alpha_{t|t}}{\tau^2\Omega_{t|t}} \right)$$

$$\mu_t^* = \frac{\phi\alpha_{t+1}\Omega_{t|t} + \tau^2\alpha_{t|t}}{\tau^2 + \phi^2\Omega_{t|t}}.$$

De manera similar se procede para  $\nu_t^*$ ,

$$\begin{aligned} \nu_t^* &= \left[ -\frac{\partial^2}{\partial\alpha_t^2} \log g_{ic}(\alpha_t|\alpha_{t+1:n}, \mathbf{y}_{1:n}) \right]^{-1} \\ &= \left[ -\frac{\partial^2}{\partial\alpha_t^2} \left\{ -\frac{\alpha_t}{\Omega_{t|t}} - \frac{\phi\alpha_t}{\tau^2} + \frac{\phi\alpha_{t+1}}{\tau^2} + \frac{\alpha_{t|t}}{\Omega_{t|t}} \right\} \right]^{-1} \\ &= \left[ \frac{1}{\Omega_{t|t}} + \frac{\phi^2}{\tau^2} \right]^{-1} \\ &= \left[ \frac{\tau^2 + \phi^2\Omega_{t|t}}{\tau^2\Omega_{t|t}} \right]^{-1} \\ &= \frac{\tau^2\Omega_{t|t}}{\tau^2 + \phi^2\Omega_{t|t}}. \end{aligned}$$

## Apéndice C: Funciones y rutinas de código realizados en R y Fortran

En este apartado presentamos el código que se realizó en el programa estadístico R y algunas subrutinas realizadas en Fortran que fueron llamadas en forma de Dyanmic Link DLL's a R, todo esto con el propósito de hacer más eficiente el trabajo computacional de la investigación.

### 1. Librerías de R utilizadas

```
library(msm)
library(mvtnorm)
library(numDeriv)
library(coda)
```

### 2. Funciones comunes que se utilizaron a lo largo del trabajo

```
# Función para generar un proceso AR(1) de la forma
# a_t = phi*a_t-1 + n_t; , n_t ~iid N(0,sqrt(var))
AR1.sim<-function(n,par,cal=100)
#par: vector de parametros=(phi,sd)
{
  m<-n+cal
  phi<-par[1]
  sd<-par[2]
  x<-rep(0,m)
  for (i in 2:m)
  {
    x[i]<-phi*x[i-1]+rnorm(1,0,sd)
  }
  x[(cal+1):m]
}

#Función que censura la serie en L (D=F)
st.cen<-function(st,L,dir=FALSE)
# st:serie de tiempo; L:punto de censura
# dir:direccion de censura (FALSE censura datos por izquierda en L)
{
  if(dir==F)
  {
    st[st<=L]<-L
    return(st)
  }
  else
  {
    st[st>=L]<-L
    return(st)
  }
}
```

## Apéndice

---

```
#Función para obtener el vector indicador de censura
pos<-function(stc,L)
#stc: serie de tiempo censurada en L
# 0:censurada; 1: no-censurada
{
  n<-length(stc)
  p<-rep(1,n)
  for (i in 1:n)
  {
    if (stc[i]==L) p[i]<-0
  }
  pc<-(n-sum(p))*100/n #porcentaje de censura, p: vector indicador
  list(delta=p,PC=pc)
}

#Función para las recursiones de Kalman para las Predicciones y filtro de alpha (variable de estado)..
pyf.Alpha<-function(serie,par)
#y: serie observada (completa)
#par: vector de parámetros (sigma2,phi,tau2)
{
  mu<-par[1]; sigma2<-par[2]
  phi<-par[3]; tau2<-par[4]
  n<-length(serie)
  omega.f<-alpha.f<-omega.p<-alpha.p<-arg<-mut<-rep(0,n)
  #condiciones iniciales
  alpha.p[1]<-0
  omega.p[1]<-tau2/(1-phi^2)
  for (t in 1:(n-1))
  {
    omega.f[t]<-sigma2*omega.p[t]/(sigma2+omega.p[t])
    omega.p[t+1]<-phi^2*omega.f[t]+tau2
    alpha.f[t]<-alpha.p[t]+(omega.f[t]/sigma2)*(serie[t]-alpha.p[t]-mu)
    alpha.p[t+1]<-phi*alpha.f[t]
  }
  omega.f[n]<-sigma2*omega.p[n]/(sigma2+omega.p[n])
  alpha.f[n]<-alpha.p[n]+(omega.f[n]/sigma2)*(serie[n]-alpha.p[n]-mu)
  list(alpha.p=alpha.p,alpha.f=alpha.f,omega.f=omega.f,omega.p=omega.p)
}

#Suavizamiento para alpha..
sualpha<-function(alfaf,omegaf,phi,tau2)
{
  n<-length(alfaf)
  alfa.s<-rep(NA,n)
  alfa.s[n]<-alfaf[n]
  for (j in (n-1):1)
  {
    alfa.s[j]<-(tau2*alfaf[j]+phi*omegaf[j]*alfa.s[j+1])/(tau2+phi^2*omegaf[j])
  }
  alfa.s
}

#Funciones especiales para implementar el algoritmo EM
```



```

#funcion para obtener  $E(Z-\mu-\alpha.p)^2$ 
EZ.2<-function(serie,par,delta)
#y: serie observada (censurada)
#par: vector de parámetros ( $\mu$ ,  $\sigma^2$ ,  $\phi$ ,  $\tau^2$ )
#deta: vector de indica la censura en las observaciones
{
  mu<-par[1]; sigma2<-par[2]
  phi<-par[3]; tau2<-par[4]
  n<-length(serie)
  EZ.2<-EZ<-omega.f<-alpha.f<-omega.p<-alpha.p<-rep(0,n)
  #condiciones iniciales
  alpha.p[1]<-0
  omega.p[1]<-tau2/(1-phi^2)
  for (t in 1:(n-1))
  {
    if(delta[t]==0)
    {
      Lt<-(serie[t]-mu-alpha.p[t])/sqrt(sigma2+omega.p[t])
      Ht<-dnorm(Lt)/pnorm(Lt)
      EZ.2[t]<-omega.p[t]+sigma2+sqrt(omega.p[t]+sigma2)*Ht*(mu+alpha.p[t]-serie[t])
      EZ[t]<-mu+alpha.p[1]-sqrt(omega.p[t]+sigma2)*Ht
    }
    else{
      EZ[t]<-serie[t]
      EZ.2[t]<-(serie[t]-mu-alpha.p[t])^2
    }
    omega.f[t]<-sigma2*omega.p[t]/(sigma2+omega.p[t])
    omega.p[t+1]<-phi^2*omega.f[t]+tau2
    alpha.f[t]<-alpha.p[t]+(omega.f[t]/sigma2)*(EZ[t]-mu-alpha.p[t])
    alpha.p[t+1]<-phi*alpha.f[t]
  }
  if(delta[n]==0)
  {
    Lt<-(serie[n]-mu-alpha.p[n])/sqrt(sigma2+omega.p[n])
    Ht<-dnorm(Lt)/pnorm(Lt)
    EZ.2[n]<-sqrt(omega.p[n]+sigma2)*Ht*(mu+alpha.p[n]-serie[n])+omega.p[n]+sigma2
    EZ[n]<-mu+alpha.p[n]-sqrt(omega.p[n]+sigma2)*Ht
  }
  else
  {
    EZ[n]<-serie[n]
    EZ.2[n]<-(serie[n]-mu-alpha.p[n])^2
  }
  list(EZ.2=EZ.2,EZ=EZ)
}

#funcion para obtener la función  $Q(\theta | \theta(j))$  en el algoritmo EM
fQ.EM<-function(EZ.2,EZ,par)
#y: serie observada (censurada)
#par: vector de parámetros ( $\mu$ ,  $\sigma^2$ ,  $\phi$ ,  $\tau^2$ )
#deta: vector de indica la censura en las observaciones
{
  mu<-par[1]; sigma2<-par[2]
  phi<-par[3]; tau2<-par[4]
  n<-length(EZ)

