

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE CIENCIAS

**REGRESIÓN AUTOMÁTICA DIFUSA APLICADA A DATOS
ECONÓMICOS**

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO
MATEMÁTICO**

RODRIGO PAÚL CAJAMARCA CHAUCA

rpaulcajamarca@hotmail.com

DIRECTOR: DR. HERMANN MENA

hermann.mena@epn.edu.ec

QUITO, OCTUBRE 2011

DECLARACIÓN

Yo, RODRIGO PAÚL CAJAMARCA CHAUCA, declaro bajo juramento que el trabajo aquí escrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su reglamento y por la normatividad institucional vigente.

RODRIGO PAÚL CAJAMARCA CHAUCA

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por RODRIGO PAÚL CAJAMAR-
CA CHAUCA bajo mi supervisión.

Dr. Hermann Mena
DIRECTOR DE PROYECTO

AGRADECIMIENTOS

A mi familia, porque en ella siempre encuentro el apoyo emocional y la motivación para superarme cada día.

A Hermann Mena, por haberme brindado su amistad y apoyo para la consecución del presente trabajo.

Gracias sinceras, a todas aquellas personas que confiaron en mí y que me apoyaron a lo largo de la carrera.

Rodrigo Paúl

DEDICATORIA

A mis padres, José Gonzalo, Blanca Estela y a mi hermano Juan Carlos.

Rodrigo Paúl

RESUMEN

Por lo general, los procesos económicos son modelados mediante técnicas estadísticas. Si bien esta metodología se utiliza con frecuencia, muchas de estas técnicas no dan buenos resultados, debido principalmente a la presencia de incertidumbre en los datos o falta de información del fenómeno de estudio. De la necesidad de desarrollar nuevas técnicas o métodos alternativos que permitan tratar este tipo de problemas surge la modelación difusa.

Este trabajo se encuadra en un contexto donde la Estadística e Inteligencia Computacional van de la mano: la Regresión Difusa. La idea fundamental de esta regresión es generalizar conceptos de regresión tradicional a conjuntos difusos. Concretamente, se investigará el potencial de aplicar los métodos automáticos de regresión difusa a cierto tipo de series económicas. En particular, se estudian los siguientes métodos: mínimos cuadrados por lotes, mínimos cuadrados recursivo, aprendizaje desde el ejemplo modificado y agrupamiento difuso combinado. Adicionalmente, se propone el método de mínimos cuadrados recursivo combinado el cual es una de las principales contribuciones de este trabajo. Cada uno de estos métodos han sido descritos e implementados en R para el caso unidimensional y se generalizan para el caso de entradas y salidas múltiples. Finalmente, se muestran resultados numéricos del PIB no petrolero, la Reserva Internacional de Libre Disponibilidad, un índice de liquidez financiera y los índices de precios al consumidor y productor, en las cuales se visualiza el comportamiento y desempeño de los métodos comparándolos con modelos SARIMA.

ABSTRACT

In general, several applications in economics are modeled using statistics techniques. Although, this methodology is pretty standard the results are not that good mainly because the uncertainty of the model is actually not taken into account. Fuzzy techniques are able to handle this problem.

This work states where Statistics and Computational Intelligence converge: Fuzzy Regression. The main idea of this technique is generalize concepts of statistic regression to fuzzy sets. Here, we investigate the potential of using automated fuzzy methods for modeling some economic data. Particularly, we discussed: Batch Least Squares, Recursive Least Squares, Learning from Example and Combined Clustering. In addition, we proposed a novel method Combined Recursive Least Squares which represents our main theoretical contribution. All the methods are described and implemented for the one dimensional case and generalized for the multiple entrance-multiple output case. Finally, numerical tests for: the non oil PIB, RILD, and some financial index show the good performance of the methods compared with SARIMA models.

Índice general

Índice general	8
Índice de figuras	10
Capítulo 1: INTRODUCCIÓN	14
Capítulo 2: SISTEMAS DIFUSOS	17
2.1. Conjuntos difusos	18
2.1.1. Relaciones básicas	21
2.1.2. Operaciones	22
2.1.3. Propiedades	22
2.2. Funciones de pertenencia	24
2.2.1. Tipos de funciones de pertenencia	24
2.2.2. Características de las funciones de pertenencia	30
2.2.3. Clasificación de los conjuntos difusos	31
2.3. Sistemas de inferencia difusos	33
2.3.1. Estructura de un sistema de inferencia difuso	33
2.3.2. Tipos de sistemas de inferencia difuso	36
Capítulo 3: MÉTODOS AUTOMÁTICOS DE REGRESIÓN DIFUSA	41
3.1. Principales definiciones	42
3.2. Mínimos cuadrados por lotes (MCL)	44
3.2.1. Mínimos cuadrados ordinarios	44
3.2.2. Mínimos cuadrados por lotes	47

3.3. Mínimos cuadrados recursivo (MCR)	50
3.4. Aprendizaje desde el ejemplo modificado (AEM)	56
Capítulo 4: MÉTODOS AUTOMÁTICOS DE REGRESIÓN DIFUSA COMBINADOS	
DOS	60
4.1. Agrupamiento difuso combinado (ADC)	60
4.2. Mínimos cuadrados recursivo combinado (MCRC)	66
4.3. Generalización al caso de entrada-salida múltiple	70
4.3.1. Mínimos cuadrados por lotes entrada-salida múltiple	70
4.3.2. Mínimos cuadrados recursivo entrada-salida múltiple	70
4.3.3. Aprendizaje desde el ejemplo modificado entrada-salida múltiple	73
4.3.4. Agrupamiento difuso combinado entrada-salida múltiple	73
4.3.5. Mínimos cuadrados recursivo combinado entrada-salida múltiple	73
4.4. Aplicación a modelos económicos	77
4.4.1. Producto Interno Bruto (PIB) no Petrolero de Ecuador	79
4.4.2. Reserva Internacional de Libre Disponibilidad (RILD) de Ecuador	83
4.4.3. Índice de Liquidez de Instituciones Financieras	90
4.4.4. Índices de Precios al Consumidor y al Productor de Ecuador .	104
Capítulo 5: CONCLUSIONES Y RECOMENDACIONES	114
Apéndice	116
A. Librería de funciones	117
Bibliografía	128

Índice de figuras

2.1. Límites para los conjuntos	18
2.2. Función de pertenencia delta $\bar{x} = 2$	25
2.3. Función de pertenencia gaussiana $\bar{x} = 3$ y $\sigma = 1$	25
2.4. Función de pertenencia campana generalizada $a = 1, c = 2,5$ y $b = 2$.	26
2.5. Función de pertenencia sigmoideal $a = 3$ y $c = 1,5$	27
2.6. Función de pertenencia triangular $a = 1, b = 2$ y $c = 4$	28
2.7. Función de pertenencia triangular simétrica $\bar{x} = 2,5$ y $\sigma = 1,5$	28
2.8. Función de pertenencia trapezoidal $a = 0,5, b = 1,5, c = 2,5$ y $d = 4$.	29
2.9. Función de pertenencia gamma $a = 2$ y $k = 2$	30
2.10. Sistema de inferencia difuso	34
3.1. Esquema de funcionamiento de mínimos cuadrados recursivo	50
4.1. PIB mínimos cuadrados por lotes (MCL)	79
4.2. PIB mínimos cuadrados recursivo 100 iteraciones (MCR)	80
4.3. PIB aprendizaje desde el ejemplo modificado (AEM)	80
4.4. PIB agrupamiento difuso combinado (ADC)	81
4.5. PIB mínimos cuadrados recursivo combinado 100 iteraciones (MCRC)	81
4.6. PIB residuos de los métodos automáticos	82
4.7. RILD mínimos cuadrados por lotes (MCL)	85
4.8. RILD mínimos cuadrados recursivo 100 iteraciones (MCR)	85
4.9. RILD mínimos cuadrados recursivo 500 iteraciones (MCR)	86
4.10. RILD agrupamiento difuso combinado (ADC)	86

4.11. RILD mínimos cuadrados recursivo combinado 100 iteraciones (MCRC)	
Gaussiana	87
4.12. RILD mínimos cuadrados recursivo combinado 500 iteraciones (MCRC)	
Gaussiana	87
4.13. RILD residuos de los métodos automáticos y <i>back testing</i>	88
4.14. RILD contraste	89
4.15. Banco Pichincha mínimos cuadrados recursivo (MCR)	91
4.16. Banco Pichincha agrupamiento difuso combinado (ADC)	92
4.17. Banco Pichincha mínimos cuadrados recursivo combinado 100 itera- ciones (MCRC) Gaussiana	92
4.18. Banco Pichincha mínimos cuadrados recursivo combinado 500 itera- ciones (MCRC) Gaussiana	93
4.19. Banco Pichincha residuos de los métodos automáticos y <i>back testing</i>	93
4.20. Banco Pichincha contraste	94
4.21. Banco Internacional mínimos cuadrados recursivo (MCR)	96
4.22. Banco Internacional agrupamiento difuso combinado (ADC)	96
4.23. Banco Internacional mínimos cuadrados recursivo combinado 100 iteraciones (MCRC) Gaussiana	97
4.24. Banco Internacional mínimos cuadrados recursivo combinado 500 iteraciones (MCRC) Gaussiana	97
4.25. Banco Internacional residuos de los métodos automáticos y <i>back tes- ting</i>	98
4.26. Banco Internacional contraste	99
4.27. Banco Amazonas mínimos cuadrados recursivo (MCR)	100
4.28. Banco Amazonas agrupamiento difuso combinado (ADC)	100

4.29. Banco Amazonas mínimos cuadrados recursivo combinado 100 iteraciones (MCRC) Gaussiana	101
4.30. Banco Amazonas mínimos cuadrados recursivo combinado 500 iteraciones (MCRC) Gaussiana	101
4.31. Banco Amazonas residuos de los métodos automáticos y <i>back testing</i>	102
4.32. Banco Amazonas contraste	102
4.33. IPC mínimos cuadrados recursivo (MCR)	105
4.34. IPC agrupamiento difuso combinado (ADC)	106
4.35. IPC mínimos cuadrados recursivo combinado 100 iteraciones (MCRC) Gaussiana	106
4.36. IPC mínimos cuadrados recursivo combinado 500 iteraciones (MCRC) Gaussiana	107
4.37. IPC residuos de los métodos automáticos y <i>back testing</i>	107
4.38. IPC contraste	108
4.39. IPP mínimos cuadrados recursivo (MCR)	110
4.40. IPP agrupamiento difuso combinado (ADC)	110
4.41. IPP mínimos cuadrados recursivo combinado 100 iteraciones (MCRC) Gaussiana	111
4.42. IPP mínimos cuadrados recursivo combinado 500 iteraciones (MCRC) Gaussiana	111
4.43. IPP residuos de los métodos automáticos y <i>back testing</i>	112
4.44. IPP contraste	113

Lista de algoritmos

1.	Mínimos cuadrados por lotes	49
2.	Mínimos cuadrados recursivo	54
3.	Mínimos cuadrados recursivo modificado	55
4.	Aprendizaje desde el ejemplo modificado	59
5.	Agrupamiento difuso combinado	65
6.	Mínimos cuadrados recursivo combinado	69
7.	Mínimos cuadrados por lotes entrada-salida múltiple	71
8.	Mínimos cuadrados recursivo entrada-salida múltiple	72
9.	Aprendizaje desde el ejemplo modificado entrada-salida múltiple . . .	74
10.	Agrupamiento difuso combinado entrada-salida múltiple	75
11.	Mínimos cuadrados recursivo combinado entrada-salida múltiple . . .	76

CAPÍTULO 1

INTRODUCCIÓN

Actualmente existen muchos métodos y técnicas diseñadas para modelar una gran variedad de fenómenos económicos. Lamentablemente, muchos de estos métodos no han dado buenos resultados debido principalmente a la presencia de incertidumbre en los datos; este es el caso de varias técnicas tradicionales, e.g., regresión clásica, series de tiempo, etc. En la mayoría de casos la falta de conocimiento del sistema, falta de confianza en la información, o la presencia de datos incompletos/imputados constituyen los obstáculos más grandes que los métodos de modelización deben enfrentar. Afortunadamente, en situaciones como esta la modelación difusa constituye una opción real y competitiva, ver [1].

Este trabajo se encuadra en un contexto donde la Estadística e Inteligencia Computacional van de la mano: la Regresión Difusa. La idea fundamental de esta regresión es generalizar conceptos de técnicas tradicionales de regresión a datos que se pueden modelar con conjuntos difusos. Estos conjuntos, en general, describen de mejor manera los fenómenos inmersos en el sistema. La regresión probabilística solamente puede modelar el fenómeno de la incertidumbre mediante la inclusión de intervalos de confianza. La aleatoriedad puede considerarse como uno de los componentes de la vaguedad, junto a la imprecisión. Sin embargo, otras formas de incertidumbre se pueden encontrar en la ambigüedad, la incongruencia, problemas de especificación, factores no considerados por la alta complejidad del sistema, etc. En conclusión existen diversas fuentes de incertidumbre, entre las cuales la aleatoriedad es sólo una de ellas. En algunos casos es posible incorporar

estas restricciones a ciertos modelos de regresión probabilística. Lamentablemente, la inclusión de estas restricciones implica que algunos supuestos requeridos por la teoría ya no se satisfagan, como por ejemplo: el valor esperado de los residuos es cero, la varianza de los residuos es una constante σ^2 , etc. Esto no ocurre con la regresión difusa, donde la presencia de un número mayor de restricciones no constituye un impedimento desde el punto de vista teórico para su aplicación, puesto que las desviaciones entre los valores de pertenencia observados y los valores de pertenencia estimados se asumen dependientes de la incertidumbre.

Es por esta razón, que es necesario una herramienta que permita manejar la ambigüedad, para estos casos la lógica difusa ha demostrado ser muy apropiada, ver [2]. En la actualidad, los sistemas difusos están siendo usados en campos como la industria y la ciencia, con aplicaciones en áreas como el control difuso, análisis de datos e inteligencia artificial, ver [3]. Sin embargo, la implementación de un sistema difuso puede consumir mucho tiempo, debido a que no existe una herramienta o instrumento que permita determinar los parámetros del sistema difuso, por lo tanto es necesario contar con un paquete informático (software) que permita construir automáticamente sistemas difusos a partir de la información disponible.

Este proyecto está orientado a formalizar, generalizar y extender las propuestas que se han formulado hasta la fecha sobre la regresión difusa, con el objetivo de proporcionar un instrumento metodológico a problemas reales en que los datos disponibles se ven afectados por factores como la imprecisión e incertidumbre. Para lo cual, nos enfocamos en los métodos automáticos de regresión difusa, los cuales constituyen una herramienta general para tratar series de datos provenientes de problemas de diferente naturaleza.

En particular se revisará los siguientes métodos:

- Mínimos cuadrados por lotes,

- Mínimos cuadrados recursivo,
- Aprendizaje desde el ejemplo modificado,
- Agrupamiento difuso combinado.

Adicionalmente, se propone e implementa el método de mínimos cuadrados recursivo combinado el cual es una de las principales contribuciones de este trabajo. En este método, se explotan las características más importantes de los métodos que se encuentran en la literatura, y además ofrece una mayor flexibilidad de diseño para la optimización de sistemas difusos.

Cada uno de estos métodos han sido descritos e implementados para el caso unidimensional, sin embargo, su generalización para el caso de entradas y salidas múltiple se sigue directamente.

Este trabajo está organizado de la siguiente manera: en el Capítulo 2, se muestra una breve revisión a la teoría de conjuntos difusos, funciones de pertenencia, y sistemas de inferencia difusos, conceptos sobre los que se sustenta este trabajo. En el Capítulo 3, se presentan detalles técnicos de los métodos mínimos cuadrados por lotes, mínimos cuadrados recursivo y aprendizaje desde el ejemplo modificado. Estos métodos se pueden encontrar en la literatura. En el Capítulo 4, se propone el método de mínimos cuadrados recursivo combinado, se generaliza cada uno de los métodos al caso de entrada y salida múltiple, estos son las principales contribuciones de este trabajo. Además, se describe el método de agrupamiento difuso combinado y se muestran resultados numéricos del PIB no petrolero, la Reserva Internacional de Libre Disponibilidad, un índice de liquidez financiera y los índices de precios al consumidor y productor, en las cuales se visualiza el comportamiento y desempeño de los métodos. Finalmente, en el Capítulo 5 se presentan conclusiones, así como las posibles líneas de investigación futuras.

CAPÍTULO 2

SISTEMAS DIFUSOS

La representación del conocimiento tradicional se basa en la lógica bivalente o booleana¹, no obstante, el pensamiento humano y su comportamiento están estrictamente conectados con la imprecisión e incertidumbre. El problema de capturar la incertidumbre en problemas reales es muy difícil de superar utilizando únicamente la teoría de conjuntos clásica. La álgebra booleana tradicional, caracterizada por valores categóricos de verdad y falsedad, no es capaz de hacer frente a estos problemas, para solucionar este inconveniente se han propuesto varias metodologías que tratan de incluir dentro de la modelación diferentes aspectos de incertidumbre. Entre los métodos más populares, se destaca la lógica difusa, que constituye una extensión de la lógica tradicional para valores de pertenencia difusos. En este capítulo se revisa brevemente la teoría de conjuntos difusos y se introducen algunas definiciones que serán utilizadas en los próximos capítulos. Esto servirá para demostrar el alcance de esta metodología para su aplicación en problemas del mundo real.

¹Es una estructura que rigoriza las operaciones lógicas Y, O y NO, así como el conjunto de operaciones unión, intersección y complemento. Se denomina así en honor a George Boole, (2 de noviembre de 1815 a 8 de diciembre de 1864), matemático inglés que fue el primero en definirla como parte de un sistema lógico a mediados del siglo XIX. El álgebra de Boole fue un intento de utilizar las técnicas algebraicas para tratar expresiones de la lógica proposicional. En la actualidad, el álgebra de Boole se aplica de forma generalizada en el ámbito del diseño electrónico.

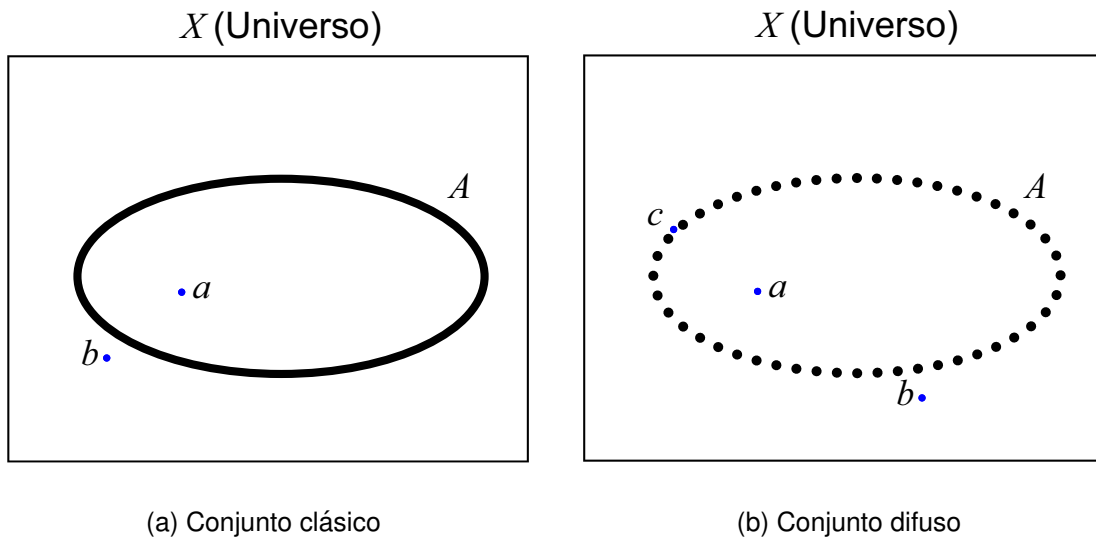


Figura 2.1: Límites para los conjuntos

2.1. Conjuntos difusos

La teoría de los conjuntos difusos fue propuesta por Lotfi Zadeh [4] para describir matemáticamente la imprecisión y la vaguedad que se presenta en el pensamiento y lenguaje diario. En los conjuntos clásicos, la transición de un elemento entre la pertenencia y no pertenencia es incuestionable debido a que los límites de estos conjuntos están muy bien definidos. La Figura 2.1a permite comprender esta idea desde una perspectiva bidimensional. La línea continua representa los límites del conjunto A ; es claro que el punto a es elemento del conjunto A y el punto b no pertenece al conjunto A .

Para un elemento en un universo que contiene conjuntos difusos, esta transición es gradual, ya que existen varios niveles de pertenencia debido al hecho de que los límites de los conjuntos difusos son vagos o ambiguos. La Figura 2.1b muestra la ambigüedad de los límites en un conjunto difuso \tilde{A} en el universo X . La región punteada representa los límites de \tilde{A} , en la región central del conjunto difuso, el punto a pertenece completamente al conjunto. Fuera de los límites del conjunto difuso, el

punto b no pertenece. Sin embargo, la pertenencia del punto c es ambigua, ya que este se encuentra en la región punteada. Si la *no pertenencia* de un elemento a un conjunto se representa por el número 0 y la *pertenencia* completa se representa por 1, entonces el punto c de la Figura 2.1b debe tener algún valor intermedio de pertenencia en el intervalo $[0, 1]$. Es decir, la pertenencia de un punto en \tilde{A} se aproxima al valor de 1 a medida que el punto se acerca a la región central del conjunto y el valor de pertenencia se acerca a 0 a medida que el punto se aleja de los límites de \tilde{A} . Por lo tanto, la pertenencia de un elemento del universo se mide por la función que describe la vaguedad o ambigüedad. La visión de la teoría clásica de conjuntos hace énfasis en que los límites de los conjuntos clásicos deben estar bien definidos, de esta manera la pertenencia al conjunto está definida claramente. Es decir, un elemento puede pertenecer o no a un conjunto. Sin embargo, muchos conjuntos no pueden ser caracterizados de esta manera, para vencer las limitaciones de los conjuntos clásicos, se introdujo los conceptos de los conjuntos difusos.

Sea X el conjunto universo. Un conjunto difuso \tilde{A} en X está caracterizado por una *función de pertenencia* $\mu_{\tilde{A}}$ que toma valores en el intervalo $[0, 1]$, ver [4]. Por lo tanto, un conjunto difuso es una generalización de un conjunto clásico pues la función de pertenencia permite que tome valores en el intervalo $[0, 1]$ en lugar de solo $\{0, 1\}$ (no pertenece ó pertenece respectivamente).

Un conjunto difuso \tilde{A} en el universo X está representado como un conjunto de pares,

$$\tilde{A} = \{(x, \mu_{\tilde{A}}(x)) \mid x \in X\}$$

donde $\mu_{\tilde{A}} : X \rightarrow [0, 1]$ es la función de pertenencia para \tilde{A} , que asigna a cada elemento $x \in X$ un grado de pertenencia en el intervalo $\mu_{\tilde{A}} \in [0, 1]$ para el conjunto \tilde{A} y dado que X es el conjunto universo podemos decir que $\tilde{A} \subseteq X$, para más deta-

les ver [5].

Cuando el universo X es discreto y finito con cardinalidad n , esto es $X = \{x_1, \dots, x_n\}$, un conjunto difuso \tilde{A} puede ser representado como

$$\tilde{A} = \left\{ \frac{\mu_{\tilde{A}}(x_1)}{x_1} + \dots + \frac{\mu_{\tilde{A}}(x_n)}{x_n} \right\} = \left\{ \sum_{i=1}^n \frac{\mu_{\tilde{A}}(x_i)}{x_i} \right\}$$

Y si el universo X es un intervalo de número reales, el conjunto difuso \tilde{A} puede ser expresado como

$$\tilde{A} = \left\{ \int_X \frac{\mu_{\tilde{A}}(x)}{x} \right\}$$

Los símbolos \sum , $+$, \int , en las ecuaciones anteriores, hacen referencia a la unión de los elementos del conjunto, mas no, las operaciones matemáticas, para una descripción detallada ver [6]. Igualmente, el operador divisor se usa para conectar un elemento y su valor de pertenencia. Esta notación es muy útil porque se la puede extender al caso multidimensional. Para todo $j = 1, \dots, M$

$$\tilde{A}_j = \left\{ \int_X \frac{\mu_{\tilde{A}_j}(x)}{x} \right\}$$

Un conjunto difuso vacío es aquel que tiene valores de pertenencia igual a cero para todos los elementos del conjunto, es decir $\tilde{A} = \emptyset$ si $\mu_{\tilde{A}}(x) = 0$ para todo $x \in X$, para más detalles ver [7].

Observación 1. La colección de todos los conjuntos difusos en el universo X se nota como el *conjunto difuso potencia* $P(X)$. Y puesto que todos los conjuntos difusos se pueden solapar, la cardinalidad $|P(X)|$ es infinita, ver [8] para más detalles.

2.1.1. Relaciones básicas

Sean \tilde{A} y \tilde{B} dos conjuntos difusos en el universo X .

$$\tilde{A} = \{(x, \mu_{\tilde{A}}(x)) \mid x \in X\}, \mu_{\tilde{A}}(x) \in [0, 1]$$

$$\tilde{B} = \{(x, \mu_{\tilde{B}}(x)) \mid x \in X\}, \mu_{\tilde{B}}(x) \in [0, 1]$$

Subconjunto $\tilde{A} \subseteq \tilde{B}$ si y solamente si $\mu_{\tilde{A}}(x) \leq \mu_{\tilde{B}}(x)$ para todo $x \in X$.

Igualdad Dos conjuntos difusos, \tilde{A} y \tilde{B} , son iguales si sus valores de pertenencia son iguales, es decir $\tilde{A} = \tilde{B}$ si y solamente si $\mu_{\tilde{A}}(x) = \mu_{\tilde{B}}(x)$ para todo $x \in X$.

Pertenencia de un elemento (“ \in ”) Normalmente este símbolo no se utiliza en los conjuntos difusos, debido a que la función de pertenencia es la encargada de asignar el grado de pertenencia de cada elemento. Sin embargo, “ \in ” se puede usar para el universo X , dado que el grado de pertenencia de cada elemento en X es 1 como en un conjunto ordinario.

Cualquier conjunto difuso \tilde{A} definido en un universo X es un subconjunto del universo, es decir, $\tilde{A} \subseteq X \Rightarrow \mu_{\tilde{A}}(x) \leq \mu_X(x)$, una descripción más detallada se puede obtener en [9].

Observación 2. En los conjuntos ordinarios, si un elemento x pertenece a un conjunto A entonces $\mu_A(x) = 1$, de lo contrario $\mu_A(x) = 0$, en términos de los conjuntos difusos. Si para los conjuntos ordinarios A y B tenemos $A \subseteq B$, entonces cada elemento de A se encuentra en B , en cuyo caso $\mu_A(x) = \mu_B(x) = 1$ y para los elementos de B que no están en A , tenemos $\mu_A(x) = 0 < \mu_B(x) = 1$. La igualdad puede ser expresada en términos de los subconjuntos como $A = B$ si y solamente si $A \subseteq B$ y $B \subseteq A$.

2.1.2. Operaciones

Sean \tilde{A} y \tilde{B} dos conjuntos difusos en el universo X .

Unión La unión de los difusos \tilde{A} y \tilde{B} es el conjunto difuso definido por la función de pertenencia

$$\mu_{\tilde{A} \cup \tilde{B}}(x) = \mu_{\tilde{A}}(x) \vee \mu_{\tilde{B}}(x)$$

Intersección La intersección de los difusos \tilde{A} y \tilde{B} es el conjunto difuso definido por la función de pertenencia

$$\mu_{\tilde{A} \cap \tilde{B}}(x) = \mu_{\tilde{A}}(x) \wedge \mu_{\tilde{B}}(x)$$

Complemento El complemento del conjunto difuso \tilde{A} es el conjunto difuso definido por la función de pertenencia

$$\mu_{\tilde{\tilde{A}}}(x) = 1 - \mu_{\tilde{A}}(x)$$

Para más información ver [10].

2.1.3. Propiedades

Las propiedades de los conjuntos clásicos se pueden adaptar para los conjuntos difusos, estas propiedades se detallan a continuación:

Conmutativa

$$\begin{aligned}\tilde{A} \cup \tilde{B} &= \tilde{B} \cup \tilde{A} \\ \tilde{A} \cap \tilde{B} &= \tilde{B} \cap \tilde{A}\end{aligned}$$

Asociativa

$$\begin{aligned}\tilde{A} \cup (\tilde{B} \cup \tilde{C}) &= (\tilde{A} \cup \tilde{B}) \cup \tilde{C} \\ \tilde{A} \cap (\tilde{B} \cap \tilde{C}) &= (\tilde{A} \cap \tilde{B}) \cap \tilde{C}\end{aligned}$$

Distributiva

$$\begin{aligned}\tilde{A} \cup (\tilde{B} \cap \tilde{C}) &= (\tilde{A} \cup \tilde{B}) \cap (\tilde{A} \cup \tilde{C}) \\ \tilde{A} \cap (\tilde{B} \cup \tilde{C}) &= (\tilde{A} \cap \tilde{B}) \cup (\tilde{A} \cap \tilde{C})\end{aligned}$$

Transitiva

Si $\tilde{A} \subseteq \tilde{B}$ y $\tilde{B} \subseteq \tilde{C}$, entonces $\tilde{A} \subseteq \tilde{C}$

Idempotencia

$$\begin{aligned}\tilde{A} \cup \tilde{A} &= \tilde{A} \\ \tilde{A} \cap \tilde{A} &= \tilde{A}\end{aligned}$$

Identidad

$$\begin{aligned}\tilde{A} \cup \emptyset &= \tilde{A} \text{ y } \tilde{A} \cap X = \tilde{A} \\ \tilde{A} \cap \emptyset &= \emptyset \text{ y } \tilde{A} \cup X = X\end{aligned}$$

Involutiva

$$\overline{\tilde{A}} = \tilde{A}$$

Los *principios de De Morgan* formulados para los conjuntos clásicos también se pueden aplicar a los conjuntos difusos.

$$\begin{aligned}\overline{\tilde{A} \cup \tilde{B}} &= \tilde{A} \cap \tilde{B} \\ \overline{\tilde{A} \cap \tilde{B}} &= \tilde{A} \cup \tilde{B}\end{aligned}$$

Casi todas las operaciones de los conjuntos clásicos se pueden aplicar a los conjuntos difusos, excepto el axioma medio de exclusión² y el de contradicción³, debido a que los conjuntos difusos se pueden solapar, por tanto un conjunto difuso y su complemento se pueden superponer, para más detalles ver [8].

²Axioma medio de exclusión $A \cup \overline{A} = X$

³Axioma de contradicción $A \cap \overline{A} = \emptyset$

2.2. Funciones de pertenencia

En general, cualquier función $\mu : X \rightarrow [0, 1]$ puede servir para describir un conjunto difuso, en la práctica la forma de la función debe reflejar el problema para el cual se está construyendo el conjunto difuso. Esta función debe mostrar nuestra percepción del concepto que se desea representar para resolver el problema, dado esto, es necesario describir los tipos de funciones de pertenencia más usados. Todas estas funciones están definidas en el universo de los números reales, una descripción más detallada en [11].

2.2.1. Tipos de funciones de pertenencia

Delta Es una función que toma el valor de 1 solamente en el punto \bar{x} del universo X y 0 en cualquier otro lugar, esta descrita por la siguiente forma

$$\mu(x) = \begin{cases} 1, & \text{si } x = \bar{x} \\ 0, & \text{si } x \neq \bar{x} \end{cases}$$

Esta función es muy importante, ya que se utiliza en los sistemas de inferencia como decodificador del sistema difuso si un valor puntual es necesario (ver Sección 2.3.1). Esta función de pertenencia se ilustra en la Figura 2.2.

Gaussiana Esta función tiene dos parámetros:

- \bar{x} es el centro de la función y
- σ es el ancho de la función.

