UNIVERSIDAD MICHOACANA
DE SAN NICOLÁS DE HIDALGO
Cuna de héroes, crisol de pensadores

UNIVERSIDAD MICHOACANA DE SAN NICOLÁS DE HIDALGO

Facultad de Ingeniería Eléctrica

División de Estudios de Posgrado

MODELOS DIFUSOS PARA PRONÓSTICOS DE SERIES DE TIEMPO

TESIS

Que para obtener el grado de

MAESTRO EN CIENCIAS EN INGENIERÍA ELÉCTRICA

Presenta

Juan de Dios Pelayo Gómez

Dr. en Ciencias Computacionales Juan José Flores Romero

Director de Tesis

Morelia, Michoacán Octubre 2017

MODELOS DIFUSOS PARA PRONÓSTICOS DE SERIES DE TIEMPO

Los Miembros del Jurado de Examen de Grado aprueban la **Tesis de Maestría en Ciencias en Ingeniería Eléctrica** de ***Juan de Dios Pelayo Gómez***

Dr. José Antonio Camarena Ibarrola
Presidente del Jurado

Dr. Juan José Flores Romero
Director de Tesis

Dr. Félix Calderón Solorio
Vocal

Dr. Jaime Cerda Jacobo
Vocal

Dr. Mario Graff Guerrero
Revisor Externo (INFOTEC)

Dr. Félix Calderón Solorio
Jefe de la División de Estudios de Posgrado de la Facultad de Ingeniería Eléctrica. UMSNH (Por reconocimiento de firmas).

UNIVERSIDAD MICHOACANA DE SAN NICOLÁS DE HIDALGO
Septiembre 2017

Dedicatoria

Dedico este trabajo especialmente a Dios por darme la vida, la inteligencia y las capacidades necesarias para poder culminarlo. Agradezco todas sus bendiciones. Porque es gracias a Él que todo esto fue posible.

A mi mamá por su amor y ayuda constante, admiro su fortaleza y decisión, con las cuales me ha motivado a estudiar. Agradezco todo el esfuerzo que ha dedicado en mi preparación, desde llevarme al escuela, preparar mis alimentos, aconsejarme, llamarme la atención y siempre darme esperanza. Por todo su cariño y por cambiar mi vida al hacerme estudiar. A mi papá por su ejemplo de trabajo y esfuerzo, por su gusto por aprender y pensar. También por impulsarme a ser más crítico, por sus consejos y apoyo. Agradezco sus enseñanzas e interés por que tenga un futuro mejor. Por su cariño y compañía. Si explico todas las razones en las que me han ayudado nunca terminaría de escribir la tesis.

A mi hermano Salvador Daniel, por siempre ayudarme y explicarme con paciencia. Agradezco su compañía a lo largo de mis estudios de maestría. Por hacer que las horas de estudio se vuelvan bastante divertidas y también por ayudarme a hacer mi trabajo sin estresarme. Gracias por llevarme a la casa casi siempre.

A mis hermanas Cristina, Ana Laura y Lucy Amor, por alegrarme cada día y ser tan compresivas. Agradezco su apoyo y confianza en mí, que siempre me motivaron y animaron. Las tres son mi tercera hermana favorita.

A mis hermanos José de Jesús, Emmanuel, David, Diego y Misael por siempre ayudarme a ser mejor pero al mismo tiempo recordarme que debo relajarme. Agradezco su apoyo e interés en lo que hago. Agradezco que también ellos estudien y se esfuercen ya que así me motivan a no dejar de hacerlo. También agradezco que ya no me van a decir que para cuando terminaré la tesis.

A mis sobrinos Yoshua, Joseph y Alán, porque aún sin saberlo contagian su alegría e inocencia. Agradezco que siempre me hacen pensar en darles un buen ejemplo. Agradezco

que ellos no van a notar mis faltas de ortografía.

Al Doctor Juan Flores, por todos sus consejos, por su interés en que aprendiera, por que me facilitó bastante el proceso de aprendizaje y por ser paciente. Agradezco la oportunidad que me brindó al ser mi asesor. Al Dr. Félix Calderón por ayudarme a ver el estudio de una manera más profesional.