```

```

#Recursiones de Kalman de un paso (Generalized SSM)
fQ<-omega.f<-alpha.f<-omega.p<-alpha.p<-rep(0,n)
#condiciones iniciales
alpha.p[1]<-0
omega.p[1]<-tau2/(1-phi^2)
for (t in 1:(n-1))
{
omega.f[t]<-sigma2*omega.p[t]/(sigma2+omega.p[t])
omega.p[t+1]<-phi^2*omega.f[t]+tau2
alpha.f[t]<-alpha.p[t]+(omega.f[t]/sigma2)*(EZ[t]-mu-alpha.p[t])
alpha.p[t+1]<-phi*alpha.f[t]
}
#Funcion Q
-(-n/2*log(2*pi)-1/2*sum(log(omega.p+sigma2))-1/2*sum(EZ.2/(omega.p+sigma2)))
#fQ<-(-n/2*log(2*pi)-1/2*sum(log(omega.p+sigma2))-1/2*sum(EZ.2/(omega.p+sigma2)))
#EZ<-EZ
#list(fQ=fQ,EZ=EZ)
}

#Funcion que genera las 'm' muestras(series) de tamaño 'n',
#en el paso [1] del EM monte-carlo. En Fortran
rZa.f<-function(stc,par,delta,m)
# stc: serie de tiempo censurada; par: parametros.i
# delta: vector de censura; m: No de rep Monte-Carlo
{
t(replicate(m,fzy.f(stc,par,delta)$z))
}

```

### 3. Llamando las DLL's de Fortran a R

```

#Llamando la función de verosimilitud de Fortran
#ruta de la DLL..
setwd("D:/DocumentosARIZAHFJ/TESIS DE DOCTORADO/fvMLG/Release")
dyn.load("fvMLG.dll") #Llama la DLL
is.loaded(symbol.For("fvmlg")) #Verifica si llamo la DLL correctamente
#funcion
fvm.f<-function(par,serie)
{
fi<-0.0
lp<-length(par)
n<-length(serie)
.Fortran("fvmlg",p=as.integer(lp), par=as.double(par),n=as.integer(n)
,f=as.double(fi),serie=as.double(serie))$f
}

#funcion para obtener muestras de la distribución
# condicional conjunta f(z|y)
setwd("D:/DocumentosARIZAHFJ/TESIS DE DOCTORADO/samfzy/Release")
dyn.load("samfzy.dll") #Llama la DLL
is.loaded(symbol.For("fzy")) #Verifica si llamo la DLL correctamente

#dyn.unload("samfzy.dll") #desactiva la .DLL

fzy.f<-function(serie,par,delta)

```

```
{
  n<-length(serie)
  z<-rep(0,n)
  .Fortran("fzy",serie=as.double(serie),par=as.double(par),delta=as.double(delta)
          ,n=as.integer(n),z=as.double(z),y=as.double(1.0))
}

#funcion Q monte carlo
setwd("D:/DocumentosARIZAHFJ/TESIS DE DOCTORADO/fQ/Release")
dyn.load("fQ.dll") #Llama la DLL
is.loaded(symbol.For("fqlg")) #Verifica si llamo la DLL correctamente

#dyn.unload("fQ.dll") #desactiva la .DLL

fQ.f<-function(z,teta)
{
  n1<-dim(z)
  n<-n1[2]
  m<-n1[1]
  .Fortran("fqlg",par=as.double(teta),z=as.double(z),n=as.integer(n),m=as.integer(m),
          fp=as.double(0.0))$fp
}
```