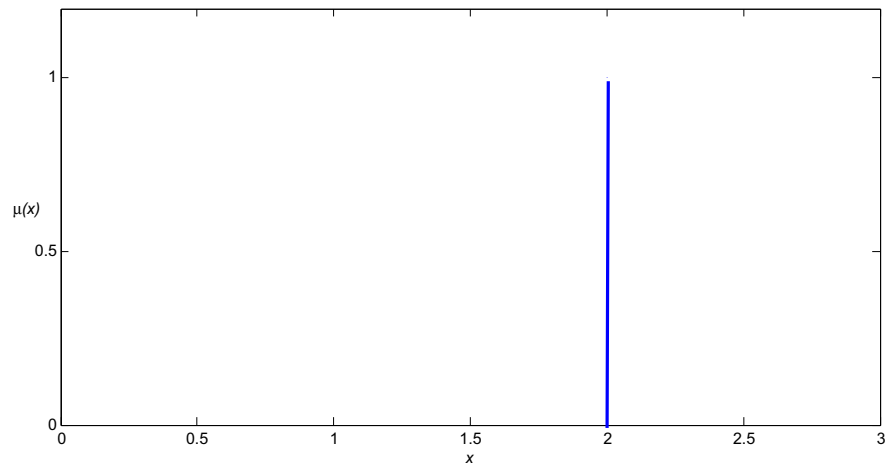


Figura 2.2: Función de pertenencia delta $\bar{x} = 2$

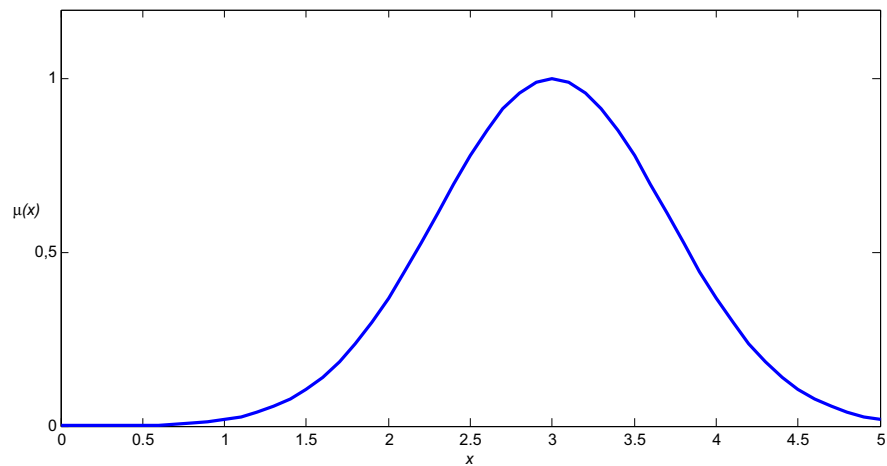


Figura 2.3: Función de pertenencia gaussiana $\bar{x} = 3$ y $\sigma = 1$

Y se expresa de la siguiente forma

$$\mu(x) = \exp\left(-\frac{(x - \bar{x})^2}{\sigma^2}\right)$$

Esta función de pertenencia se ilustra en la Figura 2.3.

Campana generalizada Esta función tiene tres parámetros:

- a es el ancho de la función
- c es el centro de la función y

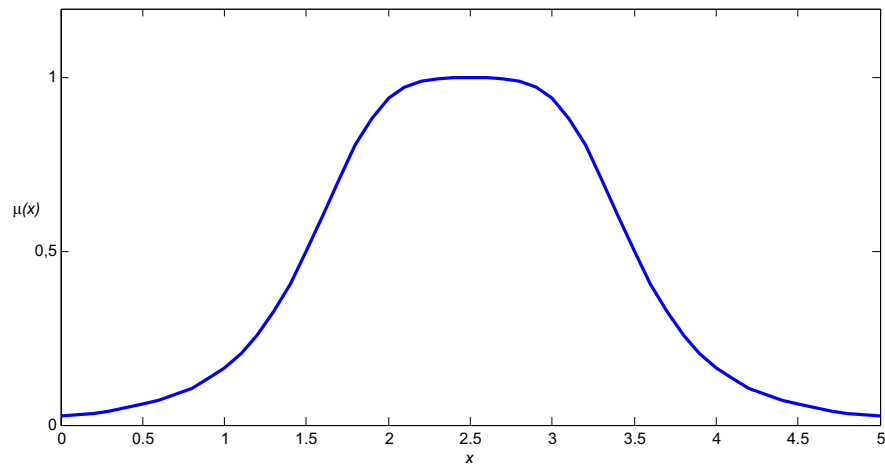


Figura 2.4: Función de pertenencia campana generalizada $a = 1$, $c = 2,5$ y $b = 2$

- b es la pendiente de la función.

Esta descrita de la siguiente forma

$$\mu(x) = \frac{1}{1 + \left| \frac{x - c}{a} \right|^{2b}}$$

Esta función de pertenencia se ilustra en la Figura 2.4.

Sigmoidal Esta función tiene dos parámetros:

- a es la pendiente en el punto de cruce y
- c es el punto de cruce.

Se expresa a continuación

$$\mu(x) = \frac{1}{1 + \exp[-a(x - c)]}$$

Esta función de pertenencia se ilustra en la Figura 2.5.

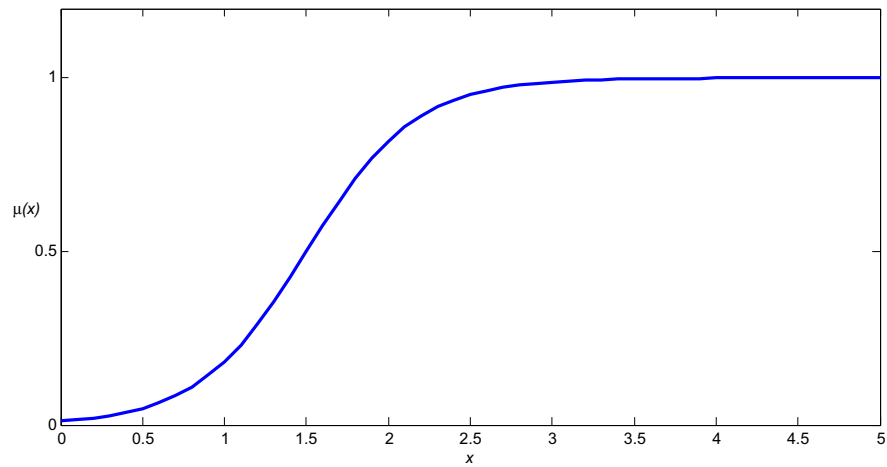


Figura 2.5: Función de pertenencia sigmoideal $a = 3$ y $c = 1,5$

Triangular Esta descrita por la siguiente forma

$$\mu(x) = \begin{cases} 0, & \text{si } x \leq a \\ \frac{x - a}{b - a}, & \text{si } a \leq x \leq b \\ \frac{c - x}{c - b}, & \text{si } b \leq x \leq c \\ 0, & \text{si } x \geq c \end{cases}$$

donde $a < b < c$ y b es el centro de la función. También se puede utilizar una forma alterna

$$\mu(x) = \max \left(\min \left(\frac{x - a}{b - a}, \frac{c - x}{c - b} \right), 0 \right)$$

Esta función de pertenencia se ilustra en la Figura 2.6.

Triangular simétrica Esta función tiene dos parámetros:

- \bar{x} es el centro de la función y
- σ es el ancho de la función.

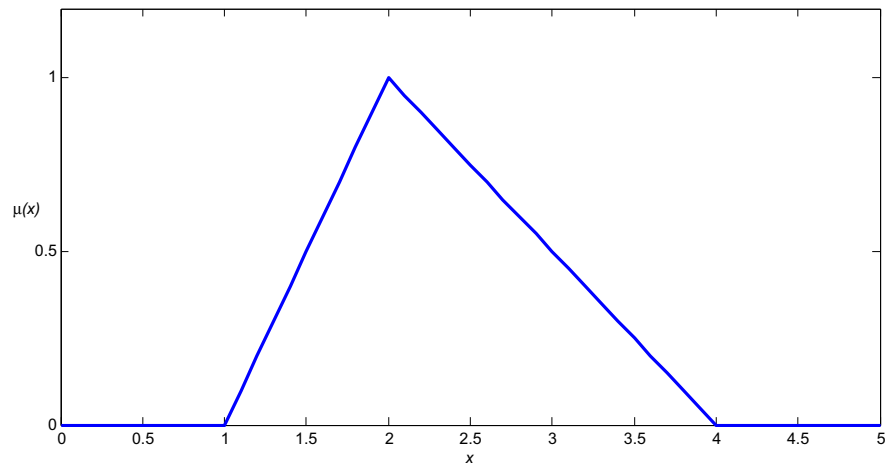


Figura 2.6: Función de pertenencia triangular $a = 1$, $b = 2$ y $c = 4$

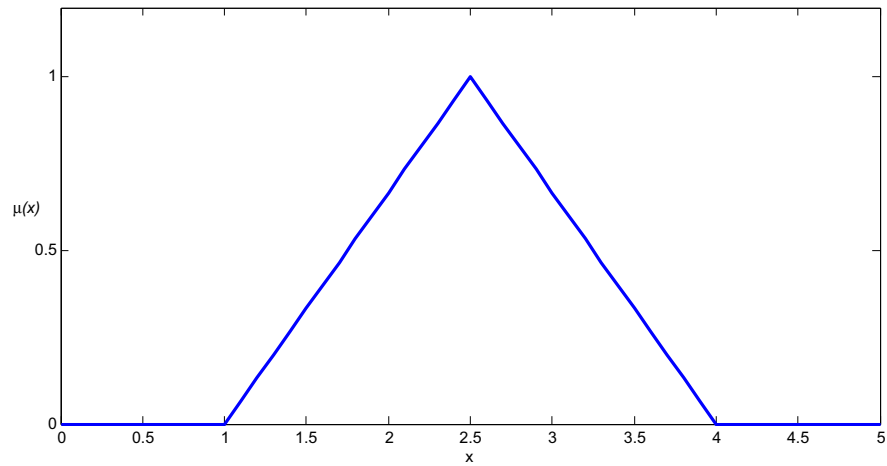


Figura 2.7: Función de pertenencia triangular simétrica $\bar{x} = 2,5$ y $\sigma = 1,5$

Y se expresa de la siguiente forma

$$\mu(x) = \max \left(\min \left(\frac{x - \bar{x} + \sigma}{\sigma}, \frac{\bar{x} + \sigma - x}{\sigma} \right), 0 \right)$$

Esta función de pertenencia se ilustra en la Figura 2.7.

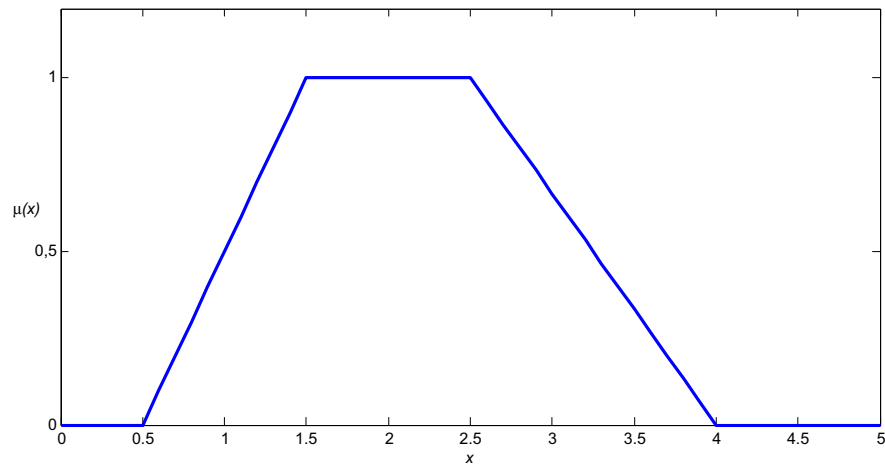


Figura 2.8: Función de pertenencia trapezoidal $a = 0,5$, $b = 1,5$, $c = 2,5$ y $d = 4$

Trapezoidal Esta descrita por los parámetros $a < b < c < d$ y tiene la forma

$$\mu(x) = \begin{cases} 0, & \text{si } x < a \\ \frac{x-a}{b-a}, & \text{si } a \leq x \leq b \\ 1 & \text{si } b \leq x \leq c \\ \frac{d-x}{d-c}, & \text{si } c \leq x \leq d \\ 0, & \text{si } d \leq x \end{cases}$$

Se puede utilizar una forma alterna

$$\mu(x) = \max \left(\min \left(\frac{x-a}{b-a}, 1, \frac{d-x}{d-c} \right), 0 \right)$$

Esta función de pertenencia se ilustra en la Figura 2.8.

Gamma Esta descrita por la siguiente forma,

$$\mu(x) = \begin{cases} 0, & \text{si } x \leq a \\ 1 - e^{k(x-a)^2}, & \text{si } x > a \end{cases}$$

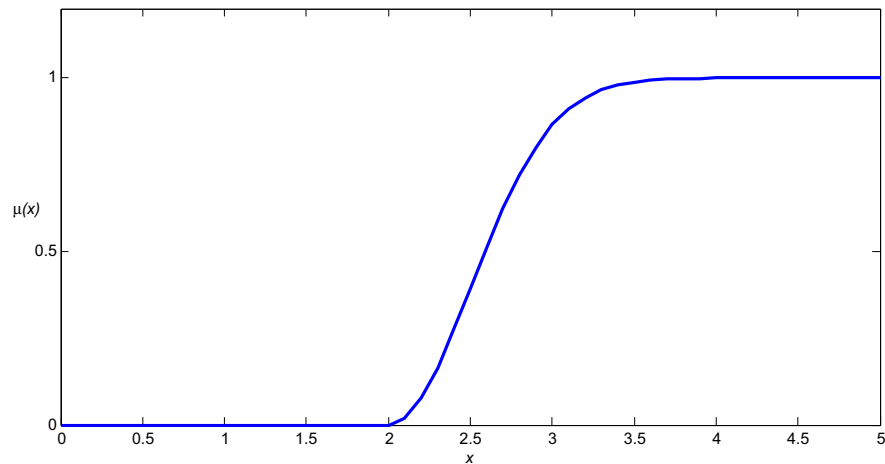


Figura 2.9: Función de pertenencia gamma $a = 2$ y $k = 2$

o

$$\mu(x) = \begin{cases} 0, & \text{si } x \leq a \\ \frac{k(x-a)^2}{1+k(x-a)^2}, & \text{si } x > a \end{cases}$$

Esta función de pertenencia se ilustra en la Figura 2.9.

Estos son solo algunos de los ejemplos de funciones que se pueden utilizar en los conjuntos difusos.

2.2.2. Características de las funciones de pertenencia

Dado que toda la información contenida en un conjunto difuso está descrita por su función de pertenencia, es necesario caracterizar estas funciones, para más detalles ver [12].

Núcleo Es la región del universo que se caracteriza por la completa pertenencia en el conjunto \tilde{A} . Los elementos que tienen un valor de pertenencia igual a 1 son los elementos del *núcleo* es decir

$$Nuc(\tilde{A}) = \{x \in X \mid \mu_{\tilde{A}}(x) = 1\}$$

Soporte Es la región de universo que se caracteriza por valores de pertenencia distintos de cero en el conjunto \tilde{A} . El *soporte* ($Sop(\tilde{A})$) de un conjunto difuso son los elementos cuyos valores de pertenencia son mayores a cero, es decir

$$Sop(\tilde{A}) = \{x \in X \mid \mu_{\tilde{A}}(x) > 0\}$$

Límites Es la región del universo que tiene valores de pertenencia distintos de cero pero a la vez no tienen completa pertenencia en el conjunto difuso \tilde{A} . Los *límites* ($Lim(\tilde{A})$) tienen elementos cuyos valores de pertenencia varían entre 0 y 1, es decir

$$Lim(\tilde{A}) = \{x \in X \mid 0 < \mu_{\tilde{A}}(x) < 1\}$$

Altura La altura de un conjunto difuso \tilde{A} es el máximo valor de la función de pertenencia, ver [6], es decir

$$hgt(\mu_{\tilde{A}}) = \text{máx}\{\mu_{\tilde{A}}(x)\}$$

Puntos de cruce Son los elementos en el universo para los cuales un conjunto difuso \tilde{A} tiene valores de pertenencia igual 0,5, es decir, para los cuales $\mu_{\tilde{A}}(x) = 0,5$.

2.2.3. Clasificación de los conjuntos difusos

Los conjuntos difusos pueden ser clasificados según su función de pertenencia, para más información ver [11], estos son:

Normal Es el conjunto cuya función de pertenencia tiene al menos un elemento x

en el universo cuya valor de pertenencia es la unidad. Formalmente, el conjunto difuso \tilde{A} es *normal* si su función de pertenencia alcanza el valor de 1, esto es

$$\sup_{x \in X} \tilde{A}(x) = 1$$

El supremo (sup) en la ecuación anterior también hace referencia a la altura del conjunto difuso \tilde{A} , $hgt(\mu_{\tilde{A}}) = \sup_{x \in X} \tilde{A}(x) = 1$.

Subnormal Si la función de pertenencia tiene valores menores a 1, entonces al conjunto se lo conoce como difuso subnormal. Si la altura $hgt(\mu_{\tilde{A}}) < 1$, el conjunto difuso es subnormal.

Convexo Es un conjunto cuya función de pertenencia tiene valores de pertenencia que son estrictamente crecientes o sus valores de pertenencia son estrictamente decrecientes o cuyos valores de pertenencia son crecientes y decrecientes para valores crecientes en los elementos en el universo. Esto es, si para cualquier x, y y z elementos del conjunto difuso \tilde{A} , la relación $x < y < z$ implica que

$$\mu_{\tilde{A}}(y) \geq \min[\mu_{\tilde{A}}(x), \mu_{\tilde{A}}(z)]$$

entonces \tilde{A} se dice que es un conjunto difuso convexo.

Simétrico Es el conjunto difuso cuya función de pertenencia es simétrica alrededor de un punto c

$$\mu_{\tilde{A}}(c + \Delta x) = \mu_{\tilde{A}}(c - \Delta x)$$

con $(c + \Delta x) \in X$ y $(c - \Delta x) \in X$.

Hay un mecanismo que permite convertir un conjunto difuso subnormal \tilde{A} en su contraparte normal. Esto se logra dividiendo la función de pertenencia para la altura

de conjunto difuso, de la siguiente manera

$$Norm(\tilde{A}) = \frac{\mu_{\tilde{A}}}{hgt(\mu_{\tilde{A}})}$$

Una propiedad especial de dos conjuntos difusos convexos \tilde{A} y \tilde{B} es que la intersección también es un conjunto difuso convexo. Es decir, para los conjuntos difusos convexos \tilde{A} y \tilde{B} se tiene que $\tilde{A} \cap \tilde{B}$ es convexo.

En este trabajo se utilizan funciones de pertenencia Gaussianas y Triangulares Simétricas, es decir, se utilizan conjuntos difusos Normales y Simétricos.

2.3. Sistemas de inferencia difusos

Los sistemas de inferencia difusos (SID) son conocidos como sistemas de regla base, para más detalles ver [13]. Estos constituyen la parte principal del sistema lógico difuso, ya que se encargan de formular reglas que sirven para describir el sistema. Los SID están basados en los conceptos de la teoría de conjuntos difusos, reglas difusas de la forma SI-ENTONCES y utilizan los conectores lógicos “Y” y “O” para crear las reglas de decisión. Los SID son únicos en el sentido de que pueden procesar simultáneamente información numérica y lingüística.

2.3.1. Estructura de un sistema de inferencia difuso

El sistema de inferencia difuso está compuesto por un codificador difuso, una regla base, una base de datos, una unidad de toma de decisiones, y finalmente un decodificador difuso. Un sistema de inferencia difuso se presenta en la Figura 2.10.

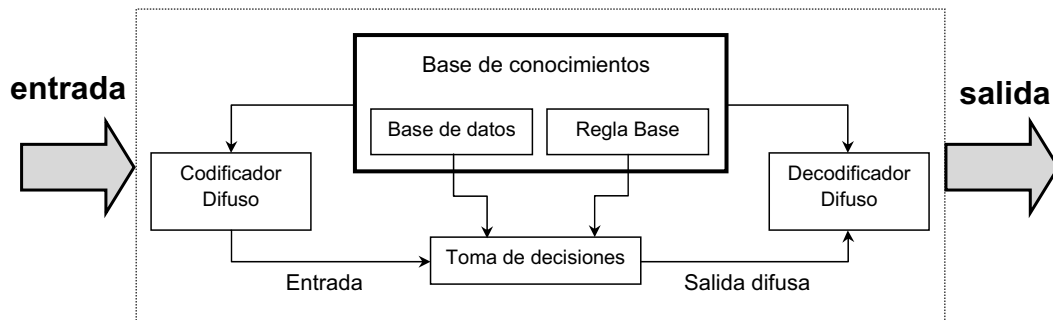


Figura 2.10: Sistema de inferencia difuso

La función de cada elemento es la siguiente:

Codificador difuso Se encarga de transformar las entradas puntuales (definidas en el universo X) en entradas difusas para alguna proposición en el antecedente;

Regla-base Está compuesta por un conjunto de reglas difusas

SI antecedente ENTONCES consecuente

se encarga de describir la relación entre las entradas y salidas;

Base de datos Aquí se almacenan los valores de los parámetros del modelo regla-base. Estos parámetros hacen referencia a factores de escalamiento y detalles de las funciones de pertenencia (como su tipo y parámetros que la conforman) de los conjuntos difusos usados;

Toma de decisiones Se encarga de ejecutar las operaciones de inferencia en las reglas, procesa las entradas usando la regla-base del sistema de inferencia difuso;

Decodificador difuso Transforma los resultados difusos de la inferencia en salidas puntuales.

El sistema de inferencia trabaja de la siguiente manera. La entrada puntual es transformada a una cantidad difusa utilizando el codificador difuso. A continuación, se forma la regla base. La base de datos y la regla base se conocen como la *base del conocimiento*. Y finalmente, el decodificador difuso se encarga de convertir los valores difusos en valores puntuales, los cuales son los resultados o la salida del sistema.

Los pasos del razonamiento difuso del sistema son:

- Comparar las variables de entrada con las funciones de pertenencia en la parte del antecedente para obtener los valores de pertenencia (proceso de codificación difusa).
- Combinar a través de un operador (generalmente la multiplicación o el mínimo) los valores de pertenencia en la parte de la premisa para obtener un *peso* de cada regla.
- Generar las consecuencias de cada regla dependiendo del peso.
- El conjunto de las consecuencias produce una salida puntual (proceso de decodificación difusa).

La proposición del antecedente es siempre del tipo “ x es A ” donde x es una variable lingüística y A es un término lingüístico constante. El grado de veracidad de la proposición depende de la similitud entre x y A . Dependiendo de la forma del consecuente podemos encontrar diferentes tipos de sistemas de inferencia, siendo los más importantes el sistema de inferencia de Mamdani y Takagi-Sugeno que se detallan a continuación.

2.3.2. Tipos de sistemas de inferencia difuso

Los sistemas de inferencia difusos más importantes son el sistema de inferencia de Mamdani propuesto por Mamdani y Assilian (1975) y el sistema de inferencia de Sugeno (1985) o Takagi-Sugeno, a este sistema también se lo conoce como sistema TS, que se describirán a continuación. La principal fuente bibliográfica de esta sección es [13].

Sistema de inferencia de Mamdani Este sistema de inferencia fue uno de los primeros en ser propuesto con el objetivo de controlar una máquina de vapor y una caldera, basado en un conjunto de reglas de control lingüístico construidas a partir del conocimiento y experiencia de los operadores. Este modelo se propuso como una forma de capturar la información cualitativa disponible en la forma de reglas si-entonces. El mecanismo de inferencia del sistema Mamdani puede ser explicado por un sistema de n -entradas y una salida descrita por M reglas difusas de tipo Mamdani.

$$\begin{aligned} R^1 &: \text{SI } x_1 \text{ es } \tilde{A}_1 \text{ y } \dots \text{ y } x_n \text{ es } \tilde{A}_n \text{ ENTONCES } y \text{ es } \tilde{B}^1 \\ R^2 &: \text{SI } x_1 \text{ es } \tilde{A}_1^2 \text{ y } \dots \text{ y } x_n \text{ es } \tilde{A}_n^2 \text{ ENTONCES } y \text{ es } \tilde{B}^2 \\ &\vdots \\ R^M &: \text{SI } x_1 \text{ es } \tilde{A}_1^M \text{ y } \dots \text{ y } x_n \text{ es } \tilde{A}_n^M \text{ ENTONCES } y \text{ es } \tilde{B}^M \end{aligned}$$

Para un conjunto de x_i entradas, con $i = 1, 2, \dots, n$. El objetivo es determinar el valor de salida puntual, el cual es un valor decodificado del conjunto de salida difuso; el mecanismo de inferencia procede de la siguiente manera:

1. Se calcula el grado de satisfacción de cada regla para cualquier conjunto

de entrada (entrada puntual) por medio

$$\beta^l = \mu_{A_1}^{\sim l}(x_1) \wedge \mu_{A_2}^{\sim l}(x_2) \wedge \cdots \wedge \mu_{A_n}^{\sim l}(x_n)$$

donde \wedge es el operador mínimo o producto.

2. Se determina el consecuente de cada regla dado por

$$\tilde{B}^{\prime l} = \beta^l \wedge \tilde{B}^l$$

3. Se suma todos los conjuntos difusos consecuentes

$$\tilde{B}'_{sum} = \bigcup_{l=1}^M \tilde{B}^{\prime l} = (\tilde{B}^{\prime 1} \cup \tilde{B}^{\prime 2} \cup \cdots \cup \tilde{B}^{\prime M})$$

4. Finalmente se decodifica el conjunto difuso agregado \tilde{B}'_{sum} , calculando la coordenada x' del centro de gravedad del conjunto difuso \tilde{B}'_{sum}

$$x' = \frac{\sum_{i=1}^n \mu_{\tilde{B}'_{sum}}^{\sim l}(x_i) \cdot x_i}{\sum_{i=1}^n \mu_{\tilde{B}'_{sum}}^{\sim l}(x_i)}$$

a esta técnica de decodificación difusa se lo conoce como el centro método del centro de gravedad (CG).

Sistema de inferencia de Takagi-Sugeno En este sistema está compuesto por dos etapas. En la primera, se calcula el grado de cumplimiento o satisfacción de cada regla utilizando el producto. Finalmente, se calcula el valor final de la salida del sistema utilizando un decodificador promedio ponderado. También

hay que considerar que se modifica la implicación (consecuente) del sistema de inferencia debido a que se utiliza funciones lineales en lugar de conjuntos difusos.

El sistema de inferencia difuso Takagi-Sugeno, para M reglas, se representa como

$$\begin{aligned}
 R^1 &: \text{SI } x_1 \text{ es } \tilde{A}_1^1 \text{ y } \dots \text{ y } x_n \text{ es } \tilde{A}_n^1 \text{ ENTONCES } y^1 = \theta_0^1 + \theta_1^1 x_1 + \dots + \theta_n^1 x_n \\
 R^2 &: \text{SI } x_1 \text{ es } \tilde{A}_1^2 \text{ y } \dots \text{ y } x_n \text{ es } \tilde{A}_n^2 \text{ ENTONCES } y^2 = \theta_0^2 + \theta_1^2 x_1 + \dots + \theta_n^2 x_n \\
 &\vdots \\
 R^M &: \text{SI } x_1 \text{ es } \tilde{A}_1^M \text{ y } \dots \text{ y } x_n \text{ es } \tilde{A}_n^M \text{ ENTONCES } y^M = \theta_0^M + \theta_1^M x_1 + \dots + \theta_n^M x_n
 \end{aligned}$$

El sistema de inferencia del modelo Takagi-Sugeno procede de la siguiente manera:

1. Se calcula el peso de cada regla difusa, utilizando el operador producto

$$\beta^l = \prod_{i=1}^n \mu_{\tilde{A}_i^l}(x_i) = \mu_{\tilde{A}_1^l}(x_1) \times \mu_{\tilde{A}_2^l}(x_2) \times \dots \times \mu_{\tilde{A}_n^l}(x_n) ; l = 1, 2, \dots, M$$

2. El valor de salida del sistema está dado por

$$y_0 = \frac{\sum_{l=1}^M \beta^l \cdot y^l}{\sum_{l=1}^M \beta^l} = \frac{\sum_{l=1}^M \beta^l \cdot (\theta_0^l + \theta_1^l x_1 + \dots + \theta_n^l x_n)}{\sum_{l=1}^M \beta^l}$$

En su lugar, la salida del sistema también se representa por la siguiente

forma normalizada

$$y_0 = \sum_{l=1}^M \gamma^l \cdot y^l = \sum_{l=1}^M \gamma^l (\theta_0^l + \theta_1^l x_1 + \dots + \theta_n^l x_n)$$

donde

$$\gamma^l = \frac{\beta^l}{\sum_{l=1}^M \beta^l}$$

es el grado de pertenencia normalizado.

A este sistema de inferencia se le llama modelo *afín de Takagi-Sugeno* y proporciona el marco adecuado para la descomposición de un sistema no lineal en modelos lineales localmente válidos, muy apropiado para aplicar técnicas de control clásicas.

La principal diferencia entre estos dos sistemas se encuentra en el consecuente de la regla difusa. El sistema de inferencia de Mamdani utiliza conjuntos difusos en el consecuente, mientras que el sistema de Takagi-Sugeno emplea funciones lineales de las variables de entrada.

Ventajas del sistema Mamdani :

- Es intuitivo.
- Tiene una amplia aceptación.
- Se adapta muy bien al diseño de expertos.

Ventajas del sistema Takagi-Sugeno :

- Es computacionalmente eficiente.

- Trabaja bien con técnicas lineales.
- Se adapta bien al análisis matemático.

Para mayor detalle se puede revisar [14].

Este capítulo sirve para presentar los conceptos fundamentales sobre los que se sustenta este trabajo de investigación, conceptos que son aplicados en las diferentes propuestas sobre métodos automáticos de regresión difusa que se encuentran en la literatura, y sirven para sustentar el método propuesto.

CAPÍTULO 3

MÉTODOS AUTOMÁTICOS DE REGRESIÓN DIFUSA

El desarrollo de la teoría de conjuntos difusos fue motivado por la percepción de que las técnicas tradicionales de análisis no son muy efectivas cuando trataban con problemas en los que las relaciones entre las variables son muy complejas, o mal definidas como para ser caracterizadas por un modelo específico. Un aspecto común que se presenta en estos problemas es la falta de conocimiento, imprecisión e incertidumbre del fenómeno que se está analizando, características que se presenta muy a menudo en diferentes campos de la ciencia e ingeniería.

Por lo general, la presencia de incertidumbre en un modelo de regresión tradicional se refleja en los intervalos de confianza, generalmente con un nivel del 95 %. Sin embargo, se pueden encontrar otras formas de incertidumbre en la ambigüedad, problemas de especificación o factores no considerados debido a la complejidad del fenómeno, que no se atribuyen a la aleatoriedad, ver [24]. Esto se traduce en la creación de restricciones para los modelos, de manera que estos permitan obtener estimaciones más ajustadas. Estas restricciones se pueden implementar en algunos modelos de regresión probabilística con consecuencias en algunos casos para los supuestos teóricos de tales modelos. Lo anterior no ocurre con la regresión difusa. En este contexto, una herramienta versátil son los métodos automáticos de regresión difusa los cuales constituyen opciones competitivas comparado con técnicas tradicionales de modelación.

En este capítulo se describe cómo utilizar los sistemas de inferencia difusos pa-

ra la estimación e identificación en los métodos automáticos de regresión difusa. Una descripción más detallada se puede encontrar en Passino Kevin y Yurkovich Stephen [15] y Ross Timothy [16], los cuales constituyen las principales fuentes bibliográficas de este capítulo. Para el desarrollo de la teoría, en primer lugar se detallan algunas definiciones y luego se describen brevemente los métodos de mínimos cuadrados por lotes, mínimos cuadrados recursivo y aprendizaje desde el ejemplo modificado.