A mis amigos, Adán, por compartir su conocimiento y recomendaciones; y por las bromas que son demasiado divertidas, por su amistad. A Juan por siempre dar ánimos y por su amistad, a Hugo, Felipe, César, Rodrigo, Bryan y Javier por su amistad. También por invitarme a las retas de fútbol aunque casi nunca iba.

A Rafael Cedeño por siempre orientarme cuando ya no sabía como seguir tanto en el desarrollo como en la redacción. Ya no lo molestaré con tantas preguntas.

Al Conacyt por otorgarme la beca que me permitió realizar estos estudios. A la universidad que ha sido mi casa durante 10 años. A toda la comunidad de la División de Estudios de Posgrado de la Facultad de Ingeniería Eléctrica.

Contenido

Lista de Términos

ACF función de autocorrelación.

Algoritmo de k-medias Método de agrupamiento que particiona una colección de observaciones en k grupos, en el que cada observación se almacena en el grupo cuya media es más cercana al valor de la observación.

ANFIS red adaptativa basada en un sistema de inferencia difusa (en inglés Adaptative Network Based Fuzzy Inference System).

ANN red neuronal artificial (en inglés Artificial Neural Network).

AR autorregresivo.

ARIMA autoregresivo integrado de media móvil, (en inglés Autoregressive Integrated Moving Average).

ARMA autorregresivo de media móvil (en inglés Autoregressive Moving Average).

ARMAX autorregresivo de media móvil con términos exógenos (en inglés Autoregressive Moving Average Exogenous).

BD datos masivos, (en inglés Big-data).

Big data Término usado para denotar a toda colección de conjuntos de datos tan grandes o complejos que se vuelve complicado el procesarlos empleando aplicaciones tradicionales de procesamiento de datos.

CDF función de distribución acumulativa (en inglés Cumulative Distribution Function).

Clúster Se aplica a los conjuntos o conglomerados de computadoras construidos mediante la utilización de hardwares comunes y que se comportan como si fuesen una única computadora.

COA centro de área (en inglés Center of Area).

COG centroide o centro de gravedad (en inglés Center of Gravity).

COS centro de los conjuntos (en inglés Center of Sets).

DE optimización por evolución diferencial (en inglés Diferencial Evolution).

ECG electrocardiograma.

EEG electroencefalograma.

EMG electromiograma.

Feedforward Se denomina así a redes neuronales artificiales que se propagan en un sólo sentido, es decir, son redes neuronales que no tienen retroalimentación.

FF pronóstico difuso (en inglés Fuzzy Forecast).

FL aprendizaje difuso (en inglés Fuzzy Learning).

FNN falsos vecinos cercanos (en inglés False Nearest Neighbors).

GA algoritmos genéticos (en inglés Genetic Algorithms).

GARCH autorregresivo general condicional con heteroscedasticidad (en inglés General Autoregressive Conditional Heteroscedasticity).

Gb Medida de capacidad de almacenamiento de un dispositivo digital que equivale a mil millones de bytes.

GHz Unidad de medida de frecuencia, que equivale a 1×10^9 Hertz.

HyFIS sistema hybrido de inferencia difusa (en inglés Hybrid Fuzzy Inference System).

ICDF función de distribución acumulativa inversa (en inglés Inverse Cumulative Distribution Funtion).

iid Independiente e idénticamente distribuida.

LOM mayor de los máximos (en inglés Largest of Maximum).

m dimensión de embebido.

M_{DF} marcador de datos faltantes.

MA media móvil.

MAPE error porcentual absoluto medio (en inglés Mean Absolute Porcentage Error).

MOM media del máximo (en inglés Middle of Maximum).

MSE error cuadrático medio (en inglés Mean Square Error).

n número de pronósticos.

N longitud de la serie de tiempo.

NAÏVE Hace referencia a un enfoque de pronóstico usado en series de tiempo que calcula la predicción de un dato como el valor del dato que lo precede. Toma este nombre de la palabra naive que en español se traduce como ingenuo o sencillo, haciendo alusión a que es la forma más simple que se puede pensar para pronosticar.

NC número de conjuntos difusos.