#### 4. Calculo del sesgo y error cuadrático mediante el algoritmo EM

```
load(file="D:/DocumentosARIZAHFJ/TESIS DE DOCTORADO/TS.RData")
#S<-dim(TS)[1]
#Empieza el calculo del sesgo..
eEM05<-matrix(0,S,length(theta))
for (j in 1:S)
{
  serie<-TS[j, ]
  secen05<-st.cen(serie,L.cen05,dir=F) #Censurando la serie al 5%
#vector de posiciones
d<-pos(secen05,L.cen05)
delta05<-d$delta
#semillas
e05<-nlminb(c(1,1,0.1,1),fvm.f,serie=secen05,lower=l.inf,upper=l.sup)
#e05 #estimador semilla
#Inicia la ejecución del algoritmo EM..
#valores de arranque
num.iter<-1
itermax<-100
epsilon<-5
guess<-e05$par
while(epsilon >= 0.001 & num.iter <= itermax)
{
  Zt<-EZ.2(secen05,guess,delta05)
  EM.e<-optim(guess,fvm.f,serie=Zt$EZ,lower=l.inf,upper=l.sup,hessian=T)
  #cat(num.iter, EM.e$par, EM.e$value, "\n")
  #cat(EM.e1$par, EM.e1$value, "\n")
  #epsilon<-max(abs(EM.e$par-guess))
  num.iter<-num.iter+1
  guess<-EM.e$par
}
```

```

}
eEM05[,j]<-EM.e$par
}
save(eEM05,file="D:/DocumentosARIZAHFJ/TESIS DE DOCTORADO/eEM05.RData")
sesgo05<-apply(eEM05,2,mean)-theta
sesgo05
#Calculando error cuadrático medio
V<-diag(var(eEM05))
ecm05<-V+sesgo05^2
ecm05

#cálculo del sesgo y ecm median los estimadores mcem
for(i in 1:500)
{
serie<-TS[i, ]
secen05<-st.cen(serie,L.cen05,dir=F) #Censurando la serie al 5%
delta05<-pos(secen05,L.cen05)$delta

#semillas
semi<-nlminb(c(mean(secen05),1,0.1,1),fvm.f,serie=secen05,lower=l.inf,upper=l.sup)
#semi #estimador semilla

#Repeticiones Monte Carlo para calentamiento:
M<-c(10,200,500,rep(1000,3))
mm<-length(M)
est.f<-rep(0,mm)
guess1<-semi$par #estimador semilla
est.p<-matrix(0,mm,length(guess1))

#system.time(
for(j in 1:mm)
{
#Z1<-rZa(secen05,guess1,delta05,m=10)
Z1<-rZa.f(secen05,guess1,delta05,m=M[j])
#est.a<-optim(guess1,fQ.mc1af,z=Z1,lower=l.inf,upper=l.sup,hessian=T)
#est.a<-optim(guess1,fQ.f,z=Z1,lower=l.inf,upper=l.sup,hessian=T)
est.p[j,]<-est.a$par
guess1<-est.a$par
H<-est.a$hessian
}
#)
#est.p
n<-length(secen05)
s1<-max(diag(sqrt(solve(H)/n)))
if(is.finite(s1)==F | s1 > 0.07) s1<-0.07
m1<-1000
lamda<-0.01
mf<-ceiling(m1*s1/lamda)
#mf
#valores de arranque
num.iter<-1
itermax<-6
epsilon<-5
guess<-est.p[length(M), ]
#system.time(

```

```

while(epsilon >= 0.001 & num.iter <= itermax)
{
  Z1<-rZa.f(secen05,guess,delta05,m=mf)
  fq1<-fQ.f(Z1,guess)
  mcEM.e<-nlminb(guess,fQ.f,z=Z1,lower=l.inf,upper=l.sup)
  #cat(num.iter, EM.e$par, EM.e$value, "\n")
  #cat(EM.e1$par, EM.e1$value, "\n")
  epsilon<-abs(mcEM.e$objective-fq1)
  num.iter<-num.iter+1
  est.p<-rbind(est.p,mcEM.e$par)
  guess<-mcEM.e$par
}
#)
e.mcem05[i,]<-est.p[dim(est.p)[1],]
print(i)
}

```

### 5. Funciones para implementar el algoritmo MCEM

```

#Lamando la DLL para muestrear de la distribución predictiva
setwd("D:/DocumentosARIZAHFJ/TESIS DE DOCTORADO/samfzy/Release")
dyn.load("samfzy.dll") #Llama la DLL
is.loaded(symbol.For("fzy")) #Verifica si llamo la DLL correctamente

#dyn.unload("samfzy.dll") #desactiva la .DLL

fzy.f<-function(serie,par,delta)
{
  n<-length(serie)
  z<-rep(0,n)
  .Fortran("fzy",serie=as.double(serie),par=as.double(par),delta=as.double(delta)
          ,n=as.integer(n),z=as.double(z),y=as.double(1.0))
}

#####3
#Función que genera las 'm' muestras(series) de tamaño 'n',
#en el paso [1] del EM monte-carlo.
#-----

rZa.f<-function(stc,par,delta,m)
# stc: serie de tiempo censurada; par: parametros.i
# delta: vector de censura; m: No de rep Monte-Carlo
{
  t(replicate(m,fzy.f(stc,par,delta)$z))
}

# Función que aproxima mediante integración monte-carlo
# a la función Q(theta|theta.i) el algoritmo EM.
#-----
fQ.mc1af<-function(z,theta)