3.1. Principales definiciones

Sea g una función que describe el sistema en estudio, definida por

$$g : \bar{X} \longrightarrow \bar{Y}$$

donde $\bar{X} \subset \mathbb{R}^n$ y $\bar{Y} \subset \mathbb{R}$. Se quiere construir un sistema difuso, representado por la función

$$f : X \longrightarrow Y$$

donde $X \subset \bar{X}$ y $Y \subset \bar{Y}$ son el dominio y recorrido respectivamente. Mediante la elección de un vector de parámetros θ se espera aproximar la función g de la forma:

$$g(x) = f(x | \theta) + e(x), \quad (3.1)$$

para todo $x = [x_1, x_2, \dots, x_n]^\top \in X$; donde: n es el número de variables de entrada, $e(x)$ representa el error aproximación el cual se espera sea mínimo. La i -ésima entrada-salida de la función g se nota como (x^i, y^i) , $x^i \in X$, $y^i \in Y$ y $y^i = g(x^i)$, para $i = 1, \dots, M$. Luego $x^i = [x_1^i, x_2^i, \dots, x_n^i]^\top$ corresponde el vector

de entradas para el i -ésimo par de datos. El conjunto de parejas ordenadas de entrada-salida del sistema se conoce como *conjunto de datos de entrenamiento* y se nota

$$G = \{(x^1, y^1), \dots, (x^M, y^M)\} \subset X \times Y \quad (3.2)$$

donde M es la cardinalidad de G .

Los métodos automáticos presentados en este capítulo generan reglas base, es decir reglas que sirven para describir la relación que existe entre las variables de entrada y salida, o usan una regla base predeterminada para modelar el sistema. En cualquier caso las reglas están compuestas por antecedentes y consecuentes, de la forma

Si antecedente ENTONCES consecuente,

las reglas permiten predecir y/o gobernar la salida del sistema con conocimiento a priori de sus entradas, ver Sección 2.3.

En este trabajo, la mayoría de métodos que se presentan incorporan las funciones de pertenencia Gaussiana para las variables de entrada, pues describen una amplia gama de fenómenos y se utiliza la función Delta como función de pertenencia de salida. También se puede utilizar funciones de pertenencia triangulares. Sin embargo, los métodos que se presentan a continuación pueden ser modificados para cualquier tipo de función de pertenencia, como por ejemplo las descritas en la Sección 2.2.

3.2. Mínimos cuadrados por lotes (MCL)

En este método se aplica el estimador de mínimos cuadrados ordinarios a conjuntos difusos. Para aplicarlo, es necesario caracterizar el comportamiento del sistema mediante una regla-base (centros y dispersiones de las funciones de pertenencia para las variables de entrada y salida). En los casos donde se desconoce el comportamiento del sistema es posible utilizar otros algoritmos con la capacidad de formar reglas, como los métodos de aprendizaje desde el ejemplo, para más información ver [16].

A continuación, se describe la regresión lineal tradicional para visualizar su analogía con este método, para una mejor descripción ver [17].

3.2.1. Mínimos cuadrados ordinarios

Se asume que, la variable dependiente está relacionada con otras variables independientes o explicativas.

Suponiendo que hay M valores para estas variables. Sea y^i la i -ésima observación de la variable dependiente y sea $(x_1^i, x_2^i, \dots, x_n^i)$ la i -ésima observación de las n variables explicativas, definidas de la misma manera que en la sección anterior. A continuación, se presenta la notación matricial que será muy útil para determinar el estimador de mínimos cuadrados. Se definen los vectores n -dimensionales

$$x^i = \begin{bmatrix} x_1^i \\ x_2^i \\ \vdots \\ x_n^i \end{bmatrix} \quad \text{y} \quad \beta = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{bmatrix}$$

Utilizando el producto interno de vectores, i.e., $x^{i\top}\beta = \beta_1x_1^i + \beta_2x_2^i + \dots + \beta_nx_n^i$, se reescribe el modelo como:

$$y^i = x^{i\top}\beta + \epsilon^i \quad i = 1, 2, \dots, M$$

y se nota

$$Y = \begin{bmatrix} y^1 \\ y^2 \\ \vdots \\ y^M \end{bmatrix}, \epsilon = \begin{bmatrix} \epsilon^1 \\ \epsilon^2 \\ \vdots \\ \epsilon^M \end{bmatrix}, X = \begin{bmatrix} x^{1\top} \\ x^{2\top} \\ \vdots \\ x^{M\top} \end{bmatrix} = \begin{bmatrix} x_1^1 \dots x_n^1 \\ x_1^2 \dots x_n^2 \\ \vdots \dots \vdots \\ x_1^M \dots x_n^M \end{bmatrix}$$

Dado que el número de columnas de X es igual al número de filas de β , se puede escribir

$$Y = X\beta + \epsilon \tag{3.3}$$

Se puede calcular el error utilizando un estimador $\hat{\beta}$ de β como

$$y^i - x^{i\top}\hat{\beta},$$

a este término se le conoce como residuo de la observación i . A continuación se forma la *suma de cuadrados residual* (SCR)

$$SCR(\hat{\beta}) = \sum_{i=1}^M (y^i - x^{i\top}\hat{\beta})^2 = (Y - X\hat{\beta})^\top (Y - X\hat{\beta})$$

luego

$$\begin{aligned} SCR(\hat{\beta}) &= (Y - X\hat{\beta})^T(Y - X\hat{\beta}) \\ &= (Y^T - \hat{\beta}^T X^T)(Y - X\hat{\beta}) \\ &= Y^T Y - \hat{\beta}^T X^T Y - Y^T X\hat{\beta} + \hat{\beta}^T X^T X\hat{\beta} \\ &= Y^T Y - 2Y^T X\hat{\beta} + \hat{\beta}^T X^T X\hat{\beta} \end{aligned} \quad (3.4)$$

al derivar con respecto a $\hat{\beta}$ (3.4)

$$\frac{\partial SCR(\hat{\beta})}{\partial \hat{\beta}} = -2X^T Y + 2X^T X\hat{\beta}$$

e igualando a cero se tiene

$$X^T X\hat{\beta} = X^T Y$$

Donde $\hat{\beta}$ es el estimador de mínimos cuadrados. A este conjunto de ecuaciones se las conoce como *ecuaciones normales*. Si la matriz $X^T X$ es no singular entonces

$$\hat{\beta} = (X^T X)^{-1} X^T Y \quad (3.5)$$

En el método de mínimos cuadrados ponderado se utiliza una matriz diagonal W , $M \times M$ cuyos elementos $w_{ii} > 0$ para todo $i = 1, 2, \dots, M$ y los elementos fuera de la diagonal son cero. Los w_{ii} pueden ser usados como ponderadores para los elementos de G . En este caso el estimador está definido por

$$\hat{\beta} = (X^T W X)^{-1} X^T W Y \quad (3.6)$$

3.2.2. Mínimos cuadrados por lotes

Para aplicar los mínimos cuadrados, se utiliza el sistema difuso definido a continuación (para más información ver [16]),

$$f(x^i | \theta) = \frac{\sum_{l=1}^R b_l \mu_{il}}{\sum_{l=1}^R \mu_{il}} \quad i = 1, \dots, M \quad (3.7)$$

donde,

$$\mu_{il} = \prod_{j=1}^n \exp \left[-\frac{1}{2} \left(\frac{x_j^i - c_j^l}{\sigma_j^l} \right)^2 \right] \quad l = 1, \dots, R \quad (3.8)$$

R representa el número de reglas en la regla-base y n es la dimensión del vector x . Es importante señalar que el parámetro R se desconoce para algunos métodos, en algunos casos es determinado por el algoritmo directamente. Los b_l son los centros de las funciones de pertenencia de salida y los c_j^l, σ_j^l son los centros y dispersiones de las funciones de pertenencia de entrada respectivamente. Hay que señalar que la dispersión relativa, σ_j^l de la j -ésima variable de entrada de la l -ésima regla es mayor que cero. Los centros y dispersiones de las funciones de pertenencia de entrada se pueden inicializar utilizando un proceso heurístico.

Observación 3. El sistema difuso (3.7)-(3.8) se utiliza en todos los métodos descritos. Es importante señalar que el parámetro R se desconoce para algunos métodos, en algunos casos es determinado por el algoritmo directamente.

Para la inicialización de los centros se utiliza la siguiente ecuación

$$c_j^l = x_j^l + \frac{1}{2}(x_j^{l+1} - x_j^l) \quad l = 1, \dots, R, j = 1, \dots, n \quad (3.9)$$

A las dispersiones se debe asignar una constante mayor que cero

$$\sigma_j^l = \sigma > 0 \quad l = 1, \dots, R, j = 1, \dots, n \quad (3.10)$$

El valor de σ se determina de manera heurística.

Luego por (3.7)

$$f(x^i | \theta) = \frac{b_1 \mu_{i1}}{\sum_{l=1}^R \mu_{il}} + \frac{b_2 \mu_{i2}}{\sum_{l=1}^R \mu_{il}} + \dots + \frac{b_R \mu_{iR}}{\sum_{l=1}^R \mu_{il}} \quad i = 1, \dots, M$$

si se nota:

$$\xi_{il} = \frac{\mu_{il}}{\sum_{l=1}^R \mu_{il}} \quad l = 1, \dots, R,$$

entonces,

$$f(x^i | \theta) = b_1 \xi_{i1} + b_2 \xi_{i2} + \dots + b_R \xi_{iR}$$

es decir,

$$y^i = f(x^i | \theta) = \theta^\top \xi^i \quad (3.11)$$

donde:

$$\theta = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_R \end{bmatrix} \quad \text{y} \quad \xi^i = \begin{bmatrix} \xi_1^i \\ \xi_2^i \\ \vdots \\ \xi_R^i \end{bmatrix} \quad i = 1, \dots, M$$

Se puede apreciar que (3.11) es similar al modelo de regresión lineal (3.3). La diferencia radica que en lugar de la matriz X , se consideran los números difusos de la matriz ξ , es decir, los lotes o particiones formadas por las funciones de pertenencia de las variables de entrada, ver Passino Kevin y Yurkovich Stephen [15].

Finalmente, si se nota $\Phi = \xi$ y se aplica el método de mínimos cuadrados, el estimador está dado por

$$\hat{\theta} = (\Phi^T \Phi)^{-1} \Phi^T y \quad (3.12)$$

note la similitud con 3.5.

El método de mínimos cuadrados por lotes ha probado ser muy eficiente para una gran variedad de aplicaciones. Sin embargo, en algunas aplicaciones la matriz $\Phi^T \Phi$ podría estar mal condicionada, debido a que la dimensión de Φ depende de M .

Este método se encuentra esquematizado en el Algoritmo 1.

Algoritmo 1 Mínimos cuadrados por lotes

Entrada: $X_{M \times n}$ matriz de las variables de entrada

Y_M matriz de la variable de salida

$\sigma > 0$

1: $R \leftarrow M - 1$

2: $c_j^l \leftarrow x_j^l + \frac{1}{2}(x_j^{l+1} - x_j^l) \quad l = 1, \dots, R, j = 1, \dots, n$

3: $\sigma_j^l \leftarrow \sigma \quad l = 1, \dots, R, j = 1, \dots, n$

4: **para** $i \leftarrow 1, M$ **hacer**

5: **para** $l \leftarrow 1, R$ **hacer**

6: $\mu_{il} \leftarrow \prod_{j=1}^n \exp \left[-\frac{1}{2} \left(\frac{x_j^i - c_j^l}{\sigma_j^l} \right)^2 \right]$

7: **fin para**

8: $\xi_{il} \leftarrow \frac{\mu_{il}}{\sum_{l=1}^R \mu_{il}}$

9: **fin para**

10: $\Phi \leftarrow \xi(x)$

11: $\hat{\theta} \leftarrow (\Phi^T \Phi)^{-1} \Phi^T y$

12: **devolver** $f(x | \hat{\theta}) \leftarrow \hat{\theta}^T \xi(x)$

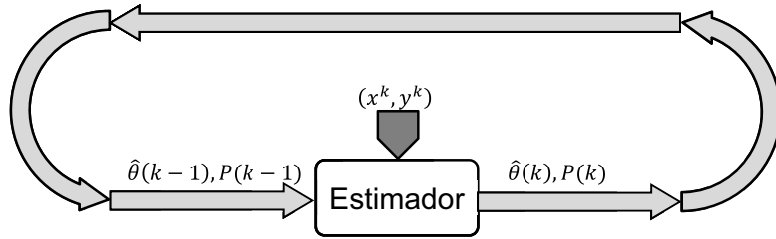


Figura 3.1: Esquema de funcionamiento de mínimos cuadrados recursivo

3.3. Mínimos cuadrados recursivo (MCR)

El método de mínimos cuadrados recursivo es, como su nombre lo indica, una versión recursiva del método de mínimos cuadrados por lotes. Este método utiliza una parte del conjunto de entrenamiento y no es necesario el cálculo de la inversa de $\Phi^T \Phi$, de esta manera es más eficiente y estable computacionalmente. Al ser un algoritmo recursivo, $\hat{\theta}(k)$ se calcula mediante $\hat{\theta}(k-1)$, x^k , y y^k , una vez que se ha calculado $\hat{\theta}(k)$ se repite la iteración K veces, donde k es un ciclo sobre el conjunto de entrenamiento G y K es el número de iteraciones. La Figura 3.1 muestra el esquema de funcionamiento de este método, para más detalles ver [18].

Al igual que para el MCL, se utiliza la función de pertenencia Gaussiana para las entradas y una función Delta para las salidas; b_i, c_j^l se definen como en la Sección 3.2. Cabe señalar que el ancho relativo, σ_j^l de la j -ésima variable de entrada de la l -ésima regla es mayor que cero. Para la inicialización de los centros y dispersiones se utiliza las ecuaciones (3.9) y (3.10) de la sección anterior.

Sea $P(k)$ la matriz $n \times n$ definida como

$$P(k) = (\Phi^T \Phi)^{-1} = \left(\sum_{i=1}^k x^i (x^i)^T \right)^{-1}, \quad (3.13)$$

y notemos $\hat{\theta}(k-1)$ el estimador de mínimos cuadrados en el instante $k-1$. Se asume que $\Phi^T \Phi$ es no singular para todo k , entonces

$$P^{-1}(k) = \sum_{i=1}^{k-1} x^i (x^i)^T + x^k (x^k)^T,$$

por tanto

$$P^{-1}(k) = P^{-1}(k-1) + x^k (x^k)^T, \quad (3.14)$$

por (3.6) se tiene

$$\begin{aligned} \hat{\theta}(k) &= (\Phi^T \Phi)^{-1} \Phi^T Y = \left(\sum_{i=1}^k x^i (x^i)^T \right)^{-1} \left(\sum_{i=1}^k x^i y^i \right) \\ &= P(k) \left(\sum_{i=1}^{k-1} x^i y^i + x^k y^k \right) \end{aligned} \quad (3.15)$$

entonces,

$$\hat{\theta}(k-1) = P(k-1) \sum_{i=1}^{k-1} x^i y^i$$

y así

$$P^{-1}(k-1) \hat{\theta}(k-1) = \sum_{i=1}^{k-1} x^i y^i,$$

reemplazando $P^{-1}(k-1)$ en la ecuación anterior; de (3.14) se obtiene

$$(P^{-1}(k) - x^k (x^k)^T) \hat{\theta}(k-1) = \sum_{i=1}^{k-1} x^i y^i$$

finalmente por (3.15)

$$\begin{aligned} \hat{\theta}(k) &= P(k) (P^{-1}(k) - x^k (x^k)^T) \hat{\theta}(k-1) + P(k) x^k y^k \\ &= \hat{\theta}(k-1) + P(k) x^k (y^k - (x^k)^T \hat{\theta}(k-1)) \end{aligned} \quad (3.16)$$

3.15 permite calcular el estimador $\hat{\theta}(k)$ en cada paso k , donde el error está dado

por $(y^k - (x^k)^\top \hat{\theta}(k-1))$.

Para actualizar $\hat{\theta}$ en (3.16) se requiere $P(k)$; tomando en cuenta que

$$P^{-1}(k) = P^{-1}(k-1) + x^k(x^k)^\top \quad (3.17)$$

se debe calcular la inversa en cada paso, lo cual tiene un elevado costo computacional, si el conjunto de datos es muy grande. Para solventar esto, se utiliza el Lema de la Matriz Inversa¹, ver [19].

Nótese que

$$\begin{aligned} P(k) &= (\Phi^\top(k)\Phi(k))^{-1}, \\ &= (P^{-1}(k-1) + x^k(x^k)^\top)^{-1}, \end{aligned}$$

por lema de la matriz inversa con $A = P^{-1}(k-1)$, $B = x^k$, $C = I$, y $D = (x^k)^\top$, se obtiene

$$P(k) = P(k-1) - P(k-1)x^k(I + (x^k)^\top P(k-1)x^k)^{-1}(x^k)^\top P(k-1) \quad (3.18)$$

y

$$\hat{\theta}(k) = \hat{\theta}(k-1) + P(k)x^k(y^k - (x^k)^\top \hat{\theta}(k-1)) \quad (3.19)$$

Con la aplicación del lema de la matriz inversa se evita la inversión de una matriz; ya que en su lugar únicamente se calcula el inverso del escalar $(I + (x^k)^\top P(k-1)x^k)^{-1}$. Las ecuaciones (3.18) y (3.19) caracterizan $P(k)$ y $\hat{\theta}(k)$ y permiten describir el método de mínimos cuadrados recursivo.

¹Sean A , C y $(C^{-1} + DA^{-1}B)$ matrices cuadradas no singulares, entonces $A + BCD$ es invertible y $(A + BCD)^{-1} = A^{-1} - A^{-1}B(C^{-1} + DA^{-1}B)^{-1}DA^{-1}$. Al lema de la matriz inversa también se lo conoce en la literatura como la identidad de Woodbury.

Para inicializar el algoritmo, es decir, elegir $\hat{\theta}(0)$ y $P(0)$, se utilizan generalmente $\hat{\theta}(0) = 0$ y $P(0) = P_0 = \alpha I$ para algún $\alpha > 0$. Otra opción es mediante una aproximación heurística.

Al igual que el método de mínimos cuadrados por lotes, se obtiene un algoritmo de mínimos cuadrados recursivo ponderado, en este caso se utiliza un factor $0 < \lambda \leq 1$ que se lo conoce como un “factor de pérdida de memoria” y sirve para dar mayor peso a la información más recientes en la optimización. Cuando $\lambda = 1$ se obtiene el método MCR original. En el método de mínimos cuadrados recursivo $P(k)$ se define como:

$$\begin{aligned} P(k) &= \frac{1}{\lambda} (I - P(k-1)x^k(\lambda I + (x^k)^\top P(k-1)x^k)^{-1}(x^k)^\top) P(k-1) \\ \hat{\theta}(k) &= \hat{\theta}(k-1) + P(k)x^k(y^k - (x^k)^\top \hat{\theta}(k-1)) \end{aligned} \quad (3.20)$$

En el sistema difuso (3.7), se reemplaza x^k con $\xi(x^k)$ en (3.20)

$$\begin{aligned} P(k) &= \frac{1}{\lambda} \left\{ I - P(k-1)\xi(x^k) [\lambda I + (\xi(x^k))^\top P(k-1)\xi(x^k)]^{-1} (\xi(x^k))^\top \right\} P(k-1) \\ \hat{\theta}(k) &= \hat{\theta}(k-1) + P(k)\xi(x^k) \left[y^k - (\xi(x^k))^\top \hat{\theta}(k-1) \right] \end{aligned}$$

lo cual se esquematiza en el Algoritmo 2.

Se puede apreciar en el Algoritmo 2 que el cálculo de μ_{ik} , $P(k)$ y $\hat{\theta}(k)$ se encuentra anidado dentro del lazo de la línea 5. Dado que los centros c_j^l y las dispersiones σ_j^l de las funciones de pertenencia de las variables de entrada son constantes, los valores de pertenencia μ_{ik} requieren de una iteración para su cálculo, lo cual se repite $K \times M$, creando una matriz de tamaño $K \times M$. Por esta razón, es necesario realizar una modificación de este algoritmo, con el fin de optimizar tiempo y recursos computacionales, esto se esquematiza en el Algoritmo 3.

Algoritmo 2 Mínimos cuadrados recursivo

Entrada: $X_{M \times n}$ matriz de las variables de entrada

Y_M matriz de la variable de salida

K número de iteraciones del algoritmo

$\sigma > 0, \alpha > 0, 0 < \lambda \leq 1$

1: $\hat{\theta}(0) \leftarrow 0$

2: $P(0) \leftarrow \alpha I$

3: $c_j^l \leftarrow x_j^l + \frac{1}{2}(x_j^{l+1} - x_j^l) \quad l = 1, \dots, R, j = 1, \dots, n$

4: $\sigma_j^l \leftarrow \sigma \quad l = 1, \dots, R, j = 1, \dots, n$

5: **para** $k \leftarrow 1, K \times M$ **hacer**

6: **para** $l \leftarrow 1, R$ **hacer**

7: $\mu_{ik} \leftarrow \prod_{j=1}^n \exp \left[-\frac{1}{2} \left(\frac{x_j^i - c_j^l}{\sigma_j^l} \right)^2 \right]$

8: **fin para**

9: $\xi_{ik} \leftarrow \frac{\mu_{ik}}{\sum_{l=1}^R \mu_{ik}}$

10: $P(k) \leftarrow \frac{1}{\lambda} \left\{ I - P(k-1)\xi(x^k) [\lambda I + (\xi(x^k))^\top P(k-1)\xi(x^k)]^{-1} (\xi(x^k))^\top \right\} P(k-1)$

11: $\hat{\theta}(k) \leftarrow \hat{\theta}(k-1) + P(k)\xi(x^k) \left[y^k - (\xi(x^k))^\top \hat{\theta}(k-1) \right]$

12: **fin para**

13: **devolver** $f(x | \hat{\theta}) \leftarrow \hat{\theta}^\top \xi(x)$

Algoritmo 3 Mínimos cuadrados recursivo modificado

Entrada: $X_{M \times n}$ matriz de las variables de entrada

Y_M matriz de la variable de salida

K número de iteraciones del algoritmo

$\sigma > 0, \alpha > 0, 0 < \lambda \leq 1$

1: $\hat{\theta}(0) \leftarrow 0$

2: $P(0) \leftarrow \alpha I$

3: $c_j^l \leftarrow x_j^l + \frac{1}{2}(x_j^{l+1} - x_j^l) \quad l = 1, \dots, R, j = 1, \dots, n$

4: $\sigma_j^l \leftarrow \sigma \quad l = 1, \dots, R, j = 1, \dots, n$

5: **para** $i \leftarrow 1, M$ **hacer**

6: **para** $l \leftarrow 1, R$ **hacer**

7: $\mu_{il} \leftarrow \prod_{j=1}^n \exp \left[-\frac{1}{2} \left(\frac{x_j^i - c_j^l}{\sigma_j^l} \right)^2 \right]$

8: **fin para**

9: $\xi_{il} \leftarrow \frac{\mu_{il}}{\sum_{l=1}^R \mu_{il}}$

10: **fin para**

11: $s \leftarrow 0$ contador auxiliar

12: **para** $k \leftarrow 1, K \times M$ **hacer**

13: **si** $s \leq M$ **entonces**

14: $s \leftarrow s + 1$

15: **si no**

16: $s \leftarrow 1$

17: **fin si**

18: $P(k) \leftarrow \frac{1}{\lambda} \left\{ I - P(k-1)\xi(x^s) [\lambda I + (\xi(x^s))^\top P(k-1)\xi(x^s)]^{-1} (\xi(x^s))^\top \right\} P(k-1)$

19: $\hat{\theta}(k) \leftarrow \hat{\theta}(k-1) + P(k)\xi(x^s) \left[y^s - (\xi(x^s))^\top \hat{\theta}(k-1) \right]$

20: **fin para**

21: **devolver** $f(x | \hat{\theta}) \leftarrow \hat{\theta}^\top \xi(x)$

3.4. Aprendizaje desde el ejemplo modificado (AEM)

En esta sección se discute un enfoque muy intuitivo para la construcción de un sistema difuso f para que se aproxime a la función g , el cual se conoce como aprendizaje desde el ejemplo modificado. Este método se basa en técnicas de optimización para determinar los parámetros del sistema difuso. En este caso se generan automáticamente reglas y funciones de pertenencia, por esta razón este método es una alternativa viable a problemas en los cuales se carece de información *a priori* del sistema.

De manera análoga a la sección anterior se consideran funciones de pertenencia Gaussiana, el método se describe de la siguiente manera:

1. **construcción de un sistema difuso inicial:** dados el par de datos de entrada y salida $(x^i, y^j) \in G$, donde $i = 1, \dots, M$, se forma un sistema difuso inicial con (x^1, y^1) y se inicializa el número de reglas $R = 1$

$$f(x^i | \theta) = \frac{b_1 \prod_{j=1}^n \exp\left(-\frac{1}{2} \left(\frac{x_j^i - c_j^1}{\sigma_j^1}\right)^2\right)}{\prod_{j=1}^n \exp\left(-\frac{1}{2} \left(\frac{x_j^i - c_j^1}{\sigma_j^1}\right)^2\right)} \quad i = 1, \dots, M \quad (3.21)$$

donde $c_j^1 = x_j^1$, $\sigma_j^1 = \sigma_0$ (parámetros de las funciones de pertenencia de las variables de entrada) y $b_1 = y^1$ (parámetro de la función de pertenencia de la variable de salida), para todo $j = 1, \dots, n$. Una vez que se inicializa el sistema difuso, el método forma reglas difusas y selecciona parámetros utilizando los Pasos 2 y 3.

2. **evaluación de los datos de entrenamiento:** se utiliza un factor de tolerancia

ε_f que sirve para determinar el error que se acepta en el cálculo. Para cada par de datos de entrenamiento $(x^i, y^i) \in G$ se evalúa

- si $|f(x^i | \theta) - y^i| \leq \varepsilon_f$, entonces el sistema difuso f representa correctamente (x^i, y^i) y por lo tanto, no es necesario añadir ninguna regla a f . Se repite este paso, con el siguiente par de datos de entrenamiento.
- si $|f(x^i | \theta) - y^i| > \varepsilon_f$, entonces se añade una regla para representar (x^i, y^i) sobre g modificando los parámetros de las funciones de pertenencia. Continuar al Paso 3

3. codificación de parámetros del sistema difuso: si no se cumple la condición del Paso 2, se añade una nueva regla $R = R + 1$, $b_R = y^i$ y $c_j^i = x_j^i$, para todo $j = 1, \dots, n$. La modificación de los σ_j^i para $i = R$ se la hace utilizando los centros de las funciones de pertenencia de la forma

$$n_j^* = \arg \min \{ |c_j^{i'} - c_j^i| : i' = 1, 2, \dots, R, i' \neq i \}$$

donde $j = 1, 2, \dots, n$ y c_j^i es fijo. Aquí n_j^* denota el índice i' de los $c_j^{i'}$ que minimiza la expresión. Y finalmente se actualiza los σ_j^i para $i = R$ colocando

$$\sigma_j^i = \frac{1}{\omega} |c_j^i - c_j^{n_j^*}| \quad j = 1, 2, \dots, n \quad (3.22)$$

donde ω es un factor que determina la sobreposición de las funciones de pertenencia. Se asume que para los datos de entrenamiento $(x^i, y^i) \in G$, $x_i^j \neq x_i^{j'}$ para cualquier $i' \neq j'$, para cada $j' \neq j'$ nunca se tendrá $\sigma_j^i = 0$. Esto implicaría que hay una función de pertenencia de entrada con dispersión cero; es decir, una función de pertenencia vacía. Se puede observar en (3.22) que el factor de ponderación ω y la dispersión σ_j^i tienen una relación inversa, es decir, pa-

ra un ω más grande se tiene una menor sobreposición. Una vez que el Paso 3 se ha completado, se repite el Paso 2, hasta que el número de datos de entrenamiento $(x^i, y^i) \in G$ se haya terminado.

Dado que el parámetro ε_f caracteriza la exactitud con la que opera el sistema difuso, la elección de ε_f determina el número de reglas, es decir, un valor muy pequeño de ε_f significa muchas reglas para un conjunto de datos y viceversa. En el Algoritmo 4 se detalla este método.

Observación 4. La parte más importante en la construcción de sistemas difusos es la elección del número de reglas difusas R , para lo cual es necesario el criterio de expertos, ver [20]. Algunas aplicaciones específicas limitan el número de reglas del sistema, ya sea por las características específicas del fenómeno o por su costo computacional. En general, existe una compensación entre la complejidad computacional y la capacidad funcional basada en el número de reglas difusas R .

Algoritmo 4 Aprendizaje desde el ejemplo modificado

Entrada: $X_{M \times n}$ matriz de las variables de entrada

Y_M matriz de la variable de salida

$\varepsilon_f > 0, \omega > 0$

1: $R \leftarrow 1$

2: $b_1 \leftarrow y^1$

3: $\hat{\theta} \leftarrow b$

4: $c_j^1 \leftarrow x_j^1 \quad j = 1, \dots, n$

5: **para** $i \leftarrow 1, M$ **hacer**

6: **para** $l \leftarrow 1, R$ **hacer**

7: $\mu_{il} \leftarrow \prod_{j=1}^n \exp \left[-\frac{1}{2} \left(\frac{x_j^i - c_j^l}{\sigma_j^l} \right)^2 \right]$

8: **fin para**

9: $\xi_{il} \leftarrow \frac{\mu_{il}}{R \sum_{l=1}^R \mu_{il}}$

10: $f^l(x) \leftarrow \hat{\theta}^\top \xi(x)$

11: **si** $|f^l(x) - y^l| > \varepsilon_f$ **entonces**

12: $R \leftarrow R + 1$

13: $b_R \leftarrow y^l$

14: $\hat{\theta} \leftarrow b$

15: $c_j^R \leftarrow x_j^l \quad j = 1, \dots, n$

16: $n_j^* \leftarrow \arg \min \{ |c_j^{l'} - c_j^l| : l' = 1, 2, \dots, R, l' \neq l \}$

17: $\sigma_j^l \leftarrow \frac{1}{\omega} |c_j^l - c_j^{n_j^*}|$

18: **fin si**

19: **fin para**

20: **devolver** $f(x | \hat{\theta}) \leftarrow \hat{\theta}^\top \xi(x)$

CAPÍTULO 4

MÉTODOS AUTOMÁTICOS DE REGRESIÓN DIFUSA COMBINADOS

En este capítulo, se plantea la formulación e implementación del método de mínimos cuadrados recursivo combinado [34], se generaliza los métodos al caso de entrada-salida múltiple, que son las principales contribuciones de este trabajo. Además se presenta el método de agrupamiento difuso combinado descrito por Kevin Passino y Stephen Yurkovich [15] y finalmente se valida el desempeño de todos los métodos descritos en este trabajo para series económicas concretas, como son:

- Producto Interno Bruto (PIB) no Petrolero de Ecuador,
- Reserva Internacional de Libre Disponibilidad (RILD) de Ecuador,
- Índice de Liquidez de Instituciones Financieras y
- Índices de Precios al Consumidor y al Productor de Ecuador.

4.1. Agrupamiento difuso combinado (ADC)

Los algoritmos de agrupamiento son procesos de aprendizaje no supervisado que tienen como objetivo descomponer un conjunto de elementos en grupos o clases, basados en la similitud, ver [25]. La *similitud* es entendida matemáticamente, en espacios métricos se define mediante distancias. Los métodos clásicos de

agrupamiento dividen los datos en un número c determinado de subconjuntos mutuamente excluyentes y se basan en la teoría clásica de conjuntos. Los métodos de agrupamiento difusos permiten que los objetos pertenezcan a diferentes clases simultáneamente, con diferentes grados de pertenencia, siendo c el número total de subconjuntos difusos definidos (clases o grupos). Además, esta metodología a probado ser muy eficiente cuando se trata de construir funciones de pertenencia para las variables de entrada de un sistema, como se podrá observar en el método de agrupamiento difuso combinado.

El método de agrupamiento difuso combinado se compone de dos etapas. En la primera, se identifican las reglas del sistema mediante el algoritmo c -medias difuso. El algoritmo particiona o divide los datos de entrada en conjuntos difusos agrupando datos similares en una misma clase; a diferencia de otros algoritmos de clasificación un objeto puede pertenecer a varias clases con algún grado de pertenencia, más detalles ver [21]. En la segunda etapa se aplica los mínimos cuadrados para caracterizar el conjunto de salida.