NEFPROX sistema neuro-difuso de aproximación de funciones (en inglés Neuro Fuzzy Systems for Function Approximation).

NN algoritmo de pronóstico no lineal basado en vecinos cercanos.

NNDE algoritmo de pronóstico no lineal basado en vecinos cercanos optimizado con evolución diferencial.

NTD dolár de Taiwan.

ODA un día a futuro (en inglés One Day Ahead).

OSA un paso a futuro (en inglés One Step Ahead).

PACF función de autocorrelación parcial.

PSO optimización por enjambre de partículas (en inglés Particle Swarm Optimization).

RAM memoria de acceso aleatorio (en inglés Random Access Memory).

RCGA algoritmos genéticos de codificación real (en inglés Real Coding Genetic Algorithms).

Ruido blanco Gaussiano Se denomina así a una señal que sigue una distribución normal con media ($\mu = E()$) y varianza (σ^2) determinadas, en la cual no existe correlación estadística entre dos instantes.

S_{CV} selector de conjuntos variables.

S_{DF} selector de datos faltantes.

S_{EP} selector de enfoque de pronóstico.

S_I selector de intersección.

S_{TC} selector de todos los conjuntos.

S_{TR} selector de todas las reglas.

SMAPE error porcentual absoluto medio simétrico (en inglés Symetric Mean Absolute Porcentage Error).

SOM menor de los máximos (en inglés Smallest of Maximum).

t_a tiempo de aprendizaje.

t_p tiempo de pronóstico.

τ tiempo de retardo.

USD dolár de Estados Unidos.

X serie de tiempo.

$\hat{X}_{N+k}$ pronósticos a futuro.

Lista de Figuras

Lista de Tablas

Resumen

El problema a resolver en esta tesis es estimar uno o varios valores a futuro, de una variable de interés obtenida de un sistema caótico, cuyo pasado se almacena en forma de una serie de tiempo. La naturaleza caótica contenida en algunas series de tiempo (generadas a partir de sistemas reales) complica el problema de pronóstico. Adicionalmente, se considera que las mediciones almacenadas pueden tener un nivel de ruido elevado, contener datos atípicos y también faltantes.

Con la finalidad de obtener estas estimaciones a futuro se plantea el algoritmo de pronóstico difuso (en inglés Fuzzy Forecast) (FF). La idea general es que situaciones actuales en la serie de tiempo pueden parecerse a situaciones del pasado. Entonces los valores a futuro pueden ser estimados basándose en los siguientes valores de esos casos similares. El algoritmo planteado extrae la información relevante de una serie de tiempo por medio de vectores de retardo. Esta información se convierte en un conjunto de reglas difusas y mediante un proceso de inferencia se obtienen los pronósticos.

En base a los resultados obtenidos se observa que las principales ventajas de este modelo de pronóstico, son: i) presenta una gran eficiencia con respecto al tiempo de ejecución para realizar la tarea de aprendizaje (construcción de la base de reglas), ii) puede procesar grandes cantidades de datos, iii) es un modelo incremental, lo cual implica que no necesita repetirse la etapa de aprendizaje para incorporar nueva información, iv) el modelo es robusto ante datos faltantes, ignorando los vectores de retardo que no están completos, v) puede operar con datos atípicos, distribuyendo los conjuntos difusos de manera que estos datos tengan la menor influencia posible y vi) este modelo es versátil, o sea, puede trabajar con series de tiempo de diversas índoles. Los resultados obtenidos, en cuanto a precisión, lo colocan en un punto medio con respecto a los métodos con los que se compara. Esto se debe principalmente a que no puede manejar adecuadamente el ruido en las mediciones y que los datos atípicos le afectan considerablemente. Aunque su desempeño no es deficiente, se debe buscar obtener errores de predicción menores. En contraparte FF se posiciona como el método más eficiente en cuanto al tiempo de aprendizaje. Por otro lado, los resultados con datos masivos muestran que FF puede procesar fácilmente $1,000,000$ de datos en poco tiempo (cerca de 23 minutos).

Palabras clave: Series de tiempo, Pronóstico, Lógica difusa, Big-data, Vectores de retardo.

Abstract

The problem of solving in this thesis is to estimate one or several future values of a variable of