```

## Apéndice

---

```
# stc: serie de tiempo censurada; par: parametros.i
# delta: vector de censura; m: No de rep Monte-Carlo
# z: matrix que contiene las 'm' muestra de los 'datos completos'
{
mean(apply(z,1,fvm.f,par=theta))
}

#en fortran

setwd("D:/DocumentosARIZAHFJ/TESIS DE DOCTORADO/fQ/Release")
dyn.load("fQ.dll") #Llama la DLL
is.loaded(symbol.For("fqlg")) #Verifica si llamo la DLL correctamente

#dyn.unload("fQ.dll") #desactiva la .DLL

fQ.f<-function(z,teta)
{
  n1<-dim(z)
  n<-n1[2]
  m<-n1[1]
  .Fortran("fqlg",par=as.double(teta),z=as.double(z),n=as.integer(n),m=as.integer(m),
          fp=as.double(0.0))$fp
}

#LOS DATOS...
#-----

n<-500 #tamaño de la serie
theta<-c(30,2,0.9,1) #vector de parametros poblacionales propuestos

#Llamando las series..

load(file="D:/DocumentosARIZAHFJ/TESIS DE DOCTORADO/TS.RData")
TS

#Limites de censura..
L.cen05<-25.66744
L.cen10<-26.61427
L.cen20<-27.77274
L.cen40<-29.33665
#-----
#ESTIMACION CON 5, 10, 20 y 40% DE CENSURA

n<-500 #tamaño de la serie
theta<-c(30,2,0.9,1) #vector de parámetros poblacionales propuestos

#Llamando las series..

load(file="D:/DocumentosARIZAHFJ/TESIS DE DOCTORADO/TS.RData")
TS

#espacio paramétrico
l.inf=c(-Inf,0,-.999999,0); l.sup=c(Inf,Inf,.99999,Inf)

e.mcem10<-matrix(0,500,4)
```

```
i<-22

for(i in 1:500)
{
serie<-TS[i, ]
secen10<-st.cen(serie,L.cen10,dir=F) #Censurando la serie al 10%
delta10<-pos(secen10,L.cen10)$delta

#semillas
semi<-nlminb(c(mean(secen10),1,0.1,1),fvm.f,serie=secen10,lower=l.inf,upper=l.sup)
#semi #estimador semilla

#Repeticiones monteCarlo para calentamiento:
M<-c(10,100,200,rep(500,2),rep(1000,3))
mm<-length(M)
est.f<-rep(0,mm)
guess1<-semi$par #estimador semilla
est.p<-matrix(0,mm,length(guess1))

#system.time( j=3
for(j in 1:mm)
{
Z1<-rZa.f(secen10,guess1,delta10,m=M[j])
est.a<-optim(guess1,fQ.f,z=Z1,lower=l.inf,upper=l.sup,hessian=T)
est.p[j,]<-est.a$par
guess1<-est.a$par
H<-est.a$hessian
}
#)
#est.p
n<-length(secen10)
s1<-max(diag(sqrt(solve(H)/n)))
if(is.finite(s1)==F | s1 > 0.07) s1<-0.07
m1<-1000
lamda<-0.01
mf<-ceiling(m1*s1/lamda)
#mf
#valores de arranque
num.iter<-1
itermax<-6
epsilon<-5
guess<-est.p[length(M), ]
#system.time(
while(epsilon >= 0.001 & num.iter <= itermax)
{
Z1<-rZa.f(secen10,guess,delta10,m=mf)
fQ1<-fQ.f(Z1,guess)
mcEM.e<-nlminb(guess,fQ.f,z=Z1,lower=l.inf,upper=l.sup)
#cat(num.iter, EM.e$par, EM.e$value, "\n")
#cat(EM.e1$par, EM.e1$value, "\n")
epsilon<-abs(mcEM.e$objective-fQ1)
num.iter<-num.iter+1
est.p<-rbind(est.p,mcEM.e$par)
guess<-mcEM.e$par
```

```
}
#)
e.mcem10[i,]<-est.p[dim(est.p)[1],]
save(e.mcem10,file="D:/DocumentosARIZAHFJ/TESIS DE DOCTORADO/e.mcem10.RData")
print(i)
}

#Calculando el sesgo
sesgo10<-apply(e.mcem10,2,mean)-theta
sesgo10

#Calculando el ecm
V<-diag(var(e.mcem10))
ecm10<-V+sesgo10^2
ecm10
```

### 6. Funciones para implementar el algoritmo SEM

```
#Packages que utilizamos
library(msm) #normal truncada
library(mvtnorm) #normal multivariada
library(coda) #Diagnostic's methods for convergence in MCMC
library(maxLik) #Derivadas numéricas
library(numDeriv) #Derivadas numéricas

#Llamando la función de verosimilitud de Fortran
#ruta de la DLL..
setwd("D:/DocumentosARIZAHFJ/TESIS DE DOCTORADO/fvMLG/Release")
dyn.load("fvMLG.dll") #Llama la DLL
is.loaded(symbol.For("fvmlg")) #Verifica si llamo la DLL correctamente
is.loaded(symbol.For("recibe")) #Verifica si llamo la DLL correctamente

fvm.f<-function(par,serie)
{
  fi<-0.0
  lp<-length(par)
  n<-length(serie)
  .Fortran("fvmlg",p=as.integer(lp), par=as.double(par),n=as.integer(n),f=as.double(fi),serie=as.double(serie))$f
}

# Función que calcula las los promedio acumulados
running.mean<-function(x)
# x: Cadena de Markov
{
  n<-length(x)
  rmean<-cumsum(x)/(1:n)
  rmean
}

# DATOS
# -----#-----#-----#-----#
#Llamando las series..
load(file="D:/DocumentosARIZAHFJ/TESIS DE DOCTORADO/TS.RData")
```



## Apéndice

---

```
TS

#espacio parametrico
l.inf=c(-Inf,0,-.999999,0); l.sup=c(Inf,Inf,.99999,Inf)

#Límites de censura..
L.cen05<-25.66744
L.cen10<-26.61427
L.cen20<-27.77274
L.cen40<-29.33665

#####
#ESTIMACION CON 5, 10, 20 y 40% DE CENSURA EN LA SERIE
#-----

#inicio el proceso de estimación SEM
theta=c(30,2,0.9,1);n=500
m<-1600 #tamaño inicial de la cadena
p<-length(theta)
sem05<-matrix(0,m,p)

e.sem05<-matrix(0,500,p)

for(i in 1:500)
{
  #serie y stc
  serie<-TS[i, ]
  L.cen05<-25.66744
  secen05<-st.cen(serie,L.cen05,dir=F) #Censurando la serie al 10%
  delta05<-pos(secen05,L.cen05)$delta

  #semillas
  semi<-nlminb(c(mean(secen05),1,0.1,1),fvm.f,serie=secen05,lower=l.inf,upper=l.sup)
  semi

  sem05[1,]<-semi$par;

  for(j in 1:(m-1))
  {
    z05<-fZ.y(secen05,sem05[j,],delta05)$z
    e1<-nlminb(sem05[j,],fvm.f,serie=z05,lower=l.inf,upper=l.sup)
    sem05[j+1,]<-e1$par
  }

  ##Diagnostico de Raftery-Lewis
  c.mu<-as.mcmc(sem05[,1])
  c.sigma2<-as.mcmc(sem05[,2])
  c.phi<-as.mcmc(sem05[,3])
  c.tau2<-as.mcmc(sem05[,4])
  a1<-raftery.diag(c.mu,q=0.5,r=0.025,s=0.95)
  a2<-raftery.diag(c.sigma2,q=0.5,r=0.025,s=0.95)
  a3<-raftery.diag(c.phi,q=0.5,r=0.025,s=0.95)
  a4<-raftery.diag(c.tau2,q=0.5,r=0.025,s=0.95)
}
```

```
#Determinando el tamaño verdadero de la cadena
le<-max(a1$resmatrix[2], a2$resmatrix[2], a3$resmatrix[2], a4$resmatrix[2])
#le

if(le > 1600) #Si es necesario aumentar las cadenas
{
  m1<-le-1600
  sem05.com<-matrix(0,m1,4)
  sem05.com[1,]<-sem05[1600,]
  for(j in 1:(m1-1))
  {
    z<-fZ.y(secen05,sem05.com[j,],delta05)$z
    e1<-nlminb(sem05.com[j,],fvm.f,serie=z,lower=1.inf,upper=1.sup)
    sem05.com[j+1,]<-e1$par
  }
  sem.f05<-rbind(sem05,sem05.com)
  #Estimacion puntual
  est.sem05<-apply(sem.f05[201:le, ],2,mean)
  #est.sem05
}
if(le <= 1600)
{
  est.sem05<-apply(sem.f05[201:1600, ],2,mean)
  #est.sem05
}

e.sem05[i,]<-est.sem05
print(i)
}

#guardando estimadores
save(e.sem05,file="D:/DocumentosARIZAHFJ/TESIS DE DOCTORADO/e.sem05.RData")

#Calculando el sesgo
sesgo05<-apply(e.sem05,2,mean)-theta
sesgo05

#Calculando el ecm
V<-diag(var(e.sem05))
ecm05<-V+sesgo05^2
ecm05
```