El agrupamiento difuso c -medias permite que los elementos de los grupos tengan diferentes grados de pertenencia. El algoritmo determina estos grados de pertenencia μ_{il} y los centros de clase \underline{v}^l de manera iterativa, minimizando la siguiente función objetivo:

$$J = \sum_{i=1}^M \sum_{l=1}^R (\mu_{il})^m |x^i - \underline{v}^l|^2, \quad (4.1)$$

donde: $m > 1$ es un parámetro que sirve para determinar la sobreposición entre las clases (también se lo conoce como factor de superposición), M es el número de pares de entrada-salida (x^i, y^i) en el conjunto de entrenamiento G , R es el número de reglas que se quiere calcular (clases o grupos), $\underline{v}^l = [v_1^l, v_2^l, \dots, v_n^l]^T$ $l = 1, \dots, R$ es el vector de centros de clases y μ_{il} $i = 1, \dots, M$ y $l = 1, \dots, R$ es el grado de

pertenencia de x^i en el l -ésimo grupo y $|x| = \sqrt{x^\top x}$, ver [22].

El sistema difuso considera que la salida es una función lineal de las entradas, i.e., sistema difuso Takagi-Sugeno, ver Sección 2.3, de la forma:

$$\text{Si } H^l \text{ entonces } g_l(x) = a_{l,0} + a_{l,1}x_1 + \cdots + a_{l,n}x_n, \quad (4.2)$$

donde: n es el número de entradas y H^l es un conjunto difuso de entrada dado por

$$H^l = \{(x, \mu_{H^l}(x)) : x \in \mathcal{X}_1 \times \cdots \times \mathcal{X}_n\}$$

aquí \mathcal{X}_i es el i -ésimo universo y $\mu_{H^l}(x)$ es la función de pertenencia asociada con H^l que representa la premisa para la regla l y $g_l(x) = \underline{a}_l^\top \hat{x}$, $\underline{a}_l^\top = [a_{l,0}, a_{l,1}x_1, \dots, a_{l,n}x_n]$ y $\hat{x} = [1, x^\top]^\top$, $l = 1, \dots, R$. El sistema difuso resultante es el promedio ponderado de las salidas $g_l(x)$ es decir:

$$f(x | \theta) = \frac{\sum_{l=1}^R g_l(x) \mu_{H^l}(x)}{\sum_{l=1}^R \mu_{H^l}(x)} \quad (4.3)$$

donde R es el número de reglas en la regla-base.

El algoritmo funciona de la siguiente manera. Primero se especifica el “factor de solapamiento” m , si $m > 1$ los puntos con menor grado de pertenencia, tiene menor influencia en el cálculo de un centro nuevo. Luego, se especifica el número de grupos (clases) R que se desea calcular, el cual es igual al número de reglas en la regla base y debe ser menor o igual al número de datos en el conjunto de entrenamiento G , es decir, $R \leq M$. Además se debe especificar un nivel de tolerancia

$\epsilon_c > 0$, este parámetro determina el error permitido en el cálculo para los centros de los grupos. Entonces se procede a inicializar los centros de los grupos \underline{v}_0^l . Para lo cual se define:

$$X_{\text{mín}} = \text{mín}\{x_j, j = 1, \dots, n\}$$

$$X_{\text{máx}} = \text{máx}\{x_j, j = 1, \dots, n\}$$

y luego

$$\underline{v}_{0j}^l = X_{\text{mín},j} + l \left(\frac{X_{\text{máx},j} - X_{\text{mín},j}}{R + 1} \right) \quad j = 1, \dots, n, \quad l = 1, \dots, R \quad (4.4)$$

A continuación, se calculan los nuevos centros \underline{v}^l utilizando los centros iniciales, de manera que la función objetivo (4.1) sea minimizada. Las condiciones para la minimización J están dadas por,

$$\underline{v}_{new}^l = \frac{\sum_{i=1}^M x^i (\mu_{il}^{new})^m}{\sum_{i=1}^M (\mu_{il}^{new})^m} \quad (4.5)$$

donde

$$\mu_{il}^{new} = \left[\sum_{k=1}^R \left(\frac{|x^i - \underline{v}_{old}^l|^2}{|x^i - \underline{v}_{old}^k|^2} \right)^{\frac{1}{m-1}} \right]^{-1} \quad (4.6)$$

para $i = 1, 2, \dots, M$ y $l = 1, 2, \dots, R$ y

$$\sum_{l=1}^R \mu_{il}^{new} = 1.$$

Luego, se compara la distancia entre los centros nuevos de los grupos \underline{v}_{new}^l y los

centros previos \underline{v}_{old}^l utilizando la condición

$$|\underline{v}_{new}^l - \underline{v}_{old}^l| < \epsilon_c \quad l = 1, 2, \dots, R \quad (4.7)$$

si se cumple (4.7) significa que los centros \underline{v}_{new}^l representan los datos, el algoritmo de agrupamiento termina y se procede con la decodificación de la salida. Si no, se continua iterando usando las ecuaciones (4.5) y (4.6) hasta que los centros nuevos satisfagan (4.7).

Para la decodificación de la salida, se calcula la función $g_l(x) = \underline{a}_l^\top(\hat{x})$, $l = 1, 2, \dots, R$ para cada regla, es decir para cada centro de grupo \underline{v}_l , se minimiza el cuadrado del error entre la función $g_l(x)$ y parte de la salida, concretamente se minimiza la función J_l

$$J_l = \sum_{i=1}^M (\mu_{il})^2 (y^i - (\hat{x}^i)^\top \underline{a}_l)^2 \quad (4.8)$$

donde, $l = 1, 2, \dots, R$, y^i es la parte de la salida del i -ésimo par de datos $\vec{d}^{(i)} = (x^i, y^i)$ y el producto entre $(\hat{x}^i)^\top$ y \underline{a}_l define la salida asociada con la l -ésima regla para el i -ésimo punto de datos de entrenamiento, ver [21].

Al observar (4.8), se puede notar que la minimización de J_l respecto a \underline{a}_l corresponde a un problema de mínimos cuadrados ponderados, cuya solución es

$$\underline{a}_l = (\hat{X}^\top D_l^2 \hat{X})^{-1} \hat{X}^\top D_l^2 Y,$$

donde,

$$\hat{X} = \begin{bmatrix} 1 & \dots & 1 \\ x^1 & \dots & x^M \end{bmatrix}$$

$$Y = [y^1, \dots, y^m]^\top$$

$$D_i^2 = (\text{diag}[\mu_{1i}, \dots, \mu_{Mi}])^2$$

El método se describe en el Algoritmo 5.

Algoritmo 5 Agrupamiento difuso combinado

Entrada: $X_{M \times n}$ matriz de las variables de entrada

Y_M matriz de la variable de salida

R número de reglas (clases o grupos)

$\epsilon_c > 0, m > 1$

1: $X_{\min} \leftarrow \min\{x_j, j = 1, \dots, n\}$

2: $X_{\max} \leftarrow \max\{x_j, j = 1, \dots, n\}$

3: $\underline{v}_{0j}^l \leftarrow X_{\min,j} + l \left(\frac{X_{\max,j} - X_{\min,j}}{R + 1} \right) \quad j = 1, \dots, n, l = 1, \dots, R$

4: $\underline{v}_{new}^l \leftarrow \underline{v}_0^l \quad l = 1, \dots, R$

5: **mientras** $|\underline{v}_{new}^l - \underline{v}_{old}^l| > \epsilon_c \quad i \leftarrow 1, \dots, M, l \leftarrow 1, \dots, R$ **hacer**

6: $\underline{\mu}_{il}^{new} \leftarrow \left[\sum_{k=1}^R \left(\frac{|x^i - \underline{v}_{old}^l|^2}{|x^i - \underline{v}_{old}^k|^2} \right)^{\frac{1}{m-1}} \right]^{-1}$

7: $\underline{v}_{new}^l \leftarrow \frac{\sum_{i=1}^M x^i (\underline{\mu}_{il}^{new})^m}{\sum_{i=1}^M (\underline{\mu}_{il}^{new})^m}$

8: **fin mientras**

9: **para** $i \leftarrow 1, M$ **hacer**

10: $\hat{x}^i \leftarrow [1, x^i]$

11: **fin para**

12: $\hat{X} \leftarrow \hat{x}^\top$

13: $Y \leftarrow [y^1, \dots, y^m]^\top$

14: $D_i^2 \leftarrow (\text{diag}[\mu_{1i}, \dots, \mu_{Mi}])^2$

15: **para** $l \leftarrow 1, R$ **hacer**

16: $\underline{a}_l \leftarrow (\hat{X}^\top D_l^2 \hat{X})^{-1} \hat{X}^\top D_l^2 Y$

17: **devolver** $g_l(x) \leftarrow \underline{a}_l^\top \hat{x}$

18: **fin para**

4.2. Mínimos cuadrados recursivo combinado (MCRC)

Hasta ahora se han descrito brevemente los métodos automáticos de regresión difusa que se encuentran en la literatura, ver [15, 16]. En esta sección, se propone el método de mínimos cuadrados recursivo combinado, lo cual constituye el principal aporte teórico de este trabajo. Este método se compone de dos etapas. Primeramente, se utiliza el algoritmo c-medias difuso para la identificación de las reglas del sistema. En la segunda etapa, se aplica el método de mínimos cuadrados recursivo para caracterizar el conjunto de salida. Se considera esta combinación como la más adecuada, debido a que los resultados obtenidos tienen una mayor precisión y además requieren un bajo nivel computacional para su cálculo, como se verá en la Sección 4.4.

El método de mínimos cuadrados recursivo, es un método que necesita que la regla-base del sistema esté completamente especificada, i.e., número de reglas, centros de funciones de pertenencia, etc. Puesto que las reglas del sistema en los métodos automáticos se encargan de predecir y/o gobernar las salidas de los sistemas difusos, es necesario un algoritmo que permita una adecuada especificación de reglas. Es aquí donde los algoritmos de agrupamiento difuso han probado ser muy eficientes tal como quedó probado para el caso del método de agrupamiento difuso combinado. Por esta razón, para mejorar la interpretabilidad de las reglas del sistema, utilizaremos en el método propuesto el algoritmo c-medias difuso.

Una vez que se tiene especificada las reglas del sistema, se procede a la aplicación del método de mínimos cuadrados recursivo. Este método posee características muy útiles para ser implementado:

- no requiere la inversión de una matriz, como en el caso de los mínimos cuadrados por lotes y agrupamiento difuso combinado; en su lugar, se calcula únicamente el inverso de un escalar,
- los mínimos cuadrados recursivos son más eficientes al incluir las reglas del sistema, ver [15],
- la convergencia del método es competitiva respecto a otras técnicas, ver [26].

El método de mínimos cuadrados recursivo combinado se describe a continuación:

1. determinar el número de centros R , e inicializarlos

$$R = c,$$

$$\underline{v}_{new}^l = \underline{v}_0^l \quad l = 1, \dots, R$$

para lo cual se aplica la ecuación (4.4) de la Sección 4.1

2. calcular una partición de los datos, asignando cada punto al grupo más cercano utilizando la función de pertenencia

$$\mu_{il}^{new} = \left[\sum_{k=1}^R \left(\frac{|x^i - \underline{v}_{old}^l|^2}{|x^i - \underline{v}_{old}^k|^2} \right)^{\frac{1}{m-1}} \right]^{-1},$$

3. calcular los nuevos centros de los grupos

$$\underline{v}_{new}^l = \frac{\sum_{i=1}^M x^i (\mu_{il}^{new})^m}{\sum_{i=1}^M (\mu_{il}^{new})^m},$$

4. si $|v_{new}^l - v_{old}^l| \leq \epsilon_c$ para $i = 1, \dots, M$, $l = 1, \dots, R$ la partición es estable, es decir el proceso se detiene. Si no se regresa al paso 2.
5. se inicializa los parámetros del método de mínimos cuadrados recursivo (MCR), $\hat{\theta}(0) = 0$, $P(0) = \alpha I$
6. se calcula los grados de pertenencia utilizando los centros, $i = 1, \dots, M$ y $l = 1, \dots, R$

$$\mu_{il} = \prod_{j=1}^n \exp \left[-\frac{1}{2} \left(\frac{x_j^i - v_j^l}{\sigma_j^l} \right)^2 \right]$$

$$\xi_{il} = \frac{\mu_{il}}{\sum_{l=1}^R \mu_{il}}$$

7. se aplica el MCR, ver Algoritmo 3

$$P(k) = \frac{1}{\lambda} \left\{ I - P(k-1) \xi(x^k) [\lambda I + (\xi(x^k))^\top P(k-1) \xi(x^k)]^{-1} (\xi(x^k))^\top \right\} P(k-1)$$

$$\hat{\theta}(k) = \hat{\theta}(k-1) + P(k) \xi(x^k) \left[y^k - (\xi(x^k))^\top \hat{\theta}(k-1) \right]$$

Este método se esquematiza en el Algoritmo 6. El cual utiliza el Algoritmo 3, ya que esta es una versión más eficiente del Algoritmo 2.

Observación 5. Existen algunas consideraciones que se deben tener en cuenta al aplicar el algoritmo c-medias difuso. Primero, se debe inicializar del número de reglas R (grupos o clases) y los valores de los centros de las clases, ya que los resultados son sensibles a la elección de estos valores iniciales.

Observación 6. La función objetivo (4.8) influye directamente en el desempeño del Algoritmo 6, por esta razón debe ser resuelta eficientemente si no se toman en cuenta los valores atípicos, ver [23].

Algoritmo 6 Mínimos cuadrados recursivo combinado

Entrada: $X_{M \times n}$ matriz de las variables de entrada

Y_M matriz de la variable de salida

K número de iteraciones del algoritmo

R número de reglas (clases o grupos)

$\epsilon_c > 0, m > 1, 0 < \lambda \leq 1$

1: $X_{\min} \leftarrow \min\{x_j, j = 1, \dots, n\}$

2: $X_{\max} \leftarrow \max\{x_j, j = 1, \dots, n\}$

3: $\underline{v}_{0j}^l \leftarrow X_{\min,j} + l \left(\frac{X_{\max,j} - X_{\min,j}}{R + 1} \right) \quad j = 1, \dots, n, l = 1, \dots, R$

4: $\underline{v}_{new}^l \leftarrow \underline{v}_0^l \quad l \leftarrow 1, \dots, R$

5: **mientras** $|\underline{v}_{new}^l - \underline{v}_{old}^l| > \epsilon_c \quad i \leftarrow 1, \dots, M, l \leftarrow 1, \dots, R$ **hacer**

6: $\underline{\mu}_{il}^{new} \leftarrow \left[\sum_{k=1}^R \left(\frac{|x^i - \underline{v}_{old}^k|^2}{|x^i - \underline{v}_{old}^k|^2} \right)^{\frac{1}{m-1}} \right]^{-1}$

7: $\underline{v}_{new}^l \leftarrow \frac{\sum_{i=1}^M x^i (\underline{\mu}_{il}^{new})^m}{\sum_{i=1}^M (\underline{\mu}_{il}^{new})^m}$

8: **fin mientras**

9: $\hat{\theta}(0) \leftarrow 0$

10: $P(0) \leftarrow \alpha I$

11: $\sigma_j^l \leftarrow \sigma \quad j = 1, \dots, n$

12: **para** $i \leftarrow 1, M$ **hacer**

13: **para** $l \leftarrow 1, R$ **hacer**

14: $\mu_{il} \leftarrow \prod_{j=1}^n \exp \left[-\frac{1}{2} \left(\frac{x_j^i - \underline{v}_j^l}{\sigma_j^l} \right)^2 \right]$

15: **fin para**

16: $\xi_{il} \leftarrow \frac{\mu_{il}}{R}$
 $\sum_{l=1}^R \mu_{il}$

17: **fin para**

18: $s \leftarrow 0$ contador auxiliar

19: **para** $k \leftarrow 1, K \times M$ **hacer**

20: **si** $s \leq M$ **entonces**

21: $s \leftarrow s + 1$

22: **si no**

23: $s \leftarrow 1$

24: **fin si**

25: $P(k) \leftarrow \frac{1}{\lambda} \left\{ I - P(k-1) \xi(x^s) [\lambda I + (\xi(x^s))^\top P(k-1) \xi(x^s)]^{-1} (\xi(x^s))^\top \right\} P(k-1)$

26: $\hat{\theta}(k) \leftarrow \hat{\theta}(k-1) + P(k) \xi(x^s) \left[y^s - (\xi(x^s))^\top \hat{\theta}(k-1) \right]$

27: **fin para**

28: **devolver** $f(x | \hat{\theta}) \leftarrow \hat{\theta}^\top \xi(x)$

4.3. Generalización al caso de entrada-salida múltiple

La extensión de los métodos al caso de entrada y salida múltiple se realizó bajo dos premisas fundamentales:

- las variables dependientes deben ser no correlacionadas entre sí,
- optimizar el tiempo de respuesta y los recursos computacionales.

4.3.1. Mínimos cuadrados por lotes entrada-salida múltiple

En este método, existen dos bloques principales de cálculo. El primero corresponde al cálculo de los valores de pertenencia μ_{il} y el segundo al del vector $\hat{\theta}$. Dado que los valores de los centros y dispersiones de las funciones de pertenencia de las variables de entrada se mantienen constantes, así como los valores de pertenencia para las variables de entrada, por lo tanto, al generalizar el método varían los vectores y_h como se muestra en la línea 11 del Algoritmo 7. También es necesario crear de una matriz de parámetros $\hat{\theta}_h$ para $h = 1, \dots, H$, tal como se describe en el Algoritmo 7.

4.3.2. Mínimos cuadrados recursivo entrada-salida múltiple

Para la generalización al caso múltiple, se ha considerado la versión modificada del método de mínimos cuadrados recursivo (Algoritmo 3). Al igual que en el caso anterior, se observan dos bloques de cálculo, en el primero se calcula los valores de pertenencia μ_{il} y en el segundo los parámetros $P(k)$ y $\hat{\theta}(k)$. Lo cual implica variar y_h^s e incluir un lazo en la línea 11, ver Algoritmo 8.

Algoritmo 7 Mínimos cuadrados por lotes entrada-salida múltiple

Entrada: $X_{M \times n}$ matriz de las variables de entrada

$Y_{M \times H}$ matriz de las variables de salida

$\sigma > 0$

1: $R \leftarrow M - 1$

2: $c_j^l \leftarrow x_j^l + \frac{1}{2}(x_j^{l+1} - x_j^l) \quad l = 1, \dots, R, j = 1, \dots, n$

3: $\sigma_j^l \leftarrow \sigma \quad l = 1, \dots, R, j = 1, \dots, n$

4: **para** $i \leftarrow 1, M$ **hacer**

5: **para** $l \leftarrow 1, R$ **hacer**

6: $\mu_{il} \leftarrow \prod_{j=1}^n \exp \left[-\frac{1}{2} \left(\frac{x_j^i - c_j^l}{\sigma_j^l} \right)^2 \right]$

7: **fin para**

8: $\xi_{il} \leftarrow \frac{\mu_{il}}{\sum_{l=1}^R \mu_{il}}$

9: **fin para**

10: $\Phi \leftarrow \xi(x)$

11: **para** $h \leftarrow 1, H$ **hacer**

12: $\hat{\theta}_h \leftarrow (\Phi^\top \Phi)^{-1} \Phi^\top y_h$

13: **devolver** $f_h(x | \hat{\theta}_h) \leftarrow \hat{\theta}_h^\top \xi(x)$

14: **fin para**

Algoritmo 8 Mínimos cuadrados recursivo entrada-salida múltiple

Entrada: $X_{M \times n}$ matriz de las variables de entrada

$Y_{M \times H}$ matriz de las variables de salida

K número de iteraciones del algoritmo

$\sigma > 0, \alpha > 0, 0 < \lambda \leq 1$

1: $\hat{\theta}(0) \leftarrow 0$

2: $P(0) \leftarrow \alpha I$

3: $c_j^l \leftarrow x_j^l + \frac{1}{2}(x_j^{l+1} - x_j^l) \quad l = 1, \dots, R, j = 1, \dots, n$

4: $\sigma_j^l \leftarrow \sigma \quad l = 1, \dots, R, j = 1, \dots, n$

5: **para** $i \leftarrow 1, M$ **hacer**

6: **para** $l \leftarrow 1, R$ **hacer**

7: $\mu_{il} \leftarrow \prod_{j=1}^n \exp \left[-\frac{1}{2} \left(\frac{x_j^i - c_j^l}{\sigma_j^l} \right)^2 \right]$

8: **fin para**

9: $\xi_{il} \leftarrow \frac{\mu_{il}}{\sum_{l=1}^R \mu_{il}}$

10: **fin para**

11: **para** $h \leftarrow 1, H$ **hacer**

12: $s \leftarrow 0$ contador auxiliar

13: **para** $k \leftarrow 1, K \times M$ **hacer**

14: **si** $s \leq M$ **entonces**

15: $s \leftarrow s + 1$

16: **si no**

17: $s \leftarrow 1$

18: **fin si**

19: $P_h(k) \leftarrow \frac{1}{\lambda} \left\{ I - P_h(k-1) \xi(x^s) [\lambda I + (\xi(x^s))^\top P_h(k-1) \xi(x^s)]^{-1} (\xi(x^s))^\top \right\} P_h(k-1)$

20: $\hat{\theta}_h(k) \leftarrow \hat{\theta}_h(k-1) + P_h(k) \xi(x^s) \left[y_h^s - (\xi(x^s))^\top \hat{\theta}_h(k-1) \right]$

21: **fin para**

22: **devolver** $f_h(x | \hat{\theta}_h) \leftarrow \hat{\theta}_h^\top \xi(x)$

23: **fin para**

4.3.3. Aprendizaje desde el ejemplo modificado entrada-salida múltiple

A diferencia de los algoritmos anteriores, este método no puede ser separado por bloques de cálculo debido a que el condicional de la línea 12 modifica los valores de los centros c_j^l y las dispersiones σ_j^l de las funciones de pertenencia de las variables entrada, lo cual hace que varíen estos valores para cada $y_h, h = 1, \dots, H$. La generalización se obtiene incluyendo un nuevo lazo que contenga todo el cálculo, línea 5 del Algoritmo 9. En este método, es necesario al aplicación de matrices dinámicas para el parámetro $\hat{\theta}_h$.

4.3.4. Agrupamiento difuso combinado entrada-salida múltiple

Se puede observar que este método tiene dos bloques. Primero se calculan los centros difusos, del Algoritmo 10 línea 5 y el luego se ejecuta la decodificación difusa, línea 16. Para su generalización, se modifica la línea 17, haciendo variar Y_h y se incluye el lazo de la línea 15, ver Algoritmo 10.

4.3.5. Mínimos cuadrados recursivo combinado entrada-salida múltiple

Este algoritmo está compuesto por tres bloques de cálculo. El primero, que se encarga de calcular los centros difusos, línea 5. El segundo bloque se determinan los valores de pertenencia μ_{il} . Y finalmente, se calcula los parámetros $P(k)$ y $\hat{\theta}(k)$. La generalización, se obtiene modificando las líneas 26 y 27, y se incluye un nuevo lazo en la línea 18, ver Algoritmo 11.

Algoritmo 9 Aprendizaje desde el ejemplo modificado entrada-salida múltiple

Entrada: $X_{M \times n}$ matriz de las variables de entrada

$Y_{M \times H}$ matriz de las variables de salida

$\varepsilon_f > 0, \omega > 0$

```
1:  $R \leftarrow 1$ 
2:  $b_1 \leftarrow y^1$ 
3:  $\hat{\theta} \leftarrow b$ 
4:  $c_j^1 \leftarrow x_j^1 \quad j = 1, \dots, n$ 
5: para  $h \leftarrow 1, H$  hacer
6:   para  $i \leftarrow 1, M$  hacer
7:     para  $l \leftarrow 1, R$  hacer
8:       
$$\mu_{il} \leftarrow \prod_{j=1}^n \exp \left[ -\frac{1}{2} \left( \frac{x_j^i - c_j^l}{\sigma_j^l} \right)^2 \right]$$

9:     fin para
10:    
$$\xi_{il} \leftarrow \frac{\mu_{il}}{R \sum_{l=1}^R \mu_{il}}$$

11:     $f^l(x) \leftarrow \hat{\theta}^\top \xi(x)$ 
12:    si  $|f_h^l(x) - y_h^l| > \varepsilon_f$  entonces
13:       $R \leftarrow R + 1$ 
14:       $b_R \leftarrow y_h^l$ 
15:       $\hat{\theta}_h \leftarrow b$ 
16:       $c_j^R \leftarrow x_j^l \quad j = 1, \dots, n$ 
17:       $n_j^* \leftarrow \arg \min \{ |c_j^{l'} - c_j^l| : l' = 1, 2, \dots, R, l' \neq l \}$ 
18:       $\sigma_j^l \leftarrow \frac{1}{\omega} |c_j^l - c_j^{n_j^*}|$ 
19:    fin si
20:  fin para
21:  devolver  $f_h(x | \hat{\theta}_h) \leftarrow \hat{\theta}_h^\top \xi_h(x)$ 
22: fin para
```

Algoritmo 10 Agrupamiento difuso combinado entrada-salida múltiple

Entrada: $X_{M \times n}$ matriz de las variables de entrada

$Y_{M \times H}$ matriz de las variables de salida

R número de reglas (clases o grupos)

$\epsilon_c > 0, m > 1$

1: $X_{\min} \leftarrow \min\{x_j, j = 1, \dots, n\}$

2: $X_{\max} \leftarrow \max\{x_j, j = 1, \dots, n\}$

3: $\underline{v}_{0j}^l \leftarrow X_{\min,j} + l \left(\frac{X_{\max,j} - X_{\min,j}}{R + 1} \right) \quad j = 1, \dots, n, l = 1, \dots, R$

4: $\underline{v}_{new}^l \leftarrow \underline{v}_0^l \quad l = 1, \dots, R$

5: **mientras** $|\underline{v}_{new}^l - \underline{v}_{old}^l| > \epsilon_c \quad i \leftarrow 1, \dots, M, l \leftarrow 1, \dots, R$ **hacer**

6: $\underline{\mu}_{il}^{new} \leftarrow \left[\sum_{k=1}^R \left(\frac{|x^i - \underline{v}_{old}^l|^2}{|x^i - \underline{v}_{old}^k|^2} \right)^{\frac{1}{m-1}} \right]^{-1}$

7: $\underline{v}_{new}^l \leftarrow \frac{\sum_{i=1}^M x^i (\underline{\mu}_{il}^{new})^m}{\sum_{i=1}^M (\underline{\mu}_{il}^{new})^m}$

8: **fin mientras**

9: **para** $i \leftarrow 1, M$ **hacer**

10: $\hat{x}^i \leftarrow [1, x^i]$

11: **fin para**

12: $\hat{X} \leftarrow \hat{x}^\top$

13: $Y \leftarrow [y^1, \dots, y^m]^\top$

14: $D_i^2 \leftarrow (diag[\mu_{i1}, \dots, \mu_{iM}])^2$

15: **para** $h \leftarrow 1, H$ **hacer**

16: **para** $l \leftarrow 1, R$ **hacer**

17: $\underline{a}_{lh} \leftarrow (\hat{X}^\top D_l^2 \hat{X})^{-1} \hat{X}^\top D_l^2 Y_h$

18: **devolver** $g_{lh}(x) \leftarrow \underline{a}_{lh}^\top \hat{x}$

19: **fin para**

20: **fin para**

Algoritmo 11 Mínimos cuadrados recursivo combinado entrada-salida múltiple

Entrada: $X_{M \times n}$ matriz las variables de entrada

$Y_{M \times H}$ matriz de las variables de salida

K número de iteraciones del algoritmo

R número de reglas (clases o grupos)

$\epsilon_c > 0, m > 1, 0 < \lambda \leq 1$

1: $X_{\min} \leftarrow \min\{x_j, j = 1, \dots, n\}$

2: $X_{\max} \leftarrow \max\{x_j, j = 1, \dots, n\}$

3: $\underline{v}_{0j}^l \leftarrow X_{\min,j} + l \left(\frac{X_{\max,j} - X_{\min,j}}{R + 1} \right) \quad j = 1, \dots, n, l = 1, \dots, R$

4: $\underline{v}_{new}^l \leftarrow \underline{v}_0^l \quad l \leftarrow 1, \dots, R$

5: **mientras** $|\underline{v}_{new}^l - \underline{v}_{old}^l| > \epsilon_c \quad i \leftarrow 1, \dots, M, l \leftarrow 1, \dots, R$ **hacer**

6: $\underline{\mu}_{il}^{new} \leftarrow \left[\sum_{k=1}^R \left(\frac{|x^i - \underline{v}_{old}^k|^2}{|x^i - \underline{v}_{old}^k|^2} \right)^{\frac{1}{m-1}} \right]^{-1}$

7: $\underline{v}_{new}^l \leftarrow \frac{\sum_{i=1}^M x^i (\underline{\mu}_{il}^{new})^m}{\sum_{i=1}^M (\underline{\mu}_{il}^{new})^m}$

8: **fin mientras**

9: $\hat{\theta}(0) \leftarrow 0$

10: $P(0) \leftarrow \alpha I$

11: $\sigma_j^l \leftarrow \sigma \quad j = 1, \dots, n$

12: **para** $i \leftarrow 1, M$ **hacer**

13: **para** $l \leftarrow 1, R$ **hacer**

14: $\mu_{il} \leftarrow \prod_{j=1}^n \exp \left[-\frac{1}{2} \left(\frac{x_j^i - \underline{v}_j^l}{\sigma_j^l} \right)^2 \right]$

15: **fin para**

16: $\xi_{il} \leftarrow \frac{\mu_{il}}{\sum_{l=1}^R \mu_{il}}$

17: **fin para**

18: **para** $h \leftarrow 1, H$ **hacer**

19: $s \leftarrow 0$ contador auxiliar

20: **para** $k \leftarrow 1, K \times M$ **hacer**

21: **si** $s \leq M$ **entonces**

22: $s \leftarrow s + 1$

23: **si no**

24: $s \leftarrow 1$

25: **fin si**

26: $P_h(k) \leftarrow \frac{1}{\lambda} \left\{ I - P_h(k-1) \xi(x^s) [\lambda I + (\xi(x^s))^\top P_h(k-1) \xi(x^s)]^{-1} (\xi(x^s))^\top \right\} P_h(k-1)$

27: $\hat{\theta}_h(k) \leftarrow \hat{\theta}_h(k-1) + P_h(k) \xi(x^s) \left[y_h^s - (\xi(x^s))^\top \hat{\theta}_h(k-1) \right]$

28: **fin para**

29: **devolver** $f_h(x | \hat{\theta}_h) \leftarrow \hat{\theta}_h^\top \xi(x)$

30: **fin para**

4.4. Aplicación a modelos económicos

En esta sección se presentan resultados de pruebas numéricas realizadas con información del Producto Interno Bruto (PIB) no petrolero de Ecuador¹, la Reserva Internacional de libre Disponibilidad (RILD) de Ecuador², un índice de liquidez financiera³ y los Índices de Precios al Consumidor (IPC) y Productor (IPP) de Ecuador⁴, con el objetivo de validar los métodos propuestos y analizar su comportamiento.

Los resultados para cada una de las aplicaciones numéricas fueron obtenidos mediante la aplicación desarrollada (ver [35]). Para ello se implementaron los métodos:

- mínimos cuadrados por lotes (MCL),
- mínimos cuadrados recursivo (MCR),
- aprendizaje desde el ejemplo modificado (AEM),
- agrupamiento difuso combinado (ADC),
- mínimos cuadrados recursivo combinado (MCRC) (método propuesto).

Estos métodos fueron implementados utilizando funciones de pertenencia Gausianas y Triangulares. Únicamente en el caso del agrupamiento difuso combinado no se implementó estas funciones, debido a que este método define específicamente su función de pertenencia.

¹Los datos son obtenidos de los Boletines del Banco Central del Ecuador.