### 7. Funciones para implementar el algoritmo IS

```
#Packages que utilizamos
library(msm) #normal truncada
library(mvtnorm) #normal multivariada

#-----
setwd("D:/DocumentosARIZAHFJ/TESIS DE DOCTORADO/fvMLG/Release")
dyn.load("fvMLG.dll") #Llama la DLL
is.loaded(symbol.For("fvmlg")) #Verifica si llamo la DLL correctamente
```

## Apéndice

---

```
is.loaded(symbol.For("recibe")) #Verifica si llamo la DLL correctamente

fvm.f<-function(par,serie)
{
  fi<-0.0
  lp<-length(par)
  n<-length(serie)
  .Fortran("fvmlg",p=as.integer(lp), par=as.double(par),n=as.integer(n),f=as.double(fi),serie=as.double(serie))$f
}

#Llamando la funcion de verosimilitud de Fortran
#ruta de la DLL..
setwd("D:/DocumentosARIZAHFJ/TESIS DE DOCTORADO/ISest/Release")
dyn.load("ISest.dll") #Llama la DLL
is.loaded(symbol.For("may")) #Verifica si llamo la DLL correctamente
is.loaded(symbol.For("day")) #Verifica si llamo la DLL correctamente
is.loaded(symbol.For("filter")) #Verifica si llamo la DLL correctamente
is.loaded(symbol.For("fadadoy")) #Verifica si llamo la DLL correctamente

#dyn.unload("ISest.dll")

#Llamando la función de verosimilitud de Fortran
#ruta de la DLL..
setwd("D:/DocumentosARIZAHFJ/TESIS DE DOCTORADO/IS2/Release")
dyn.load("IS2.dll") #Llama la DLL
is.loaded(symbol.For("dgay")) #Verifica si llamo la DLL correctamente
is.loaded(symbol.For("fydadoa")) #Verifica si llamo la DLL correctamente
is.loaded(symbol.For("ldgay")) #Verifica si llamo la DLL correctamente
is.loaded(symbol.For("lfydadoa")) #Verifica si llamo la DLL correctamente

#dyn.unload("IS2.dll")

filter.ao.f<-function(serie,par)
{
  n<-length(serie)
  a<-b<-rep(0,n)
  filtro<-.Fortran("filter", serie=as.double(serie),alfaf=as.double(a),
                 omegaf=as.double(b),par=as.double(par),n=as.integer(n))
  list(alfaf = filtro$alfaf, omegaf = filtro$omegaf)
}

#Funcion que genera una muestra de tamaño n=length(y) de g_ic(alfa|y)

rA.yf<-function(B,alfa.f,omega.f,par,Z)
{
  n<-length(alfa.f)
  A<-matrix(NA,B,n)
  mm<-rep(0.0,n)
  for(j in 1:B)
  {
    z<-Z[j,]
    A[j,]<- .Fortran("may", alfaf=as.double(alfa.f),
                  omegaf=as.double(omega.f), par=as.double(par),
                  z=as.double(z), ay=as.double(mm), n=as.integer(n))$ay
```

## Apéndice

---

```
}
A
}

#Función que evalúa a la funciones de importancia g_ic(alfa|y)
l.dAy.f<-function(x,alfa.f,omega.f, param)
{
n<-length(x)
mm<-rep(0,n)
.Fortran("day", x=as.double(x), alfaf=as.double(alfa.f),
        omegaf=as.double(omega.f), par = as.double(param), d=as.double(mm),
        df=as.double(0.0), n= as.integer(n))$df
}

dgAy.f<-function(x,alfa.f,omega.f, param)
{
n<-length(x)
mm<-rep(0,n)
.Fortran("dgay", x=as.double(x), alfaf=as.double(alfa.f),
        omegaf=as.double(omega.f), par = as.double(param), d=as.double(mm),
        df=as.double(0.0), n= as.integer(n))$df
}

ldgAy.f<-function(x,alfa.f,omega.f, param)
{
n<-length(x)
mm<-rep(0,n)
.Fortran("ldgay", x=as.double(x), alfaf=as.double(alfa.f),
        omegaf=as.double(omega.f), par = as.double(param), d=as.double(mm),
        df=as.double(0.0), n= as.integer(n))$df
}

#####

# Log-Verosimilitud de la distribución conjunta

#misma que la anterior, usando log y una parte en Fortran..

l.Lya.f<-function(serie,par,delta,alfa)
# serie: serie de datos observados
# par: parametros del SSM
# delta: vector indicador de censura(0:cen, 1:no-cen)
# alfa: una muestra de las variables de estados, aqui es p(alfa|serie)
{
n<-length(serie)
var<-matrix(0,n,n)
#alfa<-a.f[1,]
#par<-p
fy.a<-Fortran("fadadoy",serie=as.double(serie),par=as.double(par),
             delta=as.double(delta),alfa=as.double(alfa),V=as.double(var),
             fc=as.double(0.0),n=as.integer(n), pp=as.double(rep(0,n)))
#fy.a
```