²Esta información proviene de la Dirección de Estudios del Banco Central de Ecuador.

³Esta información corresponde a los Boletines Financieros publicados mensualmente por la Superintendencia de Bancos y Seguros.

⁴Los datos son obtenidos de los Boletines del Instituto Nacional de Estadísticas y Censos.

Para visualizar de mejor manera el desempeño de cada método, se estandarizaron los datos, es decir, se utilizó una escala porcentual o en su defecto se normalizaron y estandarizaron las variables, ver [27].

En algunas aplicaciones se dividió al conjunto de datos en conjunto de entrenamiento y un conjunto de pruebas (*back testing*). Concretamente, en el conjunto de entrenamiento se aplican los métodos, después se calculan las predicciones y se evalúa la capacidad predictiva de cada método. Además, para contrastar los resultados se calcularon modelos de series de tiempo (SARIMA) para las aplicaciones de la RILD, el índice de liquidez y los índices de precios al consumidor-productor. Estos cálculos se los hizo utilizando el paquete estadístico *forecast* y la función *auto.arima* del software *R*, ver [28].

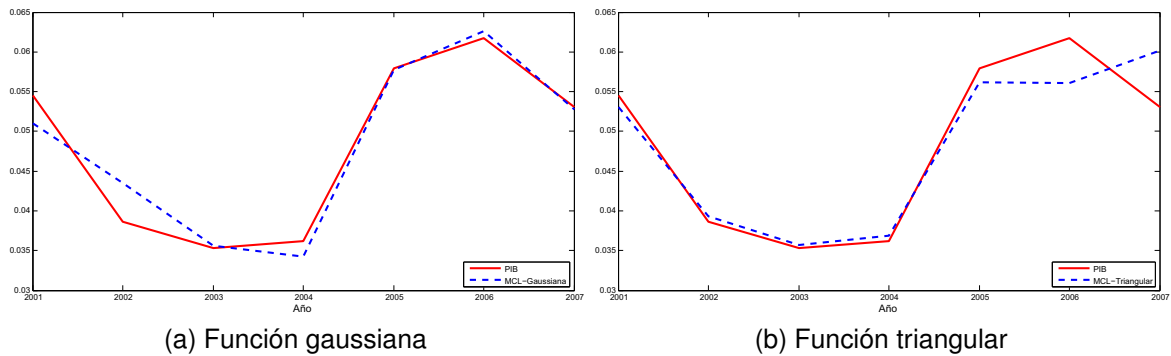


Figura 4.1: PIB mínimos cuadrados por lotes (MCL)

4.4.1. Producto Interno Bruto (PIB) no Petrolero de Ecuador

El PIB no petrolero es el valor de los bienes y servicios de uso final generados por agentes económicos durante un período. Su cálculo se deriva de la construcción de la matriz Insumo-Producto, que describe los flujos de bienes y servicios en el aparato productivo, desde la óptica de los productores y de los usuarios finales.

Las variables que se consideraron para este modelo son las siguientes:

Variable	Descripción
Y	Producto interno bruto no petrolero de Ecuador
X_1	Materia Prima
X_2	Inflación anual
X_3	Formación bruta de capital fijo (FBKF)
X_4	Consumo de hogares

Las series anuales corresponden al período 2001 - 2007. En esta aplicación no se hace *back testing* debido a poca información.

En la Figura 4.1 para el caso de la función de pertenencia Gaussiana el ajuste es mejor, pues esta función describe de mejor manera la tendencia del PIB, comparado con el ajuste de la función Triangular.

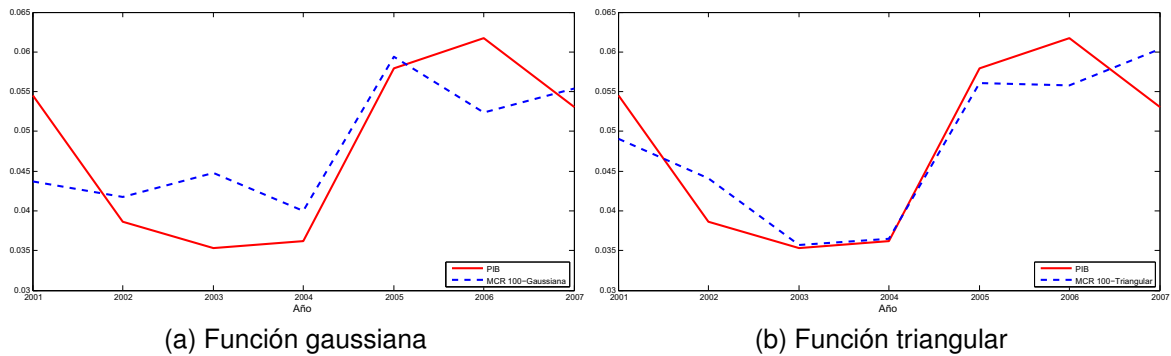


Figura 4.2: PIB mínimos cuadrados recursivo 100 iteraciones (MCR)

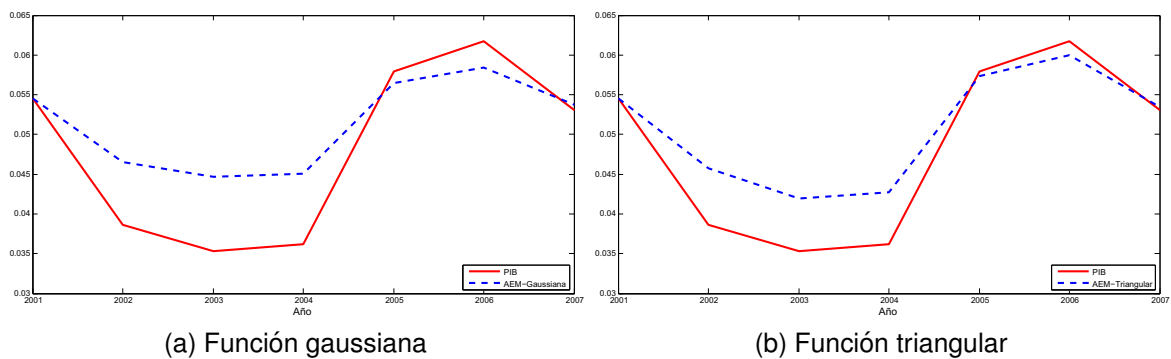


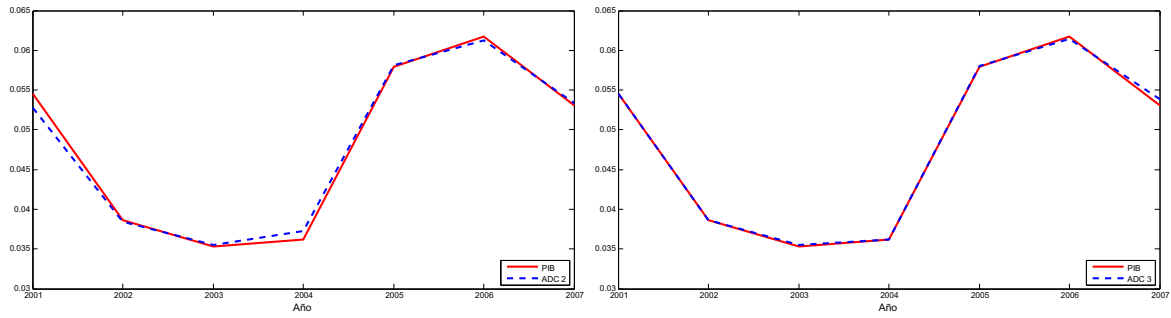
Figura 4.3: PIB aprendizaje desde el ejemplo modificado (AEM)

La Figura 4.2 presenta el ajuste del método de mínimos cuadrados recursivo, aquí se utilizaron 100 iteraciones, para ambas funciones de pertenencia, se aprecia que en ambos casos, las estimaciones no se asemejan a los valores reales de la serie.

En el caso del aprendizaje desde ejemplo modificado, ambas funciones describen eficientemente la tendencia de la serie, pero la función de pertenencia Triangular ajusta de mejor manera que la función Gaussiana, ver Figura 4.3.

La Figura 4.4 muestra el desempeño del agrupamiento difuso combinado con 2 y 3 centros, aquí el ajuste con 3 centros es superior al ajuste con 2 centros, la serie pronosticada se sobrepone a la serie del PIB.

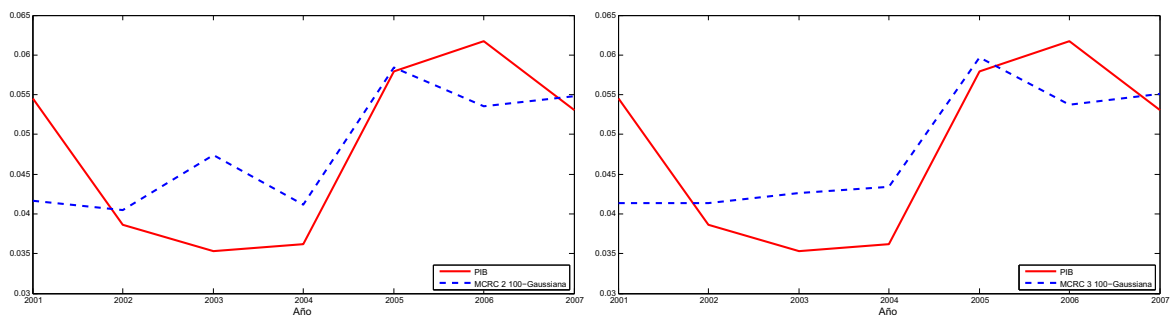
El método de mínimos cuadrados recursivo combinado se aprecia en la Figura



(a) Centros $c = 2$

(b) Centros $c = 3$

Figura 4.4: PIB agrupamiento difuso combinado (ADC)



(a) Centros $c = 2$

(b) Centros $c = 3$

Figura 4.5: PIB mínimos cuadrados recursivo combinado 100 iteraciones (MCRC)

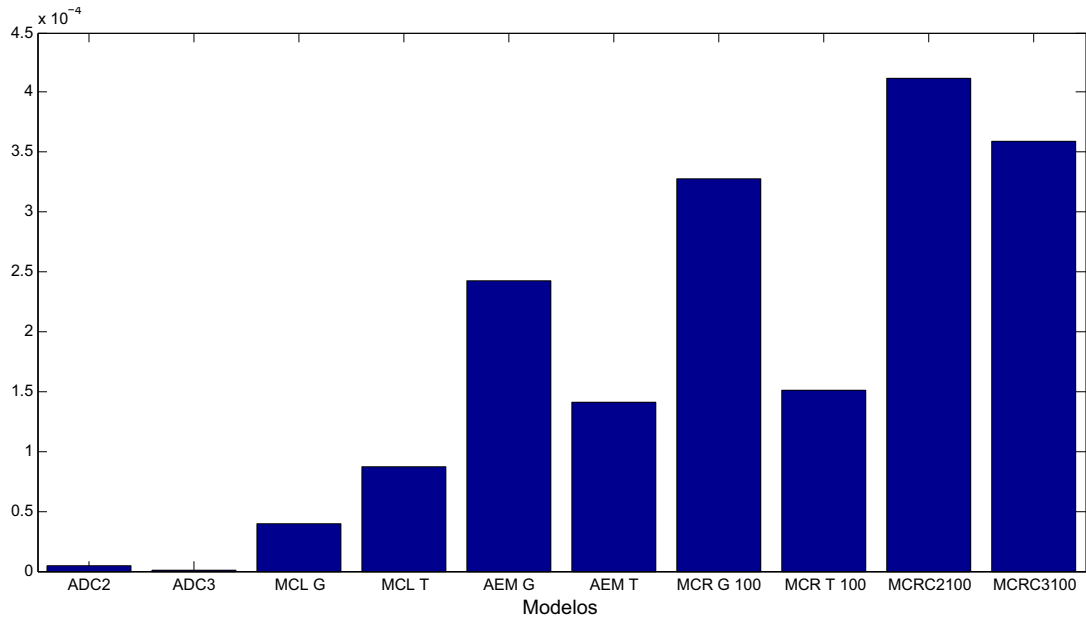


Figura 4.6: PIB residuos de los métodos automáticos

4.5, se aplicaron 100 iteraciones, con 2 y 3 centros, en este caso se observa que para ambos centros los pronósticos no se ajustan a la serie original, aun cuando se incrementa el número de centros.

La Figura 4.6 presenta la suma de errores para cada uno de los métodos. Los métodos de agrupamiento difuso combinado tienen el menor error acumulado, le sigue los métodos de mínimos cuadrados por lotes, el aprendizaje desde ejemplo modificado con función triangular y finalmente el método de mínimos cuadrados recursivo con función triangular. Se concluye que en este caso los métodos de agrupamiento difuso combinado dan mejores resultados ya que estos tienen un ajuste superior comparado con los demás métodos. Es importante notar que para esta aplicación hay solamente 7 observaciones en las series. En este contexto el ajuste obtenido para los métodos combinados es eficiente.

4.4.2. Reserva Internacional de Libre Disponibilidad (RILD) de Ecuador

La administración de las reservas internacionales requiere objetivos claros, sistemas de control fuertes y una apreciación realista de las limitaciones que se tienen. Un manejo eficiente de estas reservas implica una administración macroeconómica exitosa para un país. La mayor parte de los bancos centrales tienden a mantener reservas en forma de bonos; los cuales, devengan intereses de bajo riesgo u otros activos de riesgo similar en algún tipo de divisa. Lo anterior constituye un intento de asegurar los valores de mercado de sus activos externos, que son susceptibles de tener cambios, de manera que sean siempre mayores que el valor de los pasivos monetarios.

Es de gran importancia para un país asegurar y fortalecer su capacidad de afrontar acontecimientos inesperados, como son los choques financieros exógenos. Estrategias sanas de manejo de reservas conllevan una amplia gama de objetivos de política, tales como:

- asegurar un alto nivel de confianza en las políticas monetaria y cambiaria de la economía, particularmente en regímenes de tipo de cambio fijo;
- mantener divisas líquidas durante los ataques de los choques externos;
- darle confianza a la comunidad internacional de que la economía es capaz de cumplir con sus obligaciones externas.

Particularmente, los depósitos que realizan, tanto el sector público como el sistema financiero privado, en el Banco Central del Ecuador, por ser exigibles a la vista, están respaldados por la Reserva Internacional de Libre Disponibilidad (RILD), la cual por su carácter de activo de reserva del país y por su tamaño, es el componente de mayor importancia en los activos del Banco Central del Ecuador.

Los recursos de la RILD son invertidos, por mandato legal, en los mercados financieros internacionales, bajo los principios de seguridad, liquidez y rentabilidad. Una parte de los rendimientos obtenidos se revierte al Presupuesto General del Estado y otra parte es transferida mensualmente al Instituto Ecuatoriano de Seguridad Social (IESS), para mejorar las prestaciones a favor de sus afiliados. Básicamente, la RILD corresponde al monto de activos externos de alta liquidez que tienen una contrapartida en los pasivos del balance del Banco Central del Ecuador ⁵.

El Banco Central del Ecuador, calcula la reserva monetaria de libre disponibilidad que sirve de base para mantener los sistemas de canje, reserva financiera y otras operaciones del Banco Central del Ecuador, ver [29].

Se propone modelar la RILD considerando las variables:

Variable	Descripción
Y	Reserva internacional de libre disponibilidad (RILD) de Ecuador
X_1	Ingresos al sistema financiero
X_2	Egresos del sistema financiero
X_3	Exportaciones de petróleo
X_4	Importaciones de derivados de petróleo

La información corresponde a series mensuales del período enero 2007 - agosto 2010. Para este caso se considera como conjunto de entrenamiento al período comprendido enero 2007 - diciembre 2009 y como conjunto de pruebas (*back testing*) al período enero 2010 - agosto 2010.

A continuación, se presentan los resultados de los métodos de mínimos cuadrados por lotes, mínimos cuadrados recursivo, agrupamiento difuso combinado y mínimos cuadrados recursivo combinado.

⁵Cabe recordar que antes de adoptar la dolarización de la economía ecuatoriana, el concepto que se manejaba era el de Reserva Monetaria Internacional, mismo que significaba el saldo entre activos y pasivos externos bajo control de las autoridades económicas del país; sus componentes eran el oro monetario, los derechos especiales de giro, la posición y uso de la reserva en el Fondo Monetario Internacional (FMI), los activos en divisas y otros títulos de crédito.

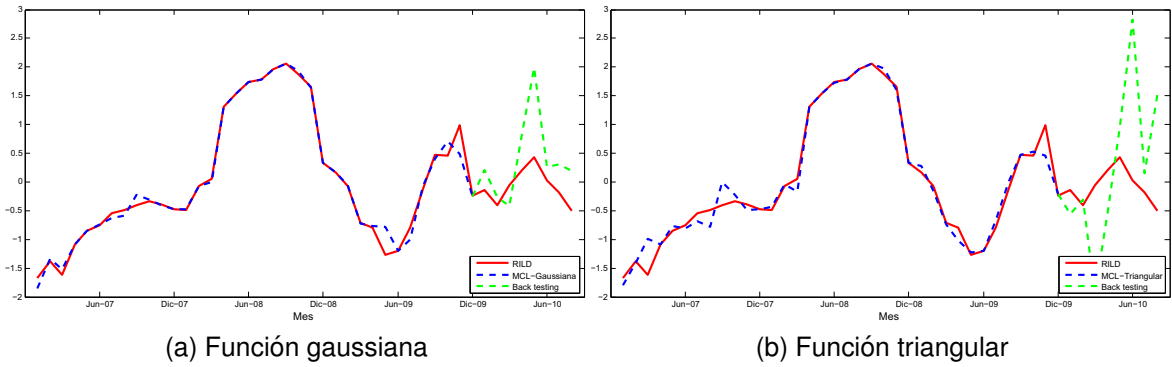


Figura 4.7: RILD mínimos cuadrados por lotes (MCL)

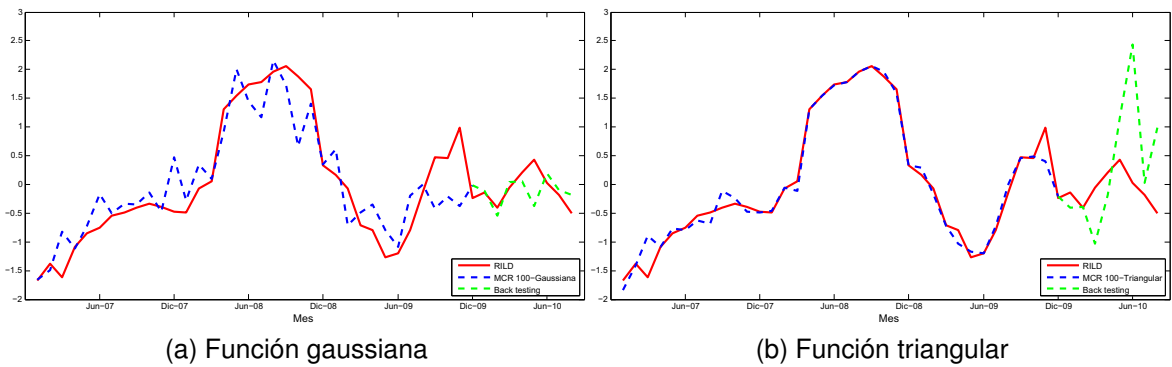


Figura 4.8: RILD mínimos cuadrados recursivo 100 iteraciones (MCR)

La Figura 4.7 muestra el ajuste para el método de mínimos cuadrados por lotes, con funciones de pertenencia Gaussiana y Triangular, se observa que el ajuste en el conjunto de entrenamiento es bueno para ambas series, pero al momento de realizar el *back testing*, los pronósticos para ambas series se tornan volátiles.

En el caso del método de mínimos cuadrados recursivo con 100 y 500 iteraciones (Figura 4.8 y Figura 4.9 respectivamente), para la función Gaussiana (Figura 4.8a y Figura 4.9a), el ajuste en el entrenamiento es un tanto volátil; aun cuando existe una similitud en el comportamiento entre la serie entrenada y la serie original. Cuando se analiza el *back testing* se nota cierta semejanza con los valores reales de la RILD. En el caso de la función de pertenencia Triangular (Figura 4.8b y Figura 4.9b), se observa que el entrenamiento es superior, pero los pronósticos en el *back*

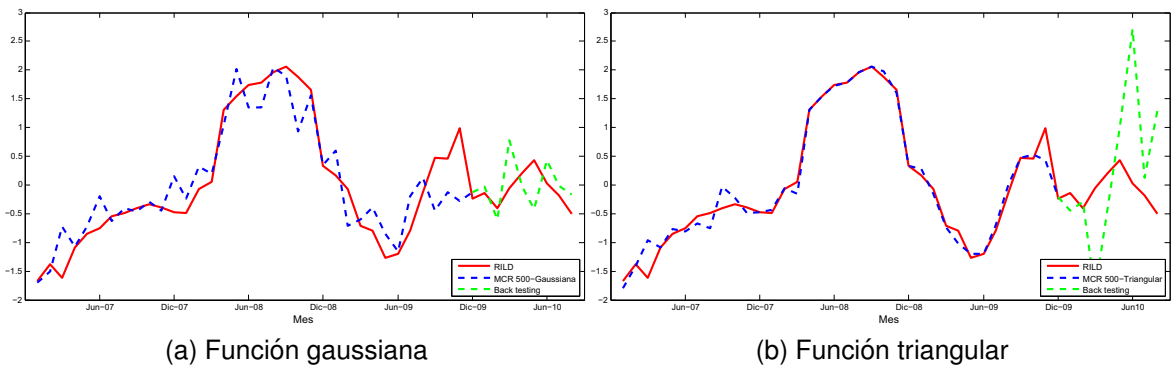


Figura 4.9: RILD mínimos cuadrados recursivo 500 iteraciones (MCR)

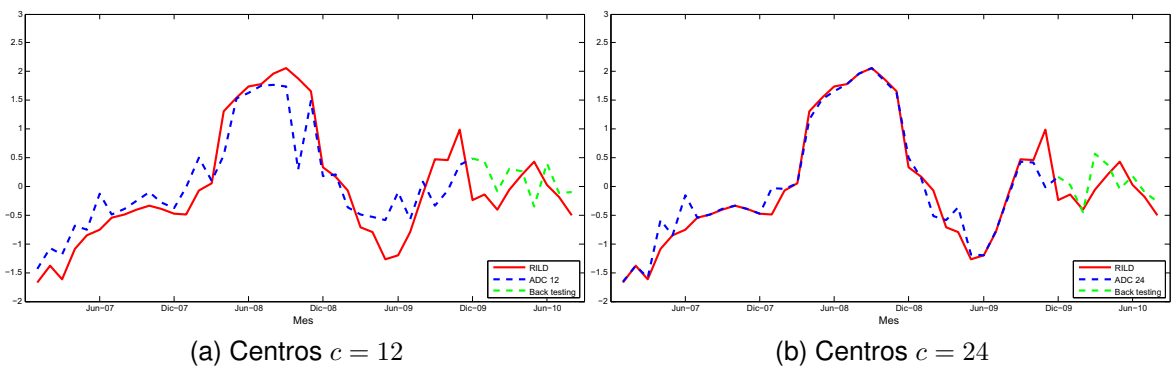


Figura 4.10: RILD agrupamiento difuso combinado (ADC)

testing no se asemejan a la serie original.

Para el agrupamiento difuso combinado, se utilizaron 12 y 24 centros, los resultados se pueden observar en la Figura 4.10. En el caso en el cual se utiliza 12 centros (Figura 4.10a) en el conjunto de entrenamiento se aprecia una semejanza con la serie original, algo muy parecido sucede con los pronósticos en el *back testing*. Cuando se aumenta a 24 centros (Figura 4.10b), se observa una mejora muy significativa tanto en el conjunto de entrenamiento como en el *back testing*, capturando de mejor manera los cambios de tendencia.

En el método de mínimos cuadrados recursivo combinado con 100 y 500 iteraciones (Figura 4.11 y Figura 4.12) se utilizaron 12 y 24 centros. Con 100 iteraciones en ajuste del conjunto de entrenamiento es muy parecido para 12 y 24 centros, pero

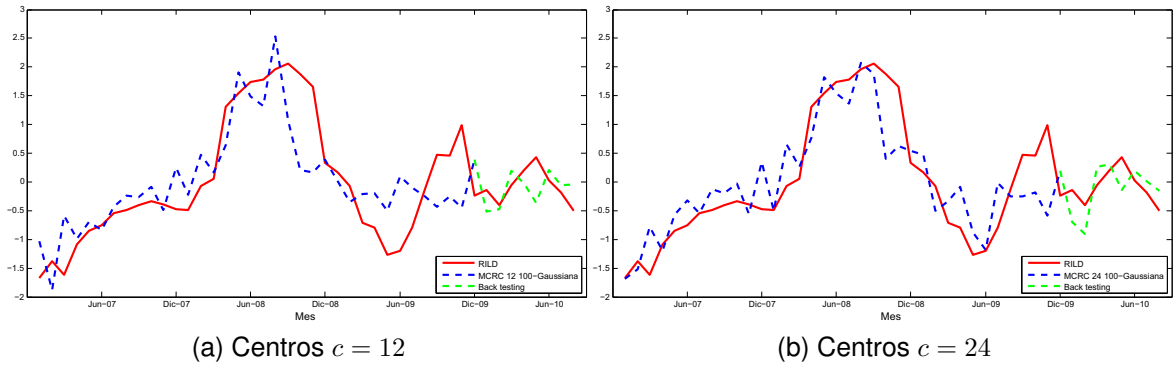


Figura 4.11: RILD mínimos cuadrados recursivo combinado 100 iteraciones (MCRC) Gaussiana

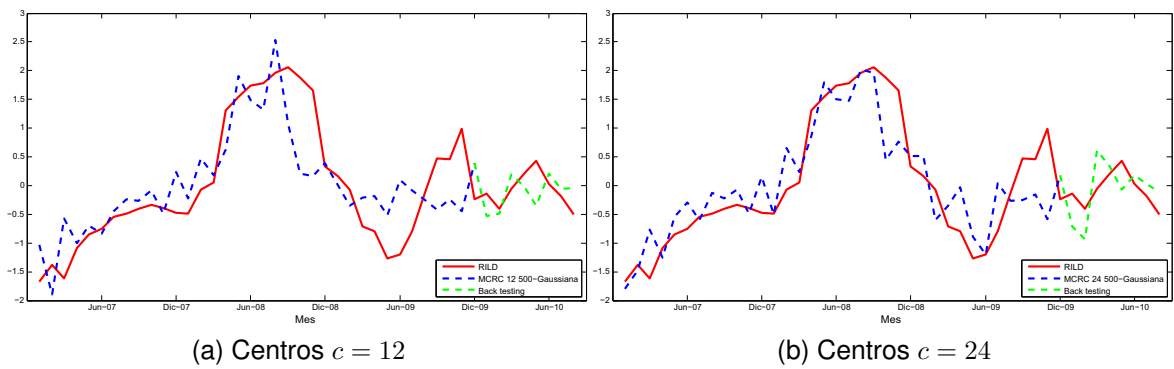


Figura 4.12: RILD mínimos cuadrados recursivo combinado 500 iteraciones (MCRC) Gaussiana

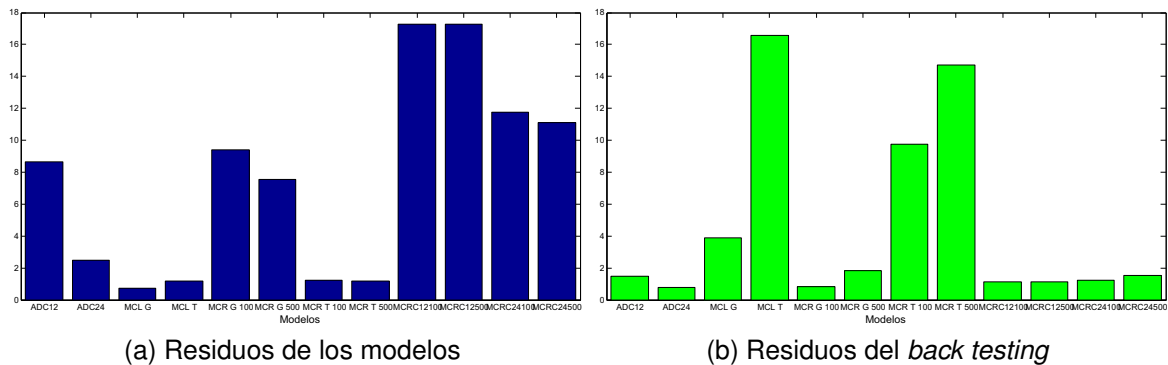
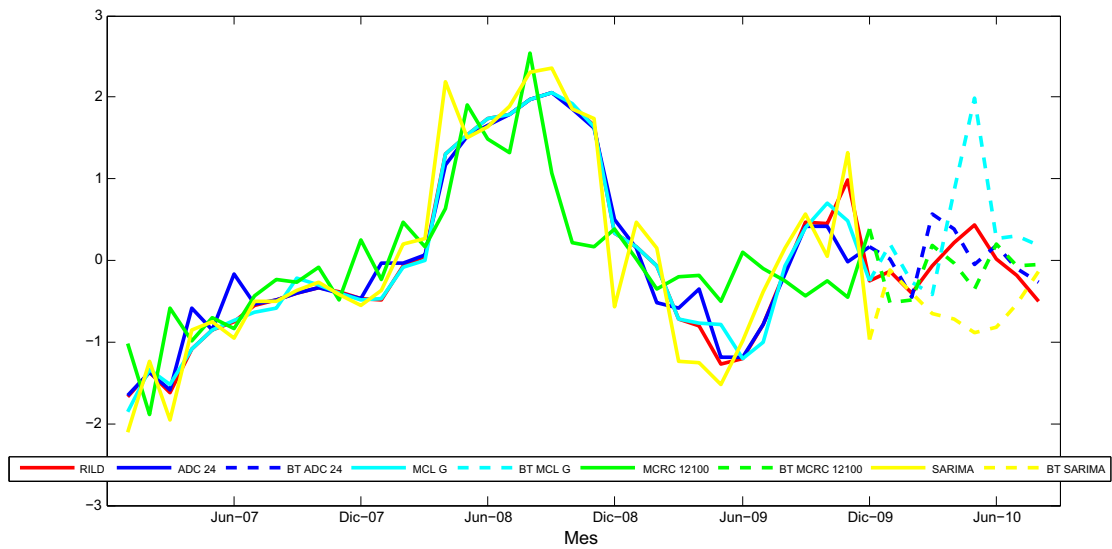


Figura 4.13: RILD residuos de los métodos automáticos y *back testing*

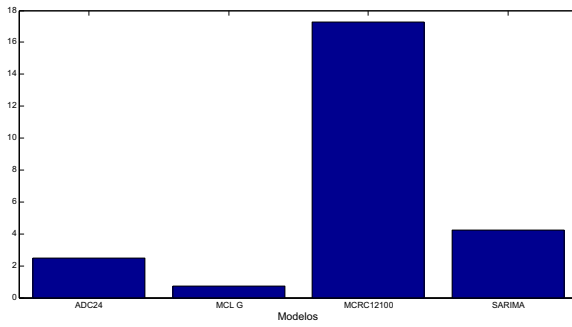
cuando se calcula los pronósticos hay mejores resultados con 12 centros, ver Figura 4.11a. Lo mismo sucede cuando se incrementa a 500 iteraciones, es decir, con 12 centros los pronósticos se asemejan mucho más a la serie real (Figura 4.12a).

La Figura 4.13 presenta para cada uno de los métodos la suma de errores del modelo y *back testing*. Los mejores métodos según estos criterios son mínimos cuadrados por lotes, agrupamiento difuso combinado 24 (24 grupos) y mínimos cuadrados recursivo combinado 12 100 (12 centros, 100 iteraciones).

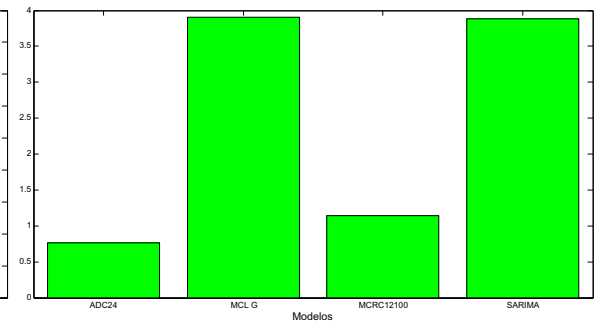
Con el objetivo de comparar los resultados de los métodos automáticos, se calculó un modelo SARIMA para la RILD utilizando el mismo conjunto de entrenamiento, los resultados del modelo SARIMA muestran que el modelo más apropiado para su estimación y pronóstico es un SARIMA(3, 0, 0)(1, 0, 0). Los resultados del modelo SARIMA son comparados con los mejores métodos automáticos en la Figura 4.14. Se puede apreciar que los métodos de mínimos cuadrados por lotes, agrupamiento difuso combinado 24 y el modelo SARIMA son los mejores cuando se ejecuta el entrenamiento (residuos de los modelos), pero cuando se calculan los pronósticos (residuos del *back testing*), los métodos de agrupamiento difuso combinado 24 y mínimos cuadrados recursivo combinado 12 100 son más competitivos, por tanto estos métodos poseen una mejor capacidad predictiva.



(a) Contraste



(b) Residuos de los modelos



(c) Residuos del *back testing*

Figura 4.14: RILD contraste

4.4.3. Índice de Liquidez de Instituciones Financieras

La liquidez de una institución financiera se entiende como la capacidad de contar con los fondos necesarios para:

- cumplir oportunamente con los compromisos financieros adquiridos, esto es cubrir retiros de depósitos a la vista, pagar una deuda, o pagar a proveedores;
- aprovechar oportunidades de negocio, es decir, compra de cartera, inversión a tasa atractiva, etc.

Cuando una institución enfrenta una escasez de fondos para cumplir sus obligaciones, tiene la necesidad de conseguir recursos alternativos o vender activos en condiciones desfavorables. Esto conlleva un alto costo financiero o una elevada tasa de descuento, incurriendo en pérdidas de valoración. Razón por la cual una institución debe establecer e implementar políticas y procedimientos idóneos que le permitan una adecuada administración de su liquidez, considerando la complejidad y volumen de las operaciones que realiza. Dichas políticas y procedimientos deben considerar los posibles escenarios y la forma en la que la institución responderá en el caso de que tales eventos se conviertan en realidad, ver [33]. Sin embargo, la preferencia por la liquidez puede llevar a acciones dañinas a la conveniencia económica general. Por ejemplo, si hay incertidumbre o los precios decaen, la población en general puede preferir mantener sus dineros, tendencia que si se generaliza, puede llevar a una disminución de la demanda o incluso a una corrida bancaria.

El siguiente modelo considera el índice definido como:

$$\text{Liquidez} = \frac{\text{Fondos disponibles}}{\text{Total de depósitos a corto plazo}}$$

por lo que se consideraron las siguientes variables:

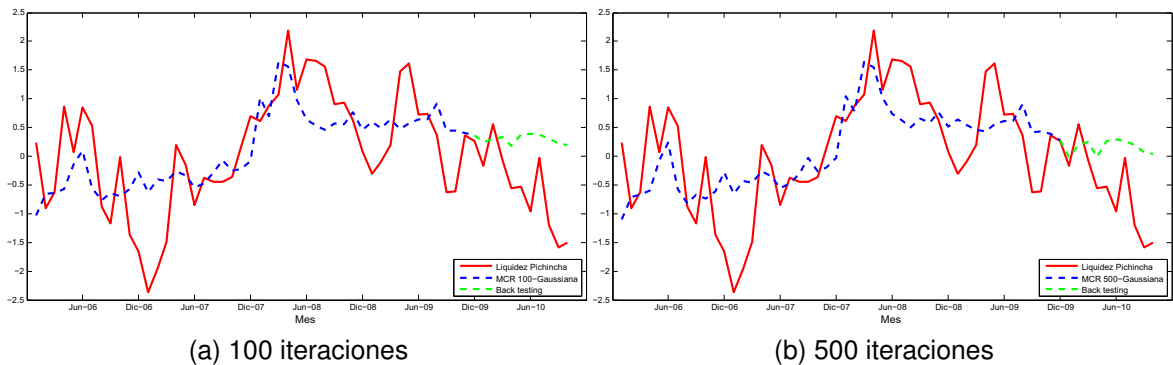


Figura 4.15: Banco Pichincha mínimos cuadrados recursivo (MCR)

Variable	Descripción
----------	-------------

Y_1	Índice de liquidez Banco Pichincha
Y_2	Índice de liquidez Banco Internacional
Y_3	Índice de liquidez Banco Amazonas
X_1	Índice de precios al consumidor
X_2	Inflación

La información corresponde a las series mensuales del período enero 2006 - octubre 2010. Como conjunto de entrenamiento se consideró al período enero 2006 - diciembre 2009 y como conjunto de pruebas (*back testing*) al período enero 2010 - octubre 2010.

A continuación, se presentan los resultados de los métodos de mínimos cuadrados recursivo, agrupamiento difuso combinado y mínimos cuadrados recursivo combinado para Banco Pichincha.

La Figura 4.15 muestra el ajuste para el método de mínimos cuadrados recursivo con 100 y 500 iteraciones y con función de pertenencia Gaussiana. Se observa que el ajuste en el conjunto de entrenamiento recoge la tendencia de la serie original, pero al momento de realizar el *back testing*, los pronósticos no se asemejan a los valores observados.

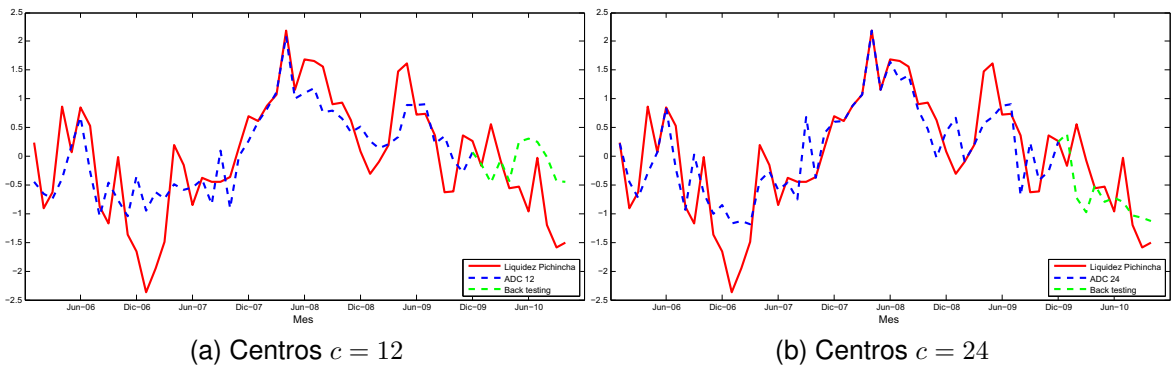


Figura 4.16: Banco Pichincha agrupamiento difuso combinado (ADC)

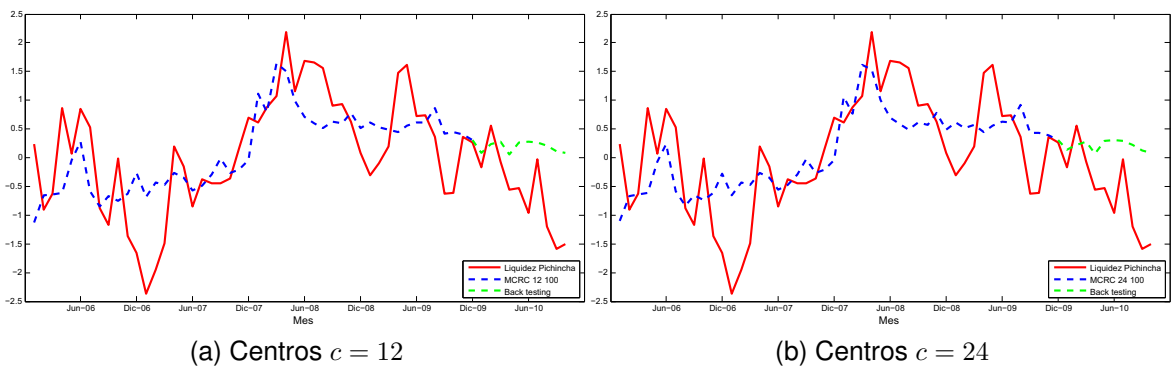


Figura 4.17: Banco Pichincha mínimos cuadrados recursivo combinado 100 iteraciones (MCRC) Gaussiana

Para el agrupamiento difuso combinado, se aplicó 12 y 24 centros, los resultados se observan en la Figura 4.16. Con 12 centros (Figura 4.16a) en el conjunto de entrenamiento se aprecia una semejanza con la serie original, algo similar sucede con los pronósticos en el *back testing*. Pero cuando se incrementa a 24 centros (Figura 4.16b), se observa una mejora muy significativa tanto en el conjunto de entrenamiento como en el *back testing*, capturando mucho mejor la tendencia de la serie original.

En el método de mínimos cuadrados recursivo combinado con 100 y 500 iteraciones (Figura 4.17 y Figura 4.18) se utilizaron 12 y 24 centros. Con 100 iteraciones el ajuste del conjunto de entrenamiento es muy parecido para 12 y 24 centros, lo

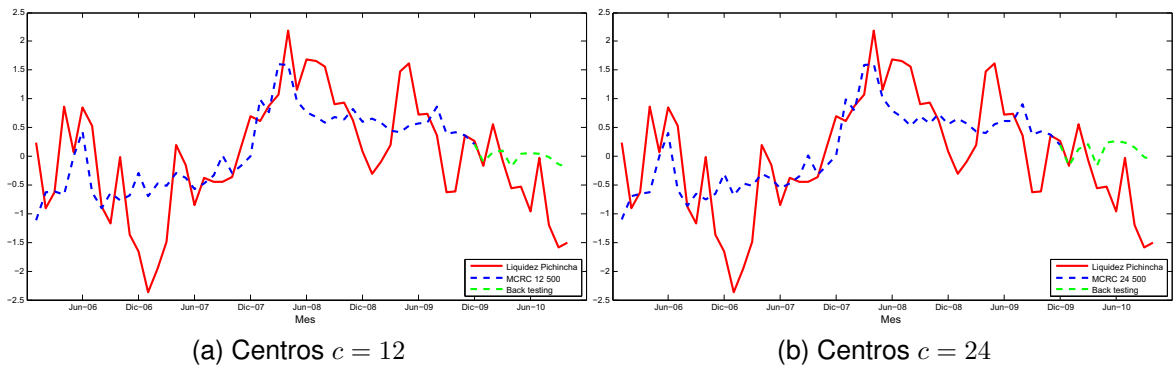


Figura 4.18: Banco Pichincha mínimos cuadrados recursivo combinado 500 iteraciones (MCRC) Gaussiana

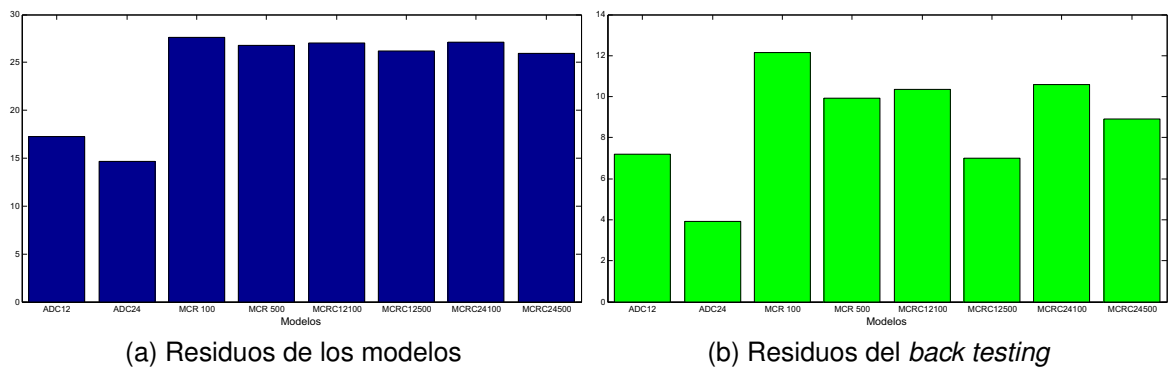
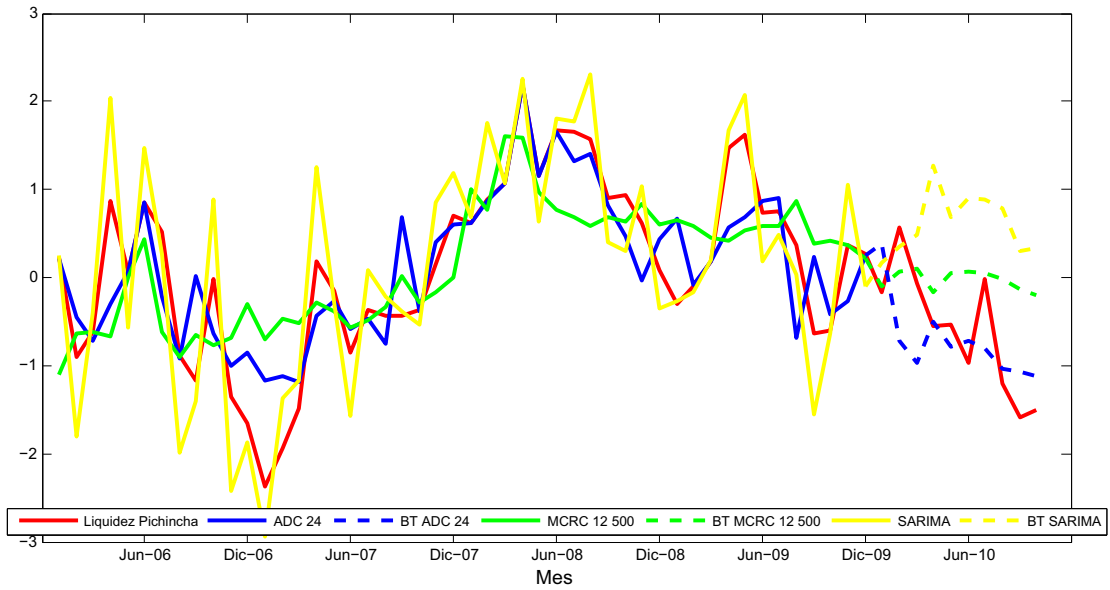


Figura 4.19: Banco Pichincha residuos de los métodos automáticos y *back testing*

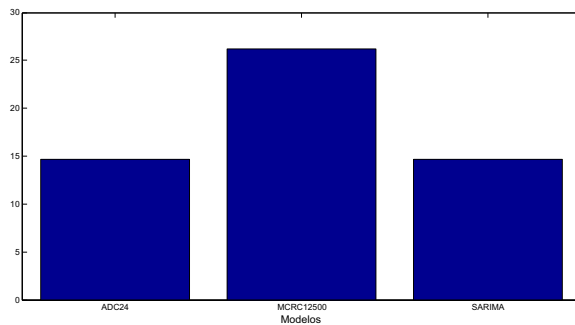
mismo ocurre cuando se calcula los pronósticos, en estos casos no se asemejan a la serie del índice de liquidez, ver Figura 4.17a. Lo mismo ocurre cuando se aumenta a 500 iteraciones, no se observa ninguna mejora (Figura 4.18a). Esto podría deberse a que el comportamiento de este indicador es muy volátil.

La Figura 4.19 presenta para cada uno de los métodos la suma de errores del modelo y *back testing*. Los mejores métodos según estos criterios son agrupamiento difuso combinado 24 (24 grupos) y mínimos cuadrados recursivo combinado 12 500 (12 centros, 500 iteraciones).

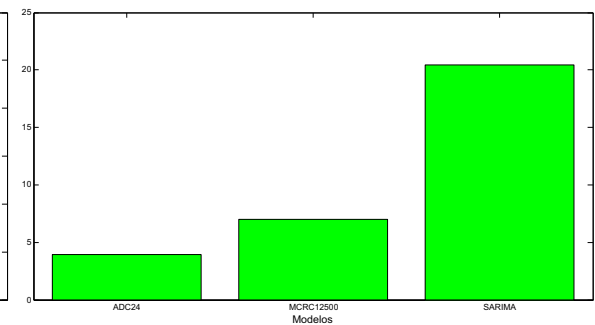
A continuación se calculó el modelo SARIMA para el índice de liquidez del Banco Pichincha utilizando el mismo conjunto de entrenamiento, los resultados del modelo



(a) Contraste



(b) Residuos de los modelos



(c) Residuos del *back testing*

Figura 4.20: Banco Pichincha contraste

SARIMA muestran que el modelo más apropiado para su estimación y pronóstico es un SARIMA(0, 1, 0)(2, 0, 0). Los resultados del modelo SARIMA son comparados con los mejores métodos automáticos en la Figura 4.20. El método de agrupamiento difuso combinado 24 y el modelo SARIMA son los mejores al momento de la estimación (residuos de los modelos), si se evalúan los pronósticos (residuos del *back testing*), los métodos de agrupamiento difuso combinado 24 y mínimos cuadrados recursivo combinado 12 500 son más competitivos.

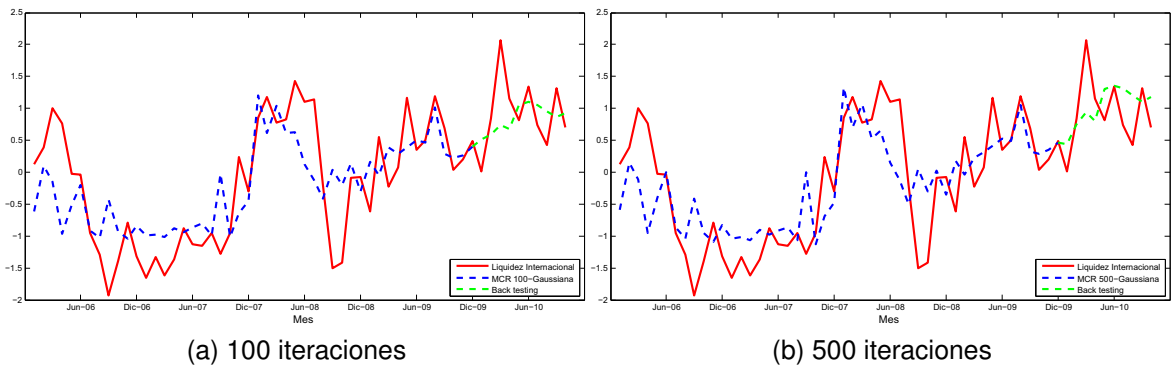


Figura 4.21: Banco Internacional mínimos cuadrados recursivo (MCR)

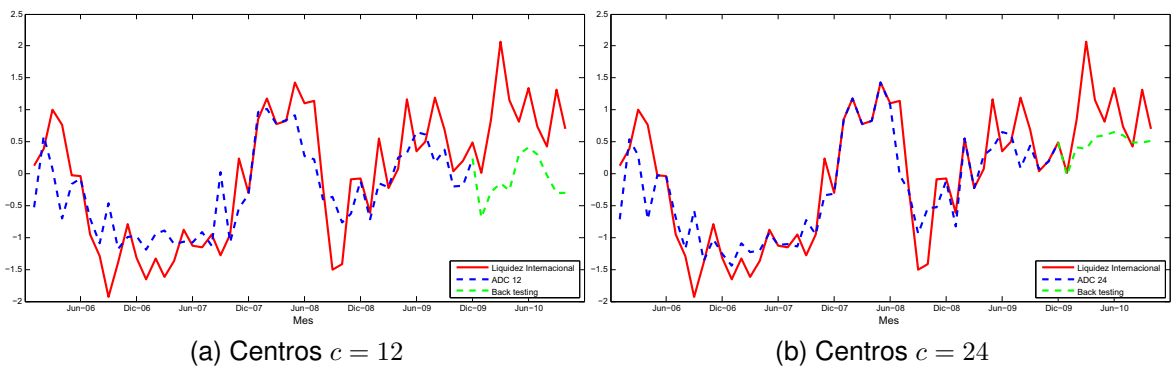


Figura 4.22: Banco Internacional agrupamiento difuso combinado (ADC)

A continuación, se presentan los resultados de los métodos de mínimos cuadrados recursivo, agrupamiento difuso combinado y mínimos cuadrados recursivo combinado para Banco Internacional.

La Figura 4.21 muestra el método de mínimos cuadrados recursivo con 100 y 500 iteraciones con función de pertenencia Gaussiana. Se observa que el ajuste en el conjunto de entrenamiento recoge la tendencia de la serie original, lo mismo ocurre cuando se ejecuta el *back testing*, en este caso no se aprecia ninguna diferencia significativa cuando se varía el número de iteraciones.

Para el método de agrupamiento difuso combinado, se aplicó 12 y 24 centros, los resultados se observan en la Figura 4.22. Con 12 centros (Figura 4.22a) en el conjunto de entrenamiento se aprecia una semejanza con la serie original, algo muy

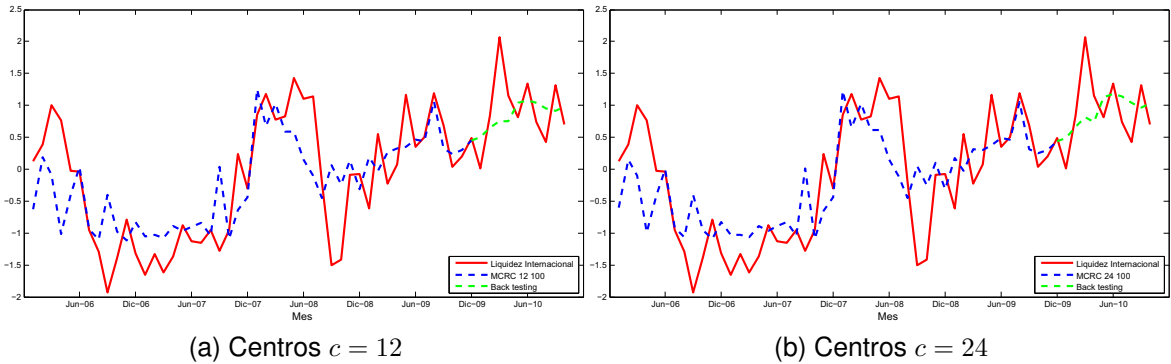


Figura 4.23: Banco Internacional mínimos cuadrados recursivo combinado 100 iteraciones (MCRC) Gaussiana

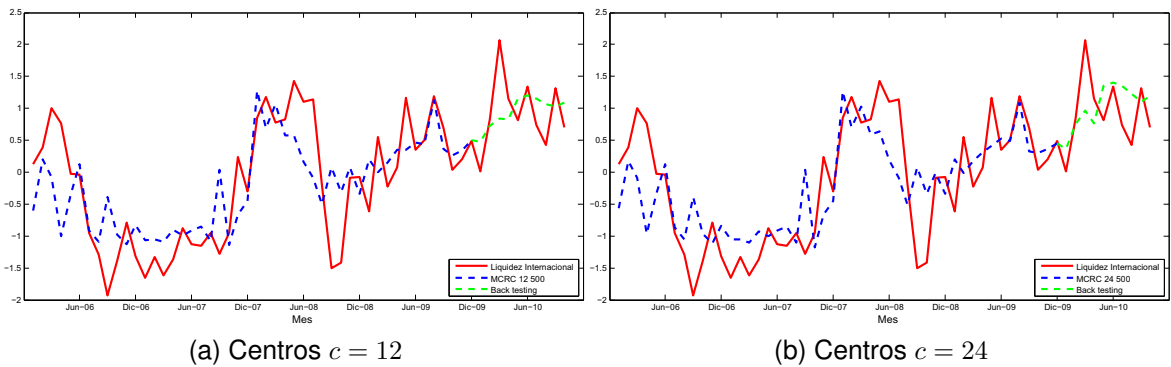


Figura 4.24: Banco Internacional mínimos cuadrados recursivo combinado 500 iteraciones (MCRC) Gaussiana

parecido sucede con los pronósticos en el *back testing*. Al incrementar a 24 centros (Figura 4.22b), se observa una mejora muy significativa tanto en el *back testing*, capturando de mejor manera la tendencia de la serie original.

El método de mínimos cuadrados recursivo combinado con 100 y 500 iteraciones (Figura 4.23 y Figura 4.24) se utilizaron 12 y 24 centros. Con 100 iteraciones el ajuste del conjunto de entrenamiento es muy parecido para 12 y 24 centros, lo mismo ocurre cuando se calcula los pronósticos, ver Figura 4.23a. De la misma forma, cuando se aumenta a 500 iteraciones, no se observa ninguna mejora (Figura 4.24a). Aun cuando el ajuste no se asemeja a la serie, este método modela la

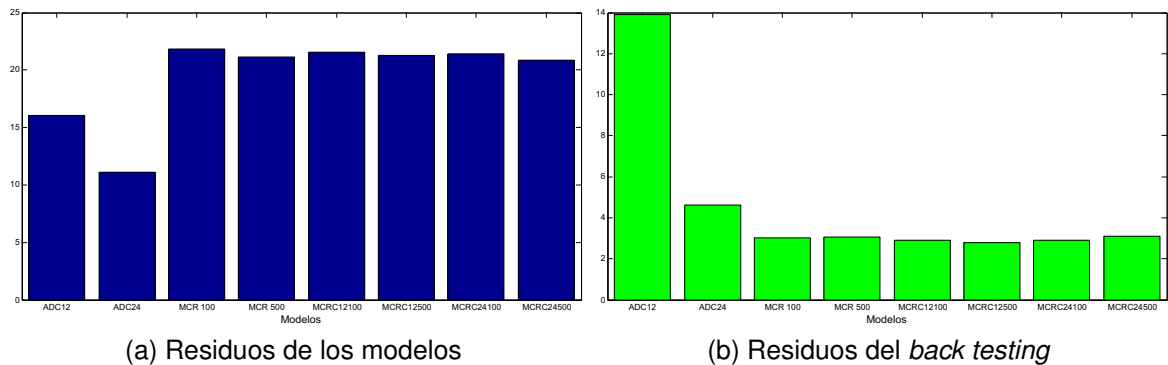
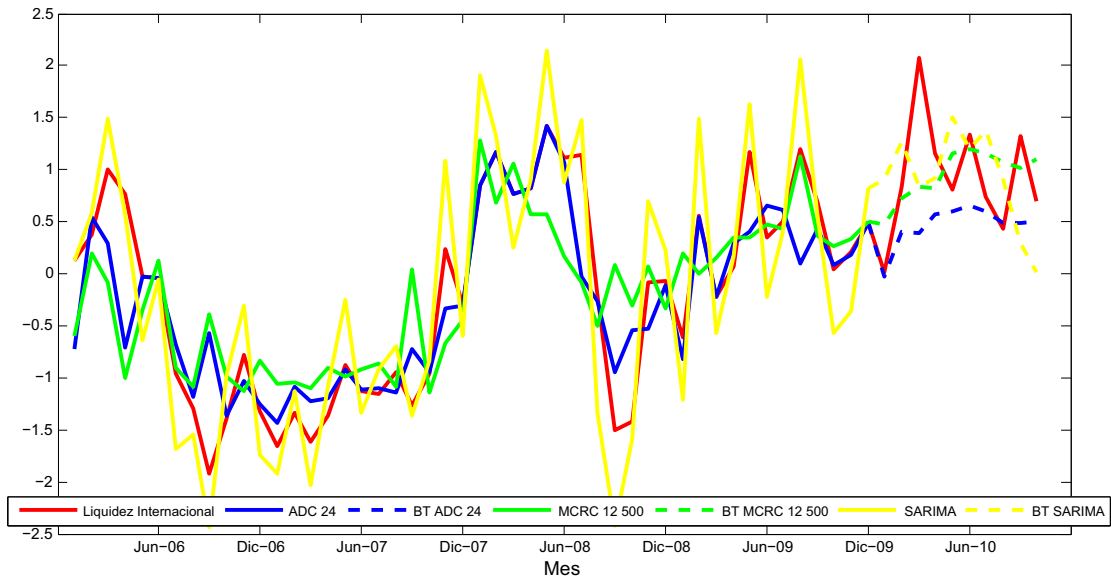


Figura 4.25: Banco Internacional residuos de los métodos automáticos y *back testing*

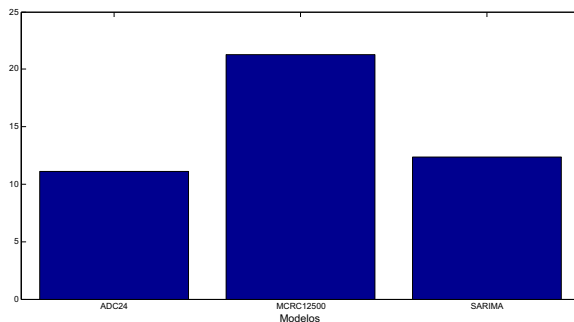
tendencia del fenómeno.

La Figura 4.25 presenta para cada uno de los métodos la suma de errores del modelo y *back testing*. Los mejores métodos según estos criterios son agrupamiento difuso combinado 24 (24 grupos) y mínimos cuadrados recursivo combinado 12 500 (12 centros, 500 iteraciones).

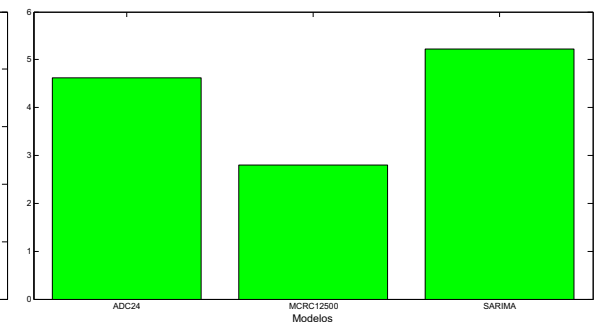
Nuevamente, se calculó un modelo SARIMA para el índice de liquidez del Banco Internacional utilizando el mismo conjunto de entrenamiento, los resultados del modelo SARIMA muestran que el modelo más apropiado para su estimación y pronóstico es un SARIMA(0, 1, 0)(0, 0, 2). El modelo SARIMA es comparado con los mejores métodos automáticos en la Figura 4.26. Al igual que el índice de liquidez de Banco Pichincha, se aprecia que el método de agrupamiento difuso combinado 24 y el modelo SARIMA son los mejores al momento de la estimación (residuos de los modelos), y al calcular los pronósticos (residuos del *back testing*), el método mínimos cuadrados recursivo combinado 12 500 es más competitivo.