## Apéndice

---

```
V<-matrix(fy.a$V,n,n)
fy.a$fc + dmvnorm(alfa,rep(0,n),V,log=T)
}

#dyn.unload("ISest.dll")

Lya.f<-function(serie,par,delta,alfa)
# serie: serie de datos observados
# par: parametros del SSM
# delta: vector indicador de censura(0:cen, 1:no-cen)
# alfa: una muestra de las variables de estados, aqui es p(alfa|serie)
{
n<-length(serie)
var<-matrix(0,n,n)
#alfa<-a.f[1,]
#par<-p
fy.a<-Fortran("fydadao",serie=as.double(serie),par=as.double(par),
delta=as.double(delta),alfa=as.double(alfa),V=as.double(var),
fc=as.double(0.0),n=as.integer(n), pp=as.double(rep(0,n)))
#fy.a
V<-matrix(fy.a$V,n,n)
dnm<- dmvnorm(alfa,rep(0,n),V)
Lya<-fy.a$fc * dnm
list(Lya=Lya,DN=dnm)
}

##la funciones que aproxima a la verosimilitud con la función de importancia

# Calculo de la verosimilitud via integración monte-carlo
# usando f(alfa|y) para obtener la muestra

Ly.mc.f22<-function(par,serie,delta,Z,B)
{
nume<-rep(0,length(serie))
filt<-filter.ao.f(serie,par)
alfaf<-filt$alfaf
omegaf<-filt$omegaf

Alfa<-rA.yf(B,alfaf,omegaf,par,Z)

for(j in 1:B)
{
L<-Lya.f(serie,par,delta,Alfa[j,])
nume[j]<-L$Lya/dgAy.f(Alfa[j,],alfaf,omegaf,par)
}
-log(mean(nume))
}

Ly.mc.f<-function(serie,par,delta,Z,B)
{
Lo<-rep(0,length(n))
filt<-filter.ao.f(serie,par)
```

## Apéndice

---

```
alfaf<-filt$alfaf
omegaf<-filt$omegaf

Alfa<-rA.yf(B,alfaf,omegaf,par,Z)

for(j in 1:B)
{
  Lo[j]<-1.Lya.f(serie,par,delta,Alfa[j,]) - 1.dAy.f(Alfa[j,],alfaf,omegaf,par)
}
-mean(Lo)
}

#dyn.unload("ISest.dll")

#####
#### Maximizando numéricamente la verosimilitud exacta,
#### utilizando integración monte carlo...
#####

#SIMULACIÓN
#-----
#Datos
#par=c(30,2,0.9,1) #parámetros poblacionales

#Llamando las series..
load(file="D:/DocumentosARIZAHFJ/TESIS DE DOCTORADO/TS.RData")
dim(TS)

#Límites de censura..
L.cen05<-25.66744
L.cen10<-26.61427
L.cen20<-27.77274
L.cen40<-29.33665

serie<-TS[1,]
M<-500 #Repeticiones Monte-Carlo.
n<-length(serie) #Longitud de la serie de tiempo
MNs<-rmvnorm(M,rep(0,n),diag(rep(1,n)))

load("D:/DocumentosARIZAHFJ/TESIS DE DOCTORADO/MNs.RData")
dim(MNs)

#####
#ESTIMACION CON 5,10,20 y 40% DE CENSURA EN LA SERIE
#-----

#inicio el proceso de estimación IS
theta=c(30,2,0.9,1)
p<-length(theta)
#espacio paramétrico
l.inf=c(-Inf,0,-.999999,0); l.sup=c(Inf,Inf,.99999,Inf)
```

```
e.is05<-matrix(0,500,p)

##--llamando los datos de la normal multivariada estándar
#load("D:/DocumentosARIZAHFJ/TESIS DE DOCTORADO/MNs.RData")
#MNs
M<-dim(MNs)[1]
n<-length(TS[1,][1:100]) #Longitud de la serie de tiempo

dim(MNs)

for(i in 1:500)
{
  #serie y stc
  serie<-TS[i, ][301:400]
  # L.cen051<-quantile(serie,0.05)
  secen05<-st.cen(serie,L.cen05,dir=FALSE) #Censurando la serie al 5%
  delta05<-pos(secen05,L.cen05)$delta

  #semillas
  #s<-arima(secen05,order=c(1,0,0))$coef
  #semi<-nlminb(c(as.numeric(s[2]),1,as.numeric(s[1]),1),fvm.f,serie=secen05,lower=1.inf,upper=1.sup)
  #semi
  #system.time(
  e.is05[i,]<-nlminb(e.is05[(i-1),],Ly.mc.f22,serie=secen05,delta=delta05,Z=MNs
,B=M,lower=1.inf,upper=1.sup)$par
  print(i)
#guardando estimadores
save(e.is05,file="D:/DocumentosARIZAHFJ/TESIS DE DOCTORADO/e.is05.RData")
}

#llamando los estimadores..
load(file="D:/DocumentosARIZAHFJ/TESIS DE DOCTORADO/e.is05.RData")

#Calculando el sesgo
sesgo05<-apply(eis05,2,mean)-theta
sesgo05

#Calculando el ecm
V<-diag(var(eis05))
ecm05<-V+sesgo05^2
ecm05
```

## 8. Funciones realizadas en Fortran

```
#Subrutina para la función de verosimilitud
! fvMLG.f90
!
! FUNCTIONS/SUBROUTINES exported from fvMLG.dll:
! fvMLG      - subroutine
!
```

## Apéndice

---

```
module datos

real*8, allocatable :: serie(:)

end module datos

subroutine recibe(n, y)

! Expose subroutine recibe to users of this DLL
!
!DEC$ ATTRIBUTES DLLEXPORT,C,REFERENCE,ALIAS:'recibe_' :: recibe

use datos
integer n
double precision y(n)
!
allocate(serie(n))
!
serie = y

end subroutine recibe

subroutine fvmlg(p,par,n, f,serie)

! Expose subroutine fvmlg to users of this DLL
!
!DEC$ ATTRIBUTES DLLEXPORT,C,REFERENCE,ALIAS:'fvmlg_'::fvmlg

! Variables
! use datos
integer p
integer i
integer n
real*8 omegaf(n), alphaf(n), omegap(n), alphap(n), arg(n), fp(n), &
      par(p), f, mu, sigma2, phi, tau2, serie(n)
!
! intrinsic LOG
!
! Body of fvMLG
mu = par(1)
sigma2 = par(2)
phi = par(3)
tau2 = par(4)
PI = 4.0*atan(1.0)

!condiciones iniciales
alphap(1)=0.0
omegap(1)=tau2/(1.0-phi**2)
arg(1)=(serie(1)-mu)**2/(omegap(1)+sigma2)
!
fp(1) = LOG(2*PI)/2 + LOG(omegap(1) + sigma2)/2 + arg(1)/2
```