(a) Contraste



(b) Residuos de los modelos



(c) Residuos del *back testing*

Figura 4.26: Banco Internacional contraste

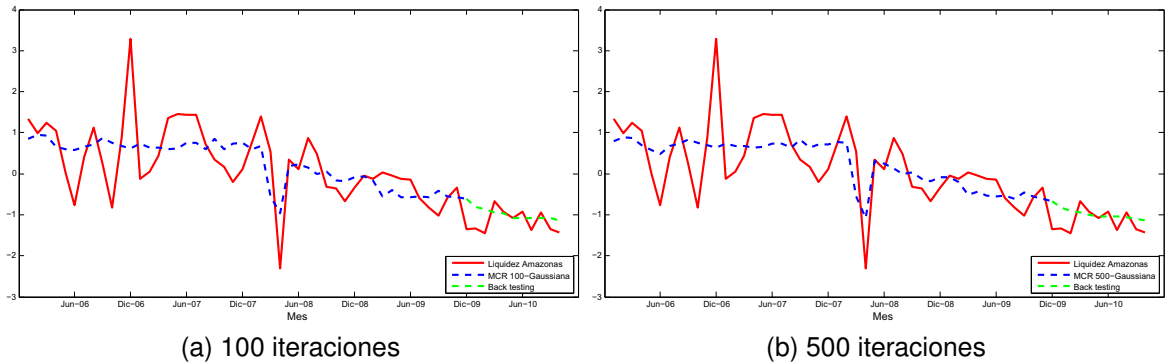


Figura 4.27: Banco Amazonas mínimos cuadrados recursivo (MCR)

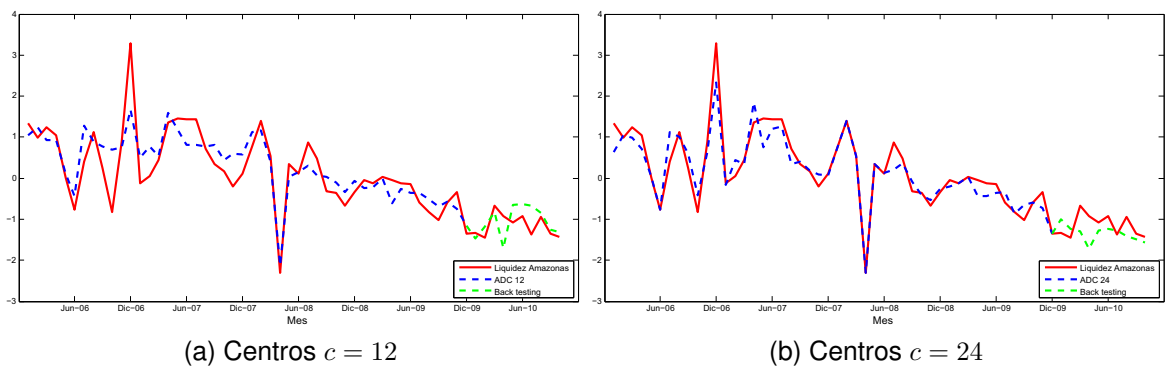


Figura 4.28: Banco Amazonas agrupamiento difuso combinado (ADC)

A continuación, se presentan los resultados de los métodos de mínimos cuadrados recursivo, agrupamiento difuso combinado y mínimos cuadrados recursivo combinado para Banco Amazonas.

En la Figura 4.27 se observa el método de mínimos cuadrados recursivo con 100 y 500 iteraciones con función de pertenencia Gaussiana. El ajuste en el conjunto de entrenamiento recoge la tendencia de la serie original, lo mismo ocurre cuando se ejecuta el *back testing*, nuevamente no se aprecia ninguna diferencia significativa cuando se varia el número de iteraciones.

Para el agrupamiento difuso combinado, se utilizó 12 y 24 centros (Figura 4.28). En este caso se observa que al incrementar el número de centros a 24, se logra un mejor ajuste en el entrenamiento, lo cual se refleja en los pronósticos, cuando se

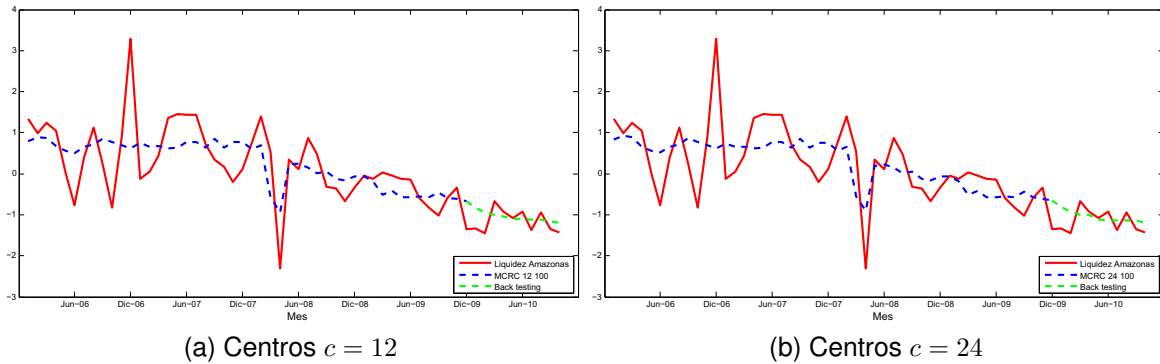


Figura 4.29: Banco Amazonas mínimos cuadrados recursivo combinado 100 iteraciones (MCRC) Gaussiana

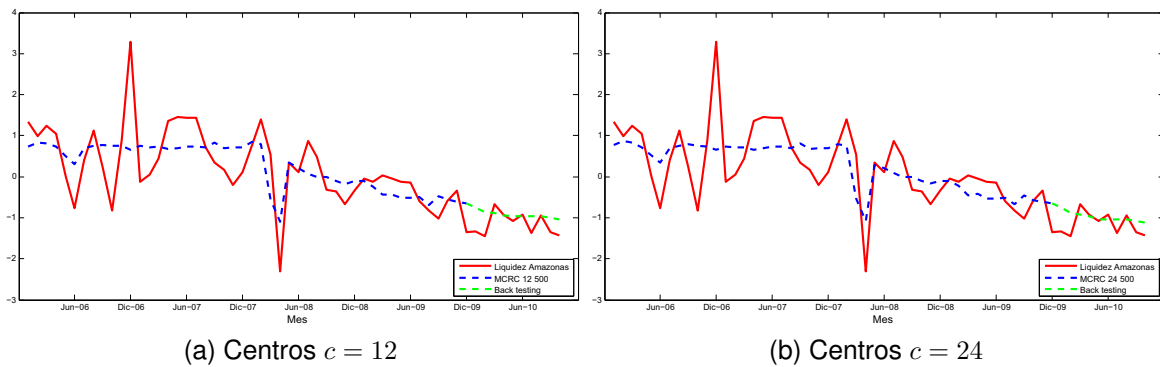


Figura 4.30: Banco Amazonas mínimos cuadrados recursivo combinado 500 iteraciones (MCRC) Gaussiana

aplica el *back testing*.

El método de mínimos cuadrados recursivo combinado con 100 y 500 iteraciones (Figura 4.23 y Figura 4.24) se utilizaron 12 y 24 centros. Con 100 iteraciones el ajuste del conjunto de entrenamiento es muy parecido para 12 y 24 centros, lo mismo ocurre cuando se calcula los pronósticos, se observa que no hay un buen ajuste, pero el modelo logra ajustar adecuadamente la tendencia de la serie, ver Figura 4.23a. De la misma forma, cuando se aumenta a 500 iteraciones, no se observa ninguna mejora, Figura 4.24a.

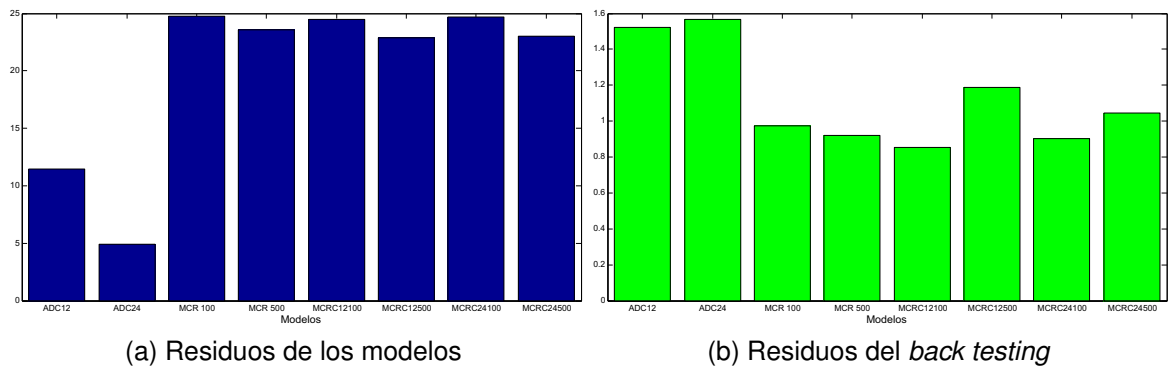


Figura 4.31: Banco Amazonas residuos de los métodos automáticos y *back testing*

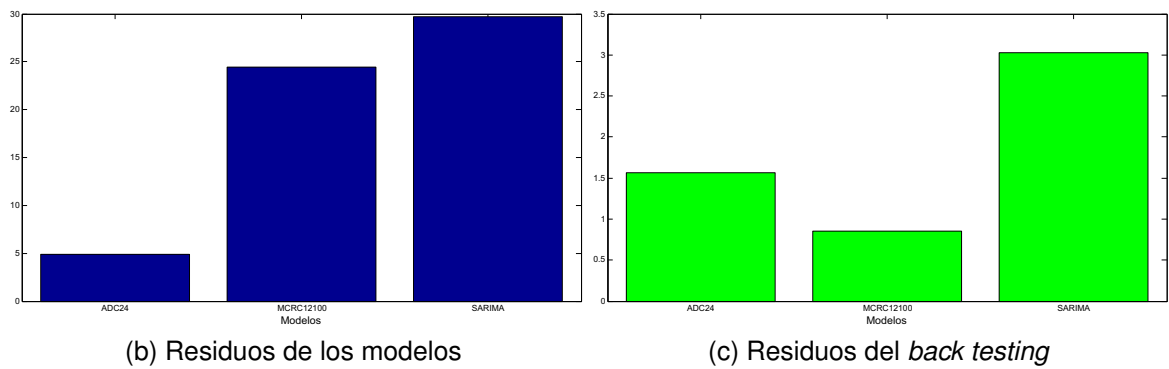
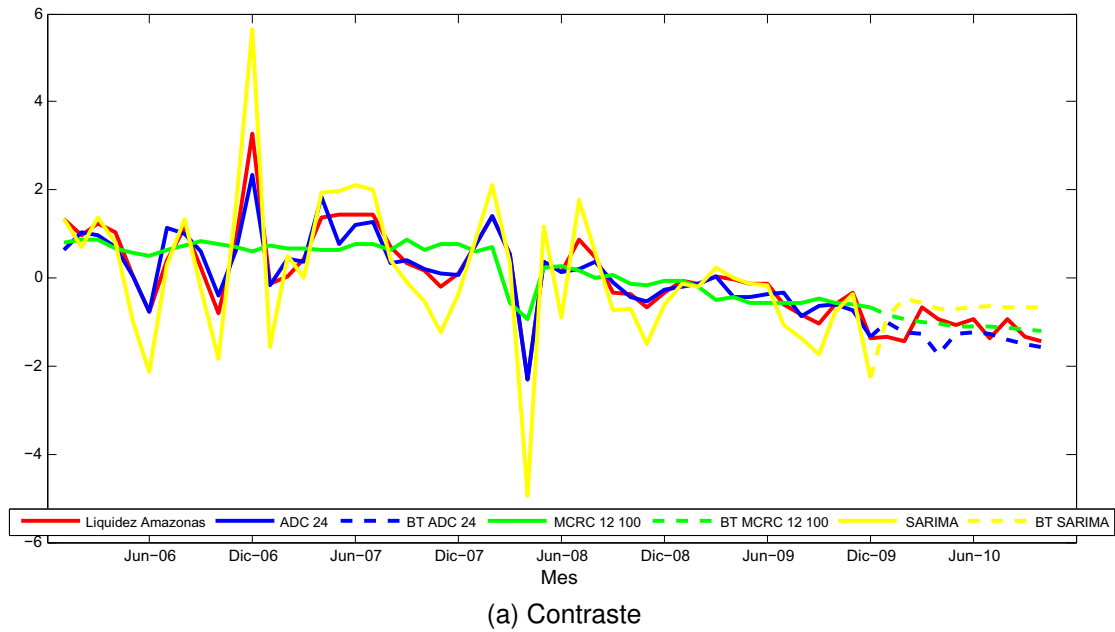


Figura 4.32: Banco Amazonas contraste

La Figura 4.31 presenta para cada uno de los métodos la suma de errores del modelo y *back testing*. Los mejores métodos según estos criterios son agrupamiento difuso combinado 24 (24 grupos) y mínimos cuadrados recursivo combinado 12 500 (12 centros, 500 iteraciones).

Al igual que en el caso anterior se calculó un modelo SARIMA para el índice de liquidez del Banco Amazonas utilizando el mismo conjunto de entrenamiento, los resultados del modelo SARIMA muestran que el modelo más apropiado para su estimación y pronóstico es un SARIMA(2, 1, 1)(0, 0, 0). Los resultados de los mejores métodos automáticos y del modelo SARIMA son comparados con en la Figura 4.32. Se puede notar que el método de agrupamiento difuso combinado 24 y el modelo SARIMA son los mejores al momento de la estimación (residuos de los modelos), pero cuando se calculan los pronósticos (residuos del *back testing*), al igual que el índice de liquidez del Internacional el método mínimos cuadrados recursivo combinado 12 500 es más competitivo.

4.4.4. Índices de Precios al Consumidor y al Productor de Ecuador

Los índices de precios al Productor (IPP) y al Consumidor (ICC) a menudo son vistos como los diferentes precios de los productos a lo largo de una cadena de producción. Por lo general, los precios de producción son los precios de las mercancías de entrada, que se utilizan para producir bienes finales. Los precios al consumidor son los precios de los bienes finales, que se venden a los consumidores.

Si el IPP e IPC están relacionados por la cadena de producción, los cambios en los precios de los materiales sin procesar (precios de productos básicos) deben pasar a través de los precios de los bienes intermedios y terminados y en última instancia al precio al consumidor, ver [30].

Según el Instituto Nacional de Estadísticas y Censos, el índice de precios al consumidor (IPC), es un indicador mensual nacional que mide los cambios en el tiempo del nivel general de los precios, correspondientes al consumo final de bienes y servicios de los hogares de estratos de ingreso: alto, medio y bajo; residentes en el área urbana del país, en donde se investiga el precio para los 299 artículos de la canasta fija de investigación [31]. El índice de precios al productor (IPP) es un indicador de alerta a la inflación, que mide la evolución de los precios de los bienes producidos para el mercado interno y la exportación, se calcula a partir de precios al productor, que se recogen en la fase de venta del bien producido mediante entrevista directa en los establecimientos manufactureros y mineros, en las unidades de producción agropecuaria (UPAs) y en los establecimientos pesqueros, ver [32].

Para esta aplicación se consideraron como variables dependientes los índices de precios y como variables independientes los precios de algunos productos básicos, los que se detallan en la siguiente tabla:

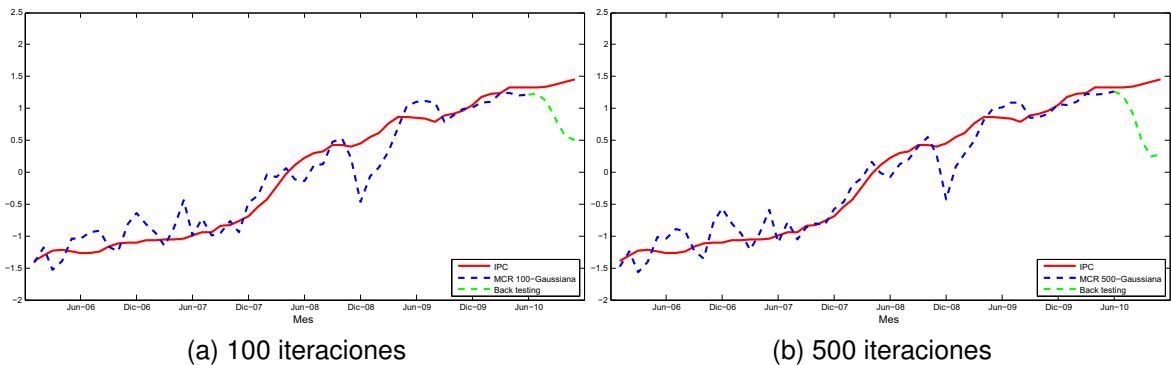


Figura 4.33: IPC mínimos cuadrados recursivo (MCR)

Variable	Descripción
Y_1	Índice de precios al consumidor (IPC) de Ecuador
Y_2	Índice de precios al productor (IPP) de Ecuador
X_1	Precio aceites
X_2	Precio azúcar
X_3	Precio cereales
X_4	Precio bebidas

La información corresponde a las series mensuales del período enero 2006 - noviembre 2010. Para esta aplicación se consideró como conjunto de entrenamiento al período enero 2006 - junio 2010 y como conjunto de pruebas (*back testing*) al período julio 2010 - noviembre 2010.

A continuación, se presentan los resultados de los métodos de mínimos cuadrados recursivo, agrupamiento difuso combinado y mínimos cuadrados recursivo combinado, para el índice de precios al consumidor.

La Figura 4.33 muestra el método de mínimos cuadrados recursivo, con 100 y 500 iteraciones, se observa que no existe una diferencia significativa entre los conjuntos de entrenamiento, cuando se varía el número de iteraciones, de hecho hay cierta volatilidad en el entrenamiento, lo cual no sucede con la serie original, dando

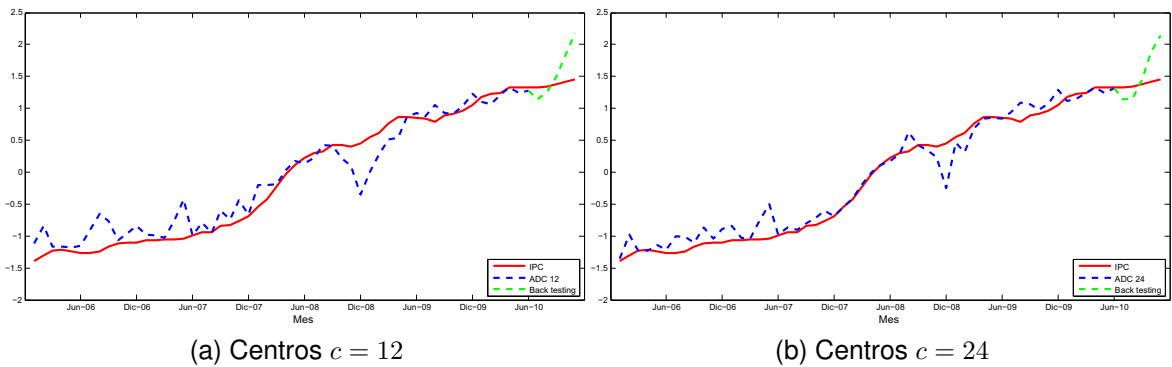


Figura 4.34: IPC agrupamiento difuso combinado (ADC)

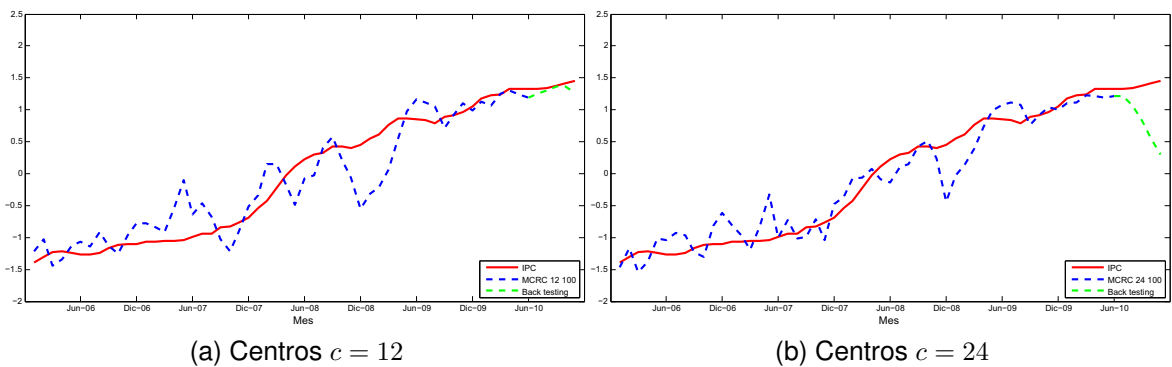


Figura 4.35: IPC mínimos cuadrados recursivo combinado 100 iteraciones (MCRC) Gaussiana

como resultado pronósticos volátiles en el *back testing*.

En el agrupamiento difuso combinado, sucede algo parecido al caso anterior (Figura 4.33), se presenta cierta volatilidad durante el entrenamiento, lo cual repercute en los pronósticos, haciendo que estos se vuelvan inestables. No se obtiene ninguna mejora cuando se varía el número de grupos (12 y 24 centros).

Con los mínimos cuadrados recursivos combinados se obtienen mejores resultados en los pronósticos, para el caso de 12 centros, aun cuando el entrenamiento no se asemeje a la serie del IPC (Figura 4.35 y Figura 4.36).

Las Figuras 4.33 a la 4.36 muestran el desempeño de cada uno de los métodos automáticos comparado con el valor observado. La Figura 4.37 presenta para cada

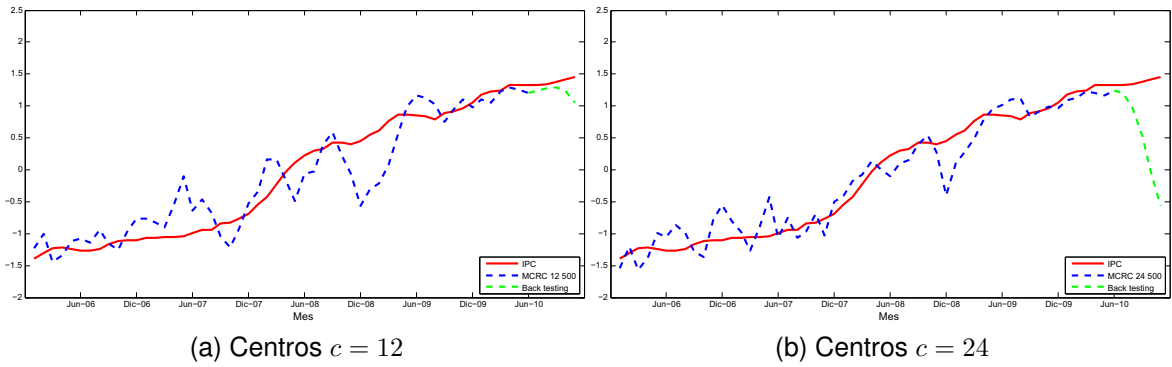


Figura 4.36: IPC mínimos cuadrados recursivo combinado 500 iteraciones (MGRC) Gaussiana

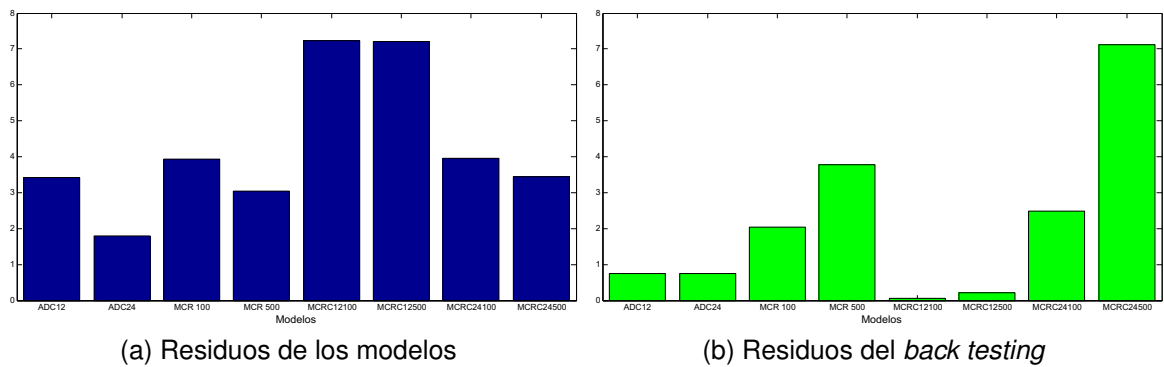
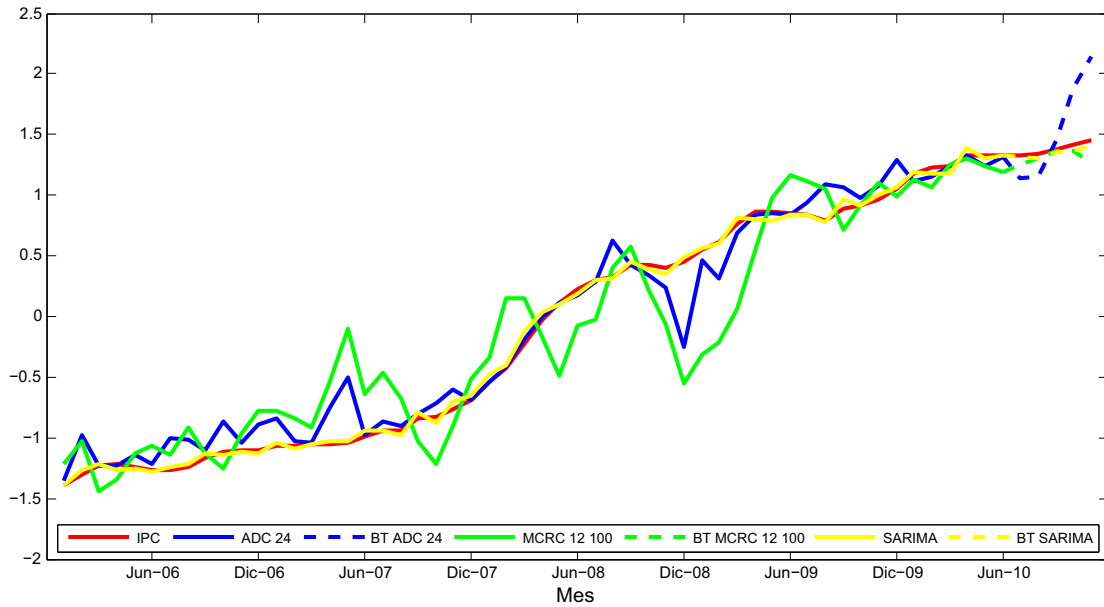
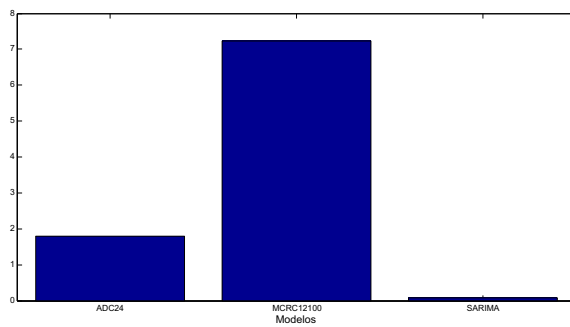


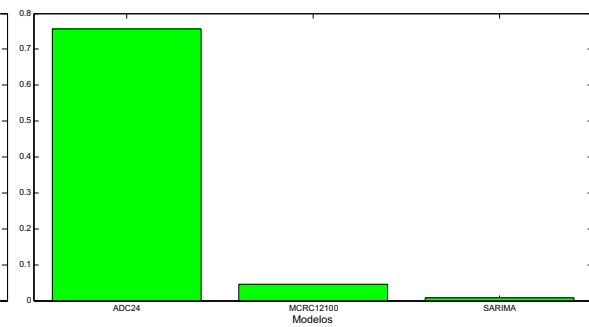
Figura 4.37: IPC residuos de los métodos automáticos y *back testing*



(a) Contraste



(b) Residuos de los modelos



(c) Residuos del *back testing*

Figura 4.38: IPC contraste

uno de los métodos la suma de errores del modelo y *back testing*. Los mejores métodos según estos criterios son agrupamiento difuso combinado 24 (24 grupos) y mínimos cuadrados recursivo combinado 12 100 (12 centros, 100 iteraciones).

Al igual que en los casos anteriores, se calculó un modelo SARIMA para el IPC utilizando el mismo conjunto de entrenamiento, los resultados del modelo SARIMA muestran que el modelo más apropiado para su estimación y pronóstico es un SARIMA(1, 1, 0)(1, 0, 0). Los resultados del modelo SARIMA son comparados con los mejores métodos automáticos en la Figura 4.38. El método de agrupamiento difuso combinado 24 y el modelo SARIMA son los mejores al momento de la estimación (residuos de los modelo), y al calcular los pronósticos (residuos del *back testing*), el método mínimos cuadrados recursivo combinado 12 500 y el modelo SARIMA son más competitivos.

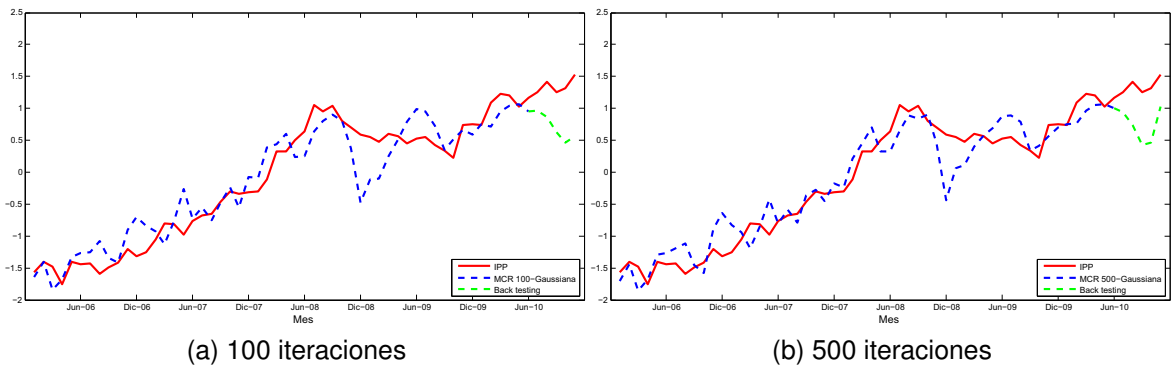


Figura 4.39: IPP mínimos cuadrados recursivo (MCR)

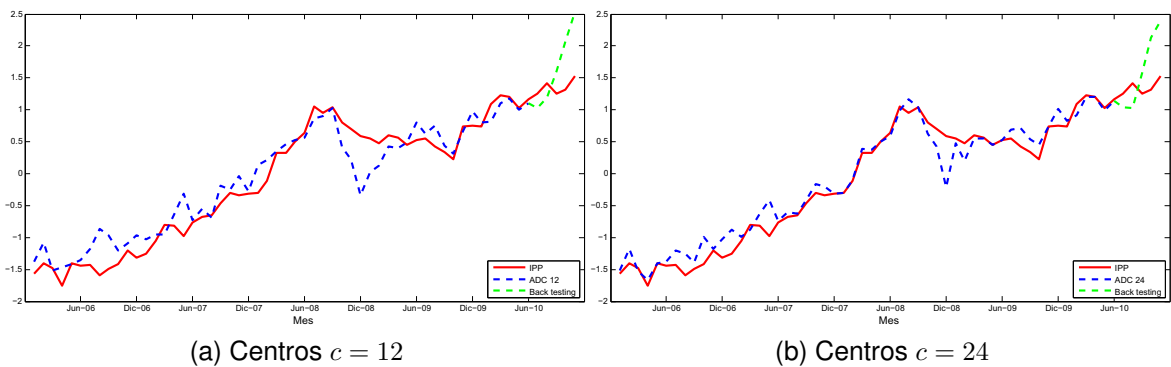


Figura 4.40: IPP agrupamiento difuso combinado (ADC)

A continuación, se presentan los resultados de los métodos de mínimos cuadrados recursivo, agrupamiento difuso combinado y mínimos cuadrados recursivo combinado, para el índice de precios al productor.

La Figura 4.39 muestra el método de mínimos cuadrados recursivo, con 100 y 500 iteraciones, se observa que no existe una diferencia significativa entre los conjuntos de entrenamiento, cuando varía el número de iteraciones, en este caso los pronósticos no se aproximan al comportamiento de la serie en el *back testing*.

Para el agrupamiento difuso combinado, se presenta una situación similar al caso del IPC, el ajuste durante la etapa de entrenamiento se vuelve volátil, lo que incide en los pronósticos, ver Figura 4.40.

Como sucedió en el IPC, el método de mínimos cuadrados recursivo combinado

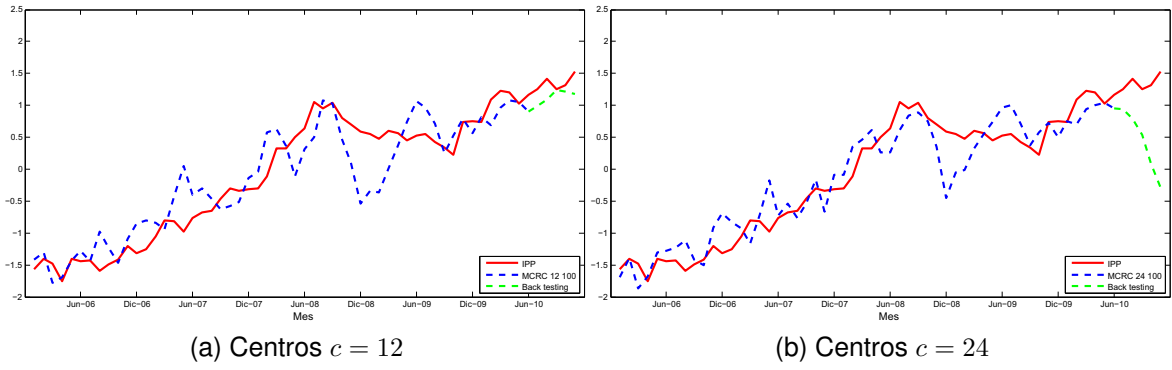


Figura 4.41: IPP mínimos cuadrados recursivo combinado 100 iteraciones (MCRC) Gaussiana

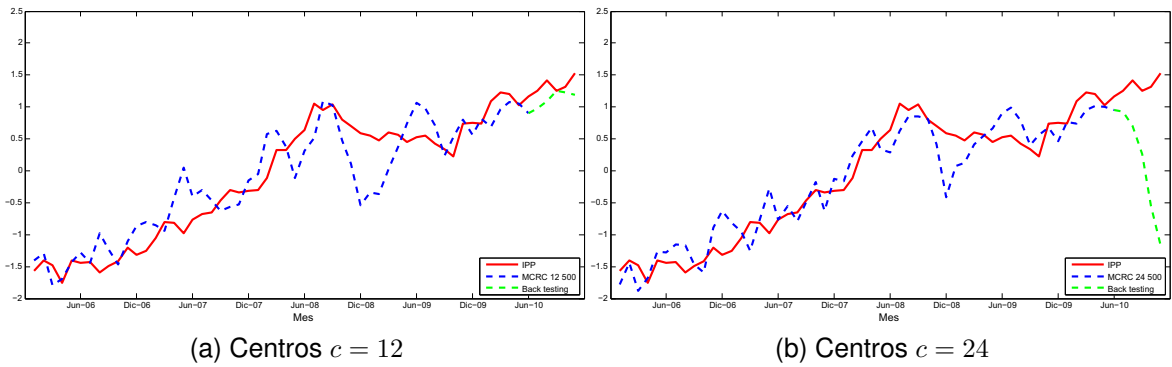


Figura 4.42: IPP mínimos cuadrados recursivo combinado 500 iteraciones (MCRC) Gaussiana

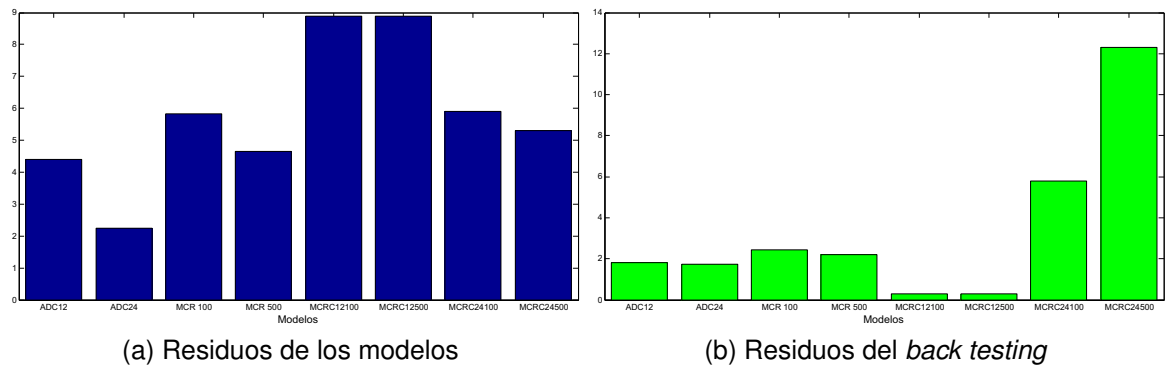
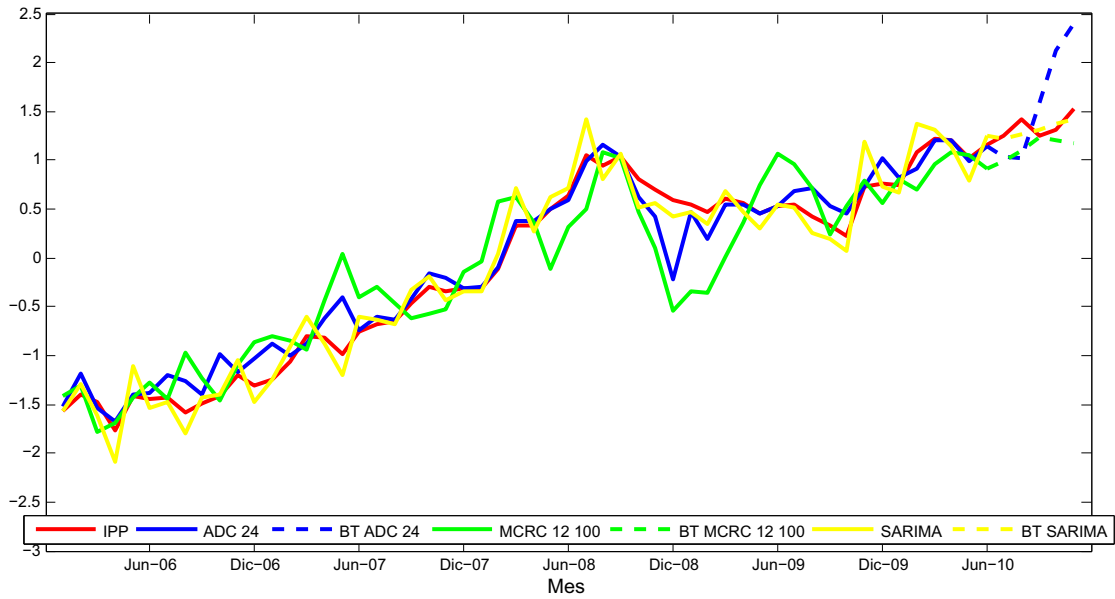


Figura 4.43: IPP residuos de los métodos automáticos y *back testing*

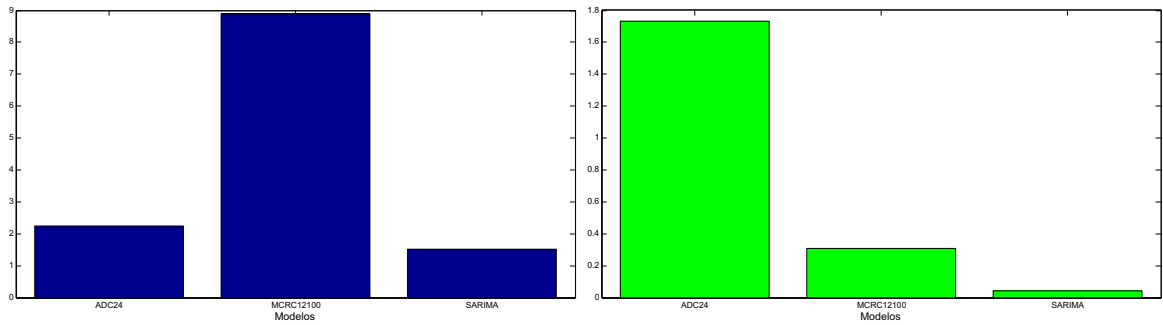
tiene mejores resultados para el caso de 12 centros, esto se observa en las Figuras 4.41 y Figuras 4.42.

En las Figuras 4.39 a la 4.42 se puede apreciar el desempeño de cada uno de los métodos automáticos comparado con el valor real. La Figura 4.43 presenta para cada uno de los métodos la suma de errores del modelo y *back testing*. Los mejores métodos según estos criterios son agrupamiento difuso combinado 24 (24 grupos) y mínimos cuadrados recursivo combinado 12 100 (12 centros, 100 iteraciones).

Se calculó un modelo SARIMA para el IPP utilizando el mismo conjunto de entrenamiento, los resultados del modelo SARIMA muestran que el modelo más apropiado para su estimación y pronóstico es un SARIMA(1, 1, 0)(0, 0, 0). Los resultados de los mejores métodos automáticos son comparados con el modelo SARIMA en la Figura 4.44. Al igual que el IPP, se puede apreciar que el método de agrupamiento difuso combinado 24 y el modelo SARIMA son los mejores al momento de la estimación (residuos de los modelos), pero cuando se calculan los pronósticos (residuos del *back testing*), el método mínimos cuadrados recursivo combinado 12 500 y el modelo SARIMA son más competitivos.



(a) Contraste



(b) Residuos de los modelos

(c) Residuos del *back testing*

Figura 4.44: IPP contraste

CAPÍTULO 5

CONCLUSIONES Y RECOMENDACIONES

En este trabajo se describen los métodos automáticos de regresión difusa: mínimos cuadrados por lotes, mínimos cuadrados recursivo, aprendizaje desde el ejemplo modificado y agrupamiento difuso combinado. Se generalizan los métodos al caso de entrada y salida múltiple, lo cual permite aplicarlos a una gama de problemas más amplia. Además, se propone un nuevo método *mínimos cuadrados recursivo combinado*, el cual ha mostrado un gran desempeño tal como se puede observar en las pruebas numéricas realizadas. En general, los resultados obtenidos son muy alentadores y corresponden a una medida cuantitativa del potencial de estos métodos en aplicaciones económicas.

Para el caso del PIB no petrolero, considerando que el número de observaciones empleado es pequeño y sin embargo los métodos logran describir y caracterizar la tendencia de la serie. Con un mayor número de observaciones (RILD, índice de liquidez e índices de precios), existe mayor eficiencia al caracterizar la tendencia y sobre todo el *back testing* en los métodos automáticos y en particular en los métodos combinados y el método propuesto. Esto se debe a que una mayor cantidad de información contribuye al mejoramiento en la construcción de reglas del sistema lo que caracteriza de mejor manera al fenómeno de estudio. El mismo criterio justifica el mejor desempeño de los métodos combinados. En general, el *back testing* del método propuesto proporciona estimaciones muy competitivas comparadas con los modelos SARIMA y los métodos de la literatura.

Como regla general, para definir métodos combinados, proponemos usar méto-

dos de extracción de reglas para inicializar los parámetros del sistema difuso y luego utilizar algún método de mínimos cuadrados, dependiendo de la aplicación que se considere, de esta manera se podría investigar los métodos:

- aprendizaje desde el ejemplo modificado combinado: aprendizaje desde ejemplo + mínimos cuadrados,
- aprendizaje desde el ejemplo modificado recursivo: aprendizaje desde ejemplo + mínimos cuadrados recursivo

El estudio de estos métodos es un problema abierto.

Finalmente, cabe señalar que la precisión requerida y el costo computacional son parámetros a tener en cuenta al momento de decidir que método aplicar, e.g., el método de mínimos cuadrados recursivo no requiere la inversión de una matriz y por tanto es más estable numéricamente. Sin embargo, su costo computacional es elevado comparado con el método de mínimos cuadrados por lotes. Por otra parte, la escala de los datos puede causar inestabilidad en los algoritmos. Por esta razón, en general, es recomendable estandarizar los datos.

Definir un criterio automático de selección de los métodos para una aplicación cualquiera es un problema abierto que esperamos sea parte de un trabajo futuro.

Apéndice

A. Librería de funciones

Para esta aplicación se desarrolló un librería en el paquete R, el cual es un lenguaje de programación para análisis estadístico que permite a los usuarios definir sus propias funciones, también posee una amplia gamma de herramientas estadísticas. Entre sus ventajas:

- Reducen el tamaño de los archivos ejecutables: gran parte del código puede estar almacenado en bibliotecas y no en el propio ejecutable lo que redundaría en una mejor modularización¹.
- Pueden estar compartidas entre varias aplicaciones: si el código es suficientemente genérico, puede resultar de utilidad para múltiples aplicaciones.
- Facilitan la gestión y aprovechamiento de la memoria del sistema: la carga dinámica permite al sistema operativo aplicar algoritmos que mejoren el rendimiento del sistema cuando se carguen estas bibliotecas. Además, al estar compartidas, basta con mantener una copia en memoria para todos los programas que la utilicen.
- Brindan mayor flexibilidad frente a cambios: es posible mejorar el rendimiento o solucionar pequeños errores distribuyendo únicamente una nueva versión de la biblioteca dinámica. Nuevamente, esta corrección o mejora será aprovechada por todas las aplicaciones que compartan la biblioteca.

¹En programación orientada a objetos, la Modularidad es la propiedad que permite subdividir una aplicación en partes más pequeñas, cada una de las cuales debe ser tan independiente como sea posible de la aplicación en sí y de las restantes partes.

```

# MÉTODOS AUTOMÁTICOS DE REGRESIÓN DIFUSA (mard)
# RODRIGO PAÚL CAJAMARCA
# FECHA INICIO 2011-03-15

```

El script esta disponible gratuitamente en [35].

```

# Función: mard_ValPertenenenciaNormal
# Cálculo de la matriz Xi para los puntos de entrenamiento (valores de pertenencia), para función normal
# ENTRADAS:
# X: Matriz de variables de entrada
# C: Matriz de centros de funciones de pertenencia de variables de entrada
# Sigma: Matriz de dispersiones de funciones de pertenencia de variables de entrada
# R: Número de reglas del sistema
# SALIDAS:
# Xi: Valores de pertenencia

mard_ValPertenenenciaNormal <- function(X,C,Sigma,R)
{
M <- dim(X)[1] # M número de entradas y salidas de datos
n <- dim(X)[2] # n número de variables de entrada
Mu <- matrix(NA,nrow=R,ncol=M)
den <- matrix(NA,nrow=1,ncol=M)
Xi <- matrix(NA,nrow=R,ncol=M)
for(l in 1:M){
  for(i in 1:R){
    Mu[i,l] <- 1
    for(j in 1:n){
      Mu[i,l] <- Mu[i,l]*exp(-0.5*((X[l,j]-C[i,j])/Sigma[i,j])^2)
    }
  }
  den[l] <- sum(Mu[,l])
  Xi[,l] <- Mu[,l]/den[l]
}
return(Xi)
}

```

```

# Función: mard_centros
#   Calcula los centros de las funciones de pertenencia para las variables de entrada
# ENTRADAS:
#   X: Matriz de variables de entrada
#   R: Número de reglas del sistema
# SALIDAS:
#   C: Matriz de centros de funciones de pertenencia de variables de entrada

mard_centros <- function(X,R)
{
M <- dim(X)[1]   # M número de entradas y salidas de datos
n <- dim(X)[2]   # n número de variables de entrada
C <- matrix(NA,nrow=R,ncol=n)
for(j in 1: n){
  for(i in 1: R){
    C[i,j] <- abs(X[i+1,j]-X[i,j])*0.5 + X[i,j]
  }
}
return(C)
}

# Función: mard_centros_iniciales
#   Calcula los centros iniciales para el agrupamiento difuso c-medias
# ENTRADAS:
#   X: Matriz de variables de entrada
#   c: Número de centros (Número de reglas del sistema)
# SALIDAS:
#   V: Matriz de centros iniciales para funciones de pertenencia de variables de entrada

mard_centros_iniciales <- function(X,c)
{
minmax <- matrix(NA,nrow=n,ncol=2)
V <- matrix(NA,nrow=R,ncol=n)
for(j in 1:n){
  minmax[j,1] <- min(X[,j])
  minmax[j,2] <- max(X[,j])
  for(i in 1:R){
    V[i,j] <- minmax[j,1] + i*((minmax[j,2] - minmax[j,1])/(R+1))
  }
}
}

```

```

}
return(V)
}

# Función: mard_centros_difusos
#   Calcula los centros difusos para el agrupamiento difuso c-medias
# ENTRADAS:
#   X: Matriz de variables de entrada
#   V: Matriz de centros iniciales para funciones de pertenencia de variables de entrada
#   R: Número de reglas del sistema
#   e_c: Tolerancia para el cálculo de centros
#   m: Parámetro de solapamiento
# SALIDAS:
#   V: Matriz de centros difusos finales para funciones de pertenencia de variables de entrada

mard_centros_difusos <- function(X,V,R,e_c,m)
{
M <- dim(X)[1] # M número de entradas y salidas de datos
n <- dim(X)[2] # n número de variables de entrada
flg <- 0
numiter <- 0
Mu <- matrix(NA,nrow=M,ncol=R)
VNew <- matrix(NA,nrow=R,ncol=n)
while(flg == 0){
  Mu <- mard_fcm_mu(X,V,m)
  for(j in 1:R){
    num <- matrix(0,nrow=1,ncol=n)
    den <- 0
    for(i in 1:M){
      num <- num + X[i,]*(Mu[i,j]^m)
      den <- den + (Mu[i,j]^m)
    }
    VNew[j,] = num/den
  }
  test <- matrix(0,nrow=1,ncol=R)
  for(i in 1:R){
    test[1,i] = sqrt(as.numeric(t(V[i,]-VNew[i,])%*%(V[i,]-VNew[i,])))
  }
  for(i in 1:R){

```



```

        flg <- 1
        if(test[1,i]>e_c){
            flg <- 0
            i = R
        }
    }
    V <- VNew
}
return(V)
}

# Función: mard_fcm_mu
#   Calcula los valores de pertenencia para el agrupamiento difuso
# ENTRADAS:
#   X: Matriz de variables de entrada
#   V: Matriz de centros difusos para funciones de pertenencia de variables de entrada
#   m: Parámetro de solapamiento
# SALIDAS:
#   Mu: Valores de pertenencia

mard_fcm_mu <- function(X,V,m)
{
M <- dim(X)[1]   # M número de entradas y salidas de datos
n <- dim(X)[2]   # n número de variables de entrada

Mu <- matrix(NA,nrow=M,ncol=R)
for(i in 1:M){
    for(j in 1:R){
        z <- 0
        for(k in 1:R){
            z <- z + (as.numeric(t(X[i,]-V[j,])%*(X[i,]-V[j,]))/as.numeric(t(X[i,]-V[k,])%*(X[i,]-V[k,])))^(1/(m-1))
        }
        Mu[i,j] <- 1/z
    }
}
return(Mu)
}

```

```

# Función: mard_Xhat
#   Crea la matriz Xhat para el agrupamiento difuso combinado
# ENTRADAS:
#   X: Matriz de variables de entrada
# SALIDAS:
#   Xhat: Matriz de variables de entrada modificada

mard_Xhat <- function(X)
{
M <- dim(X)[1]   # M número de entradas y salidas de datos
n <- dim(X)[2]   # n número de variables de entrada
Xhat <- matrix(NA,nrow=M,ncol=n+1)
for(i in 1:M){
  Xhat[i,1] <- 1
  for(j in 1:n){
    Xhat[i,j+1] <- X[i,j]
  }
}
return(Xhat)
}

# Función: mard_adc
# Calcula el método de agrupamiento difuso combinado
# ENTRADAS:
#   X: Matriz de variables de entrada
#   Y: Matriz de variable de salida
#   R: Número de reglas del sistema
#   e_c: Tolerancia para el cálculo de centros
#   m: Parámetro de solapamiento
# SALIDAS:
#   output: Estimaciones para la matriz de las variables de entrada
#   V: Matriz de centros difusos para funciones de pertenencia de variables de entrada

mard_adc <- function(X,Y,R,e_c,m)
{
M <- dim(X)[1]   # M número de entradas y salidas de datos
n <- dim(X)[2]   # n número de variables de entrada
V <- mard_centros_iniciales(X,R)

```

```

V <- mard_centros_difusos(X,V,R,e_c,m)
Mu <- mard_fcm_mu(X,V,m)
Xhat <- mard_Xhat(X)
Dhat <- matrix(0,nrow=M,ncol=M)
A <- matrix(0,nrow=n+1,ncol=R)
for(j in 1:R){
  for(i in 1:M){
    Dhat[i,i] <- Mu[i,j]^2
  }
  A[,j] <- (solve(t(Xhat)%*%Dhat%*%Xhat))%*%t(Xhat)%*%Dhat%*%Y
}
Muf <- mard_fcm_mu(X,V,m)
output <- matrix(NA,nrow=1,ncol=M)
numf <- matrix(0,nrow=1,ncol=M)
denf <- matrix(0,nrow=1,ncol=M)
for(i in 1:M){
  numf[1,i] <- 0
  denf[1,i] <- 0
  for(j in 1:R){
    numf[1,i] <- numf[1,i] + as.numeric(t(A[,j])%*%Xhat[i,]*Muf[i,j])
  }
  denf[1,i] <- denf[1,i] + as.numeric(sum(Muf[i,]))
  output[1,i] <- numf[1,i]/denf[1,i]
}
return(list("output"=output,"V"=V))
}

# Función: mard_mcr_mod
#   Calcula el método de mínimos cuadrados recursivo modificado
# ENTRADAS:
#   X: Matriz de variables de entrada
#   Y: Matriz de variable de salida
#   sigma_1: Dispersión para las funciones de pertenencia de las variables de entrada
#   NRLS: Número de iteraciones para el proceso recursivo
#   lambda: factor de pérdida de memoria
# SALIDAS:
#   output: Estimaciones para la matriz de las variables de entrada
#   thetahatresult: Parámetro theta estimado

```

```

mard_mcr_mod <- function(X,Y,sigma_1,NRLS,alpha,lambda)
{
M <- dim(X)[1] # M número de entradas y salidas de datos
n <- dim(X)[2] # n número de variables de entrada
Sigma <- matrix(sigma_1,nrow=R,ncol=n)
R <- M - 1 # R número de reglas
NRLSM <- NRLS*M
C <- matrix(NA,nrow=R,ncol=n)
C <- mard_centros(X,R)
thetahatpast <- matrix(0,nrow=R,ncol=1)
thetahat <- matrix(0,nrow=R,ncol=NRLSM)
Ppast <- alpha * diag(R)
eyeP <- diag(R)
P <- matrix(NA,nrow=R,ncol=R)
Xi <- matrix(NA,nrow=R,ncol=M)
Xi <- mard_ValPertenenciaNormal(X,C,Sigma,R)
Xiaux <- matrix(0,nrow=R,ncol=1)
lm <- 0
for(k in 1: NRLSM){
  if(lm <= M){
    lm <- lm + 1
    if(lm > M){
      lm = 1
    }
  }
  Xiaux <- Xi[,lm]
  Yaux <- Y[,lm]
  factor <- as.numeric(1/(lambda + as.numeric(t(Xiaux)%*%Ppast%*%Xiaux)))
  P <- (1/lambda)*(eyeP - factor*(Ppast%*%Xiaux%*%t(Xiaux))%*%Ppast)
  thetahat[,k] <- thetahatpast + (Yaux - as.numeric(t(Xiaux)%*%thetahatpast))*(P%*%Xiaux)
  Ppast <- P
  thetahatpast <- thetahat[,k]
}
thetahatresult <- thetahatpast
output <- matrix(NA,nrow=1,ncol=M)
output <- t(thetahatresult)%*%Xi
return(list("output"=output,"thetahatresult"=thetahatresult))
}

```

```

# Función: mard_mcl
#   Calcula el método de mínimos cuadrados por lotes
# ENTRADAS:
#   X: Matriz de variables de entrada
#   Y: Matriz de variable de salida
#   sigma_1: Dispersión para las funciones de pertenencia de las variables de entrada
# SALIDAS:
#   output: Estimaciones para la matriz de las variables de entrada
#   thetahat: Parámetro theta estimado

mard_mcl<- function(X,Y,sigma_1)
{
M <- dim(X)[1]   # M número de entradas y salidas de datos
n <- dim(X)[2]   # n número de variables de entrada
R <- M - 1       # R número de reglas
C <- matrix(NA,nrow=R,ncol=n)
C <- mard_centros(X,R)
Sigma <- matrix(sigma_1,nrow=R,ncol=n)
Xi <- matrix(NA,nrow=R,ncol=M)
Xi <- mard_ValPertenenciaNormal(X,C,Sigma,R)
Phi <- matrix(NA,nrow=M,ncol=R)
Phi <- t(Xi)
thetahat <- matrix(NA,nrow=M,ncol=1)
thetahat <- solve(t(Phi)%*%Phi)%*%t(Phi)%*%Y
output <- matrix(NA,nrow=1,ncol=M)
output <- t(thetahat)%*%Xi
return(list("output"=output,"thetahat"=thetahat,"C"=C))
}

# Función: mard_mcrs
# Calcula el método de mínimos cuadrados recursivo combinado
# ENTRADAS:
#   X: Matriz de variables de entrada
#   Y: Matriz de variable de salida
#   R: Número de reglas del sistema
#   e_c: Tolerancia para el cálculo de centros

```

```

# m: Parámetro de solapamiento
# sigma_1: Dispersión para las funciones de pertenencia de las variables de entrada
# NRLS: Número de iteraciones para el proceso recursivo
# lambda: factor de pérdida de memoria
# SALIDAS:
# output: Estimaciones para la matriz de las variables de entrada
# V: Matriz de centros difusos para funciones de pertenencia de variables de entrada

mard_mrcr <- function(X,Y,R,e_c,m,NRLS,sigma_1,alpha,lambda)
{
M <- dim(X)[1]
n <- dim(X)[2]
Sigma <- matrix(sigma_1,nrow=R,ncol=n)
V <- mard_centros_iniciales(X,R)
V <- mard_centros_difusos(X,V,R,e_c,m)
C <- matrix(NA,nrow=R,ncol=n)
C <- V
NRLSM <- NRLS*M
thetahatpast <- matrix(0,nrow=R,ncol=1)
thetahat <- matrix(0,nrow=R,ncol=NRLSM)
Ppast <- alpha * diag(R)
eyeP <- diag(R)
P <- matrix(NA,nrow=R,ncol=R)
Xi <- mard_ValPertenenciaNormal(X,C,Sigma,R)
lm <- 0
Xiaux <- matrix(0,nrow=R,ncol=1)
for(k in 1: NRLSM){
  if(lm <= M){
    lm <- lm + 1
    if(lm > M){
      lm = 1
    }
  }
  Xiaux <- Xi[,lm]
  Yaux <- Y[lm,]
  factor <- as.numeric(1/(lambda + as.numeric(t(Xiaux)%*%Ppast%*%Xiaux)))
  P <- (1/lambda)*(eyeP - factor*(Ppast%*%Xiaux%*%t(Xiaux))%*%Ppast)
  thetihat[,k] <- thetihatpast + (Yaux - as.numeric(t(Xiaux)%*%thetihatpast))*(P%*%Xiaux)
  Ppast <- P
  thetihatpast <- thetihat[,k]
}
}

```

```
}  
thetahatresult <- thetahatpast  
output <- matrix(NA,nrow=1,ncol=M)  
output <- t(thetahatresult)%*%Xi  
return(list("output"=output,"V"=V))  
}
```

Bibliografía

- [1] Medina Hurtado Santiago and Paniagua Gómez Gisela (2006), *Fuzzy Inference Systems to Creditworthiness Analysis*. Dyna, Año 75, Nro. 154, pp. 215-229.
- [2] Terano Toshiro, Asai Kiyoji and Sugeno Michio (1992), *Fuzzy Systems Theory and Its Applications*. Academic Press Inc.
- [3] Rudolf Kruse and Andreas Nürnberger (1997), *Learning Methods for Fuzzy Systems*. Department of Computer Science, University of Magdeburg Germany.
- [4] Zadeh Lotfi A. (1965), *Fuzzy Sets*. University of California, Berkeley, California.
- [5] Dubois Didier and Prade Henri (1980), *Fuzzy Sets and Systems: Theory and Applications*. Mathematics in Science and Engineering, Volume 144.
- [6] Rutkowski Leszek (2004), *Flexible Neuro-Fuzzy Systems, Structures, Learning and Performance Evaluation*. New York, Boston, Dordrecht, London, Moscow.
- [7] Ibrahim Ahmad M. (2003), *Fuzzy Logic for Embedded Systems Applications*. Elsevier Science.
- [8] El-Hawary Mohamed E. (1998), *Electric Power Applications of Fuzzy Systems*. Institute of Electrical and Electronics Engineers, Inc.
- [9] Galindo José, Urrutia Angélica and Piattini Mario (2006), *Fuzzy Databases: Modeling, Design and Implementation*. Idea Group Inc.
- [10] Lee Kwang H. (2005), *First Course on Fuzzy Theory and Applications*. Springer - Verlag Berlin Heidelberg.

- [11] Pedrycz Witold and Gomide Fernando (2007), *Fuzzy Systems Engineering Toward Human-Centric Computing*. John Wiley and Sons, Inc.
- [12] Zhang Huaguang and Liu Derong (2006), *Fuzzy Modeling and Fuzzy Control*. Birkhäuser Boston.
- [13] Palit Ajoy K. and Popovic Dobrivoje (2005), *Computational Intelligence in Time Series Forecasting, Theory and Engineering Applications*. Springer-Verlag.
- [14] Sivanandam S., Sumathi S. and Deepa S. (2007), *Introduction to Fuzzy Logic using MATLAB*. Springer Verlag.
- [15] Passino Kevin M. and Yurkovich Stephen (1998), *Fuzzy Control*. Addison Wesley Longman.
- [16] Ross Timothy J. (2004), *Fuzzy logic with engineering applications*. Second Edition. Wiley, England.
- [17] Hayashi Fumio (2000), *Econometrics*. Princeton University Press.
- [18] Nagamune Ryoza (2008), *Intelligent Control*. Department of Mechanical Engineering, University of British Columbia.
- [19] Haykin Simon (2001) *Adaptive Filter Theory*. Prentice Hall Information and System Sciences Series, Third Edition.
- [20] Laukonen Eric G. and Passino Kevin M. (1995). *Training Fuzzy Systems to Perform Estimation and Identification*. Engineering Applications Artificial Intelligence Vol. 8 No. 5 pp. 499-514. Elsevier Science Ltd. Printed in Great Britain.
- [21] Bezdek James C., Keller James and Krisnapuram Raghu (2005) *Fuzzy Models and Algorithms for Pattern Recognition and Image Processing*. Springer Science Business Media.

- [22] Xu Rui and Wunsch Donald C. II (2009), *Clustering*. John Wiley and Sons, Inc.
- [23] Giles David and Draeseke Robert (2001), *Econometric Modelling Based on Pattern Recognition via the Fuzzy c-Means Clustering Algorithm*. Working Paper EWP0101, Department of Economics, University of Victoria.
- [24] Donoso Salgado Sergio (2006), *Análisis de regresión difusa: Nuevos enfoques y aplicaciones*. Tesis Doctoral, Universidad de Granada.
- [25] De Oliveira J. Valente and Witold Pedrycz (2007), *Advances in Fuzzy Clustering and its Applications*. John Wiley and Sons, Inc.
- [26] Hayes Monson H. (1996). *Statistical Digital Signal Processing and Modeling*. John Wiley and Sons, Inc.
- [27] Montgomery Douglas C. and Runger George C. (2003). *Applied Statistics and Probability for Engineers*. Third Edition, John Wiley and Sons, Inc.
- [28] The R Project for Statistical Computing, www.r-project.org/.
- [29] Banco Central del Ecuador <http://www.bce.fin.ec/>.
- [30] Clark Todd E. (1995), *Do Producer Prices Lead Consumer Prices?*. Economic Review Third Quarter 1995.
- [31] Instituto Nacional de Estadísticas y Censos, Índice de Precios al Consumidor http://www.inec.gob.ec/web/guest/ecu_est/est_eco/ind_eco/ipc.
- [32] Instituto Nacional de Estadísticas y Censos, Índice de Precios al Productor http://www.inec.gob.ec/web/guest/ecu_est/est_eco/ind_eco/ipp.
- [33] Superintendencia de Bancos y Seguros (2003), *Normas Generales para la Aplicación de la Ley General de Instituciones del Sistema Financiero*. Libro I, Título X, Capítulo IV.

- [34] Cajamarca Rodrigo y Mena Hermann (2011), *Modelación de series económicas mediante métodos de regresión difusa*. Preprint.
- [35] Cajamarca Rodrigo y Mena Hermann (2011), *Métodos automáticos de regresión difusa - SCRIPT mard para R*. <http://www.math.epn.edu.ec/hmena/>, sección Software.