## Apéndice

---

```
!
  do 100 i = 1, (n-1)
    omegaf(i) = sigma2*omegap(i)/(sigma2+omegap(i))
    omegap(i+1) = (phi**2)*omegaf(i)+tau2
    alphaf(i) = alphap(i)+(omegaf(i)/sigma2)*(serie(i)-alphap(i)-mu)
    alphap(i+1) = phi*alphaf(i)
    arg(i+1) = (serie(i+1)-alphap(i+1)-mu)**2/(omegap(i+1)+sigma2)
    fp(i+1) = LOG(2.0*PI)/2.0 + LOG(omegap(i+1)+sigma2)/2.0 + &
    0.5*arg(i+1)
    100 continue
  !

f = SUM(fp)
return
end subroutine fvmlg

#Subrutina para obtener muestras de la distribución predictiva.
! samfzy.f90
!
! FUNCTIONS/SUBROUTINES exported from samfzy.dll:
! samfzy      - subroutine
!

subroutine rtnormal(upper,media,desv,y)
! Variables
  USE IMSLFF90
  integer cd
  real*8 upper, media, desv, y, z, zt, u, e

  external RNNOF, RNUNF

  upper=(upper-media)/desv
  !NR=1

  IF(upper .GE. -0.45) THEN
    cd = 1
    DO WHILE (cd == 1)
      z = RNNOF()
      IF(z .LT. upper) cd=2
    END DO
    zt=z
  ELSE
    cd=1
    DO WHILE(cd == 1)
      z = -log(1-RNUNF())/(-upper)
      if(z .GE. -upper) then
        u = RNUNF()
        e = Exp(-(z**2 + (-upper)**2)/2 - upper*z)
        if(u .LT. e) cd = 2
      end if
    END DO
    zt=z*-1
  END IF
  !
  y=zt*desv+media
```

## Apéndice

---

```
return
end subroutine rtnormal

subroutine rtnormalr(upper,media,desv,y)
! Variables
USE IMSLFF90
integer cd
real*8 upper, media, desv, y, z, zt, u, e, a

external RNNOF, RNUNF

upper=(upper-media)/desv
!NR=1

IF(upper .GE. 0) THEN
  cd = 1
  DO WHILE (cd == 1)
    z = RNNOF()
  IF(z .LT. upper) cd=2
  END DO
  zt=z
ELSE
  cd=1
  DO WHILE(cd == 1)
a = -upper + Sqrt(upper**2+4)/2
z = -log(1-RNUNF())/a - upper
u = RNUNF()
e = Exp(-0.5*(z-a)**2)
if(u .LT. e) cd =2
  END DO
  zt=z*-1
END IF
!
y=zt*desv+media
return
end subroutine rtnormalr

subroutine fzy(serie,par,delta,n,z,y)

! Expose subroutine fzy to users of this DLL
!
!DEC$ ATTRIBUTES DLLEXPORT,C,REFERENCE,ALIAS:'fzy_':fzy

! Variables
USE IMSLFF90
integer j, n
real*8 mu, sigma2, phi, tau2, PI, serie(n), alpha(n), z(n), omep, delta(n), par(4),L, media,desv,y
external RNNOF, RNUNF, rtnormalr

  mu      = par(1)
sigma2 = par(2)
phi = par(3)
tau2 = par(4)
PI = 4.0*atan(1.0)
! Body of samfzy
```

## Apéndice

---

```
    omep = tau2**2/(1-phi**2)
    alpha(1) = RNNOF()*sqrt(omep)

!
    if(delta(1) .EQ. 0) then
        L=serie(1)
media=mu+alpha(1)
desv=sqrt(sigma2)
call rtnormalr(L,media,desv,y)
z(1) = y
        else
            z(1)=serie(1)
        end if
!
    do 100 j=2, n
        alpha(j) = phi*alpha(j-1) + sqrt(tau2)*RNNOF()
if(delta(j) .EQ. 0) then
        L=serie(j)
media=mu+alpha(j)
desv=sqrt(sigma2)
call rtnormalr(L,media,desv,y)
        z(j) = y
    else
        z(j) = serie(j)
    end if
    100 continue
    return
end subroutine fzy

#Subrutina para la función Q en el paso E del algoritmo EM
! fQ.f90
!
! FUNCTIONS/SUBROUTINES exported from fQ.dll:
! fQ      - subroutine
!
subroutine fvmlg(p,par,n, f,serie)

! Variables
! use datos
integer p
integer i
integer n
real*8 omegaf(n), alphaf(n), omegap(n), alphap(n), arg(n), fp(n), &
    par(p), f, mu, sigma2, phi, tau2, serie(n)
!
! intrinsic LOG
!
! Body of fvMLG
mu = par(1)
sigma2 = par(2)
phi = par(3)
tau2 = par(4)
PI = 4.0*atan(1.0)
```

## Apéndice

---

```
!condiciones iniciales
alphap(1)=0.0
omegap(1)=tau2/(1.0-phi**2)
arg(1)=(serie(1)-mu)**2/(omegap(1)+sigma2)
!
fp(1) = LOG(2*PI)/2 + LOG(omegap(1) + sigma2)/2 + arg(1)/2
!
  do 100 i = 1, (n-1)
omegaf(i) = sigma2*omegap(i)/(sigma2+omegap(i))
omegap(i+1) = (phi**2)*omegaf(i)+tau2
alphaf(i) = alphap(i)+(omegaf(i)/sigma2)*(serie(i)-alphap(i)-mu)
alphaf(i+1) = phi*alphaf(i)
arg(i+1) = (serie(i+1)-alphap(i+1)-mu)**2/(omegap(i+1)+sigma2)
fp(i+1) = LOG(2.0*PI)/2.0 + LOG(omegap(i+1)+sigma2)/2.0 + &
0.5*arg(i+1)
  100 continue
!

f = SUM(fp)
return
end subroutine fvmlg

subroutine fqlg(par,Z,n,m,fp)

! Expose subroutine fqlg to users of this DLL
!
!DEC$ ATTRIBUTES DLLEXPORT,C,REFERENCE,ALIAS:'fqlg_'::fqlg

! Variables
integer p, n, m
real*8 par(4), f, Z(m,n), fa,fp, serie(n)
external fvmlg

p=4
f=0.0
fa=0.0
! Body of fqlg
Do 100 i=1,m
  serie=Z(i,1:n)
  call fvmlg(p,par,n,f,serie)
  fa=fa+f
100 continue
fp=fa/m
return
end subroutine fqlg

#Subrutina para implementar el algoritmo IS

! ISest.f90
!
! FUNCTIONS/SUBROUTINES exported from ISest.dll:
! ISest      - subroutine
!
subroutine may(alfaf,omegaf,par,z,ay,n)
```

## Apéndice

---

```
! Expose subroutine may to users of this DLL
!
!DEC$ ATTRIBUTES DLLEXPORT,C,REFERENCE,ALIAS:'may_':may

! Variables
integer n, j
double precision alfaf(n), omegaf(n), par(4), z(n), ay(n)
real*8 mu, sigma2, phi, tau2, PI, mue, vare

! Body of mA.y

mu      = par(1)
sigma2 = par(2)
phi     = par(3)
tau2    = par(4)
PI      = 4.0*atan(1.0)
!
ay(n) = alfaf(n)+sqrt(omegaf(n))*z(n)
do 100 j = (n-1), 1, -1
    mue = (tau2*alfaf(j)+phi*omegaf(j)*ay(j+1))/(tau2+phi**2*omegaf(j))
    vare = tau2*omegaf(j)/(tau2+phi**2*omegaf(j))
    ay(j) = mue+sqrt(vare)*z(j)
100 continue

return
end subroutine may

subroutine day(x,alfaf,omegaf,par,d, df, n)

! Expose subroutine day to users of this DLL
!
!DEC$ ATTRIBUTES DLLEXPORT,C,REFERENCE,ALIAS:'day_':day

! Variables
integer n, j
double precision alfaf(n), omegaf(n), par(4), x(n), d(n)
real*8 mu, sigma2, phi, tau2, PI, mue, vare, df

!body of day
mu      = par(1)
sigma2 = par(2)
phi     = par(3)
tau2    = par(4)
PI      = 4* atan(1.0)

d(n) = -0.5*LOG(2*PI) - 0.5*LOG(omegaf(n)) - (x(n)-alfaf(n))**2/(2*omegaf(n))

do 100 j = (n-1), 1, -1
mue = (tau2*alfaf(j) + phi*omegaf(j)*x(j+1))/(tau2+phi**2*omegaf(j))
vare = tau2*omegaf(j)/(tau2 + phi**2*omegaf(j))
d(j) = -0.5*LOG(2*PI) - 0.5*LOG(vare) - (x(j)-mue)**2/(2*vare)
100 continue
df = sum(d)
```

## Apéndice

---

```
        return

end subroutine day

subroutine filter(serie,alfaf,omegaf,par,n)

! Expose subroutine filter to users of this DLL
!
!DEC$ ATTRIBUTES DLLEXPORT,C,REFERENCE,ALIAS:'filter_':filter

! Variables

integer t, n
double precision alfaf(n), omegef(n), serie(n), par(4)
real*8 mu, sigma2, phi, tau2, omegef(n), alfap(n)

!body of filter
mu = par(1)
sigma2 = par(2)
phi = par(3)
tau2 = par(4)

!Recursiones de Kalman de un paso (Generalized SSM)

!condiciones iniciales
alfap(1) = 0
omegef(1) = tau2/(1-phi**2)
do 100 t = 1, (n-1)
omegef(t) = sigma2*omegef(t)/(sigma2+omegef(t))
omegef(t+1) = phi**2*omegef(t)+tau2
alfap(t) = alfaf(t)+(omegef(t)/sigma2)*(serie(t)-alfap(t)-mu)
alfap(t+1) = phi*alfap(t)
100 continue

omegef(n) = sigma2*omegef(n)/(sigma2+omegef(n))
alfap(n) = alfaf(n)+(omegef(n)/sigma2)*(serie(n)-alfap(n)-mu)

return

end subroutine filter

subroutine fadadoy(serie,par,delta,alfa,V,fc,n,pp)

! Expose subroutine fadadoy to users of this DLL
!
!DEC$ ATTRIBUTES DLLEXPORT,C,REFERENCE,ALIAS:'fadadoy_':fadadoy

! Variables

USE IMSL90

integer i, j, n
double precision serie(n), par(4), delta(n), alfa(n), V(n,n), fc, pp(n)
```

## Apéndice

---

```
real*8 dya(n), PI, mu, sigma2, phi, tau2, x(n), dn(n)

external DNORDF

!Body of fadadoy

    mu      = par(1)
sigma2 = par(2)
phi      = par(3)
tau2     = par(4)
PI = 4*atan(1.0)
!
do i= 1, n
    do j=1, n
        V(i,j) = tau2*phi**abs(i-j)/(1-phi**2)
    end do
    x(i) = (serie(i)-mu-alfa(i))/sqrt(sigma2)
    dn(i) = - 0.5*log(2*PI) - 0.5*log(sigma2) - 0.5*x(i)**2
pp(i) = DNORDF(x(i))
    dya(i) = delta(i)*dn(i) + (1-delta(i))*log(pp(i))
end do

fc=sum(dya)
return

end subroutine fadadoy

subroutine lya(serie,par,delta,alfa,V,fc,n,pp)

! Expose subroutine lya to users of this DLL
!
!DEC$ ATTRIBUTES DLLEXPORT,C,REFERENCE,ALIAS:'lya_':::lya

! Variables

USE IMSLF90

integer i, j, n
double precision serie(n), par(4), delta(n), alfa(n), V(n,n), fc, pp(n)
real*8 dya(n), PI, mu, sigma2, phi, tau2, x(n), dn(n)
parameter

external DNORDF

!Body of fadadoy

    mu      = par(1)
sigma2 = par(2)
phi      = par(3)
tau2     = par(4)
PI = 4*atan(1.0)
!
do i= 1, n
    do j=1, n
```

```
V(i,j) = tau2*phi**abs(i-j)/(1-phi**2)
end do
x(i) = (serie(i)-mu-alfa(i))/sqrt(sigma2)
dn(i) = - 0.5*log(2*PI) - 0.5*log(sigma2) - 0.5*x(i)**2
pp(i) = DNORDF(x(i))
dya(i) = delta(i)*dn(i) + (1-delta(i))*log(pp(i))
end do
call

fc=sum(dya)
return

end subroutine lya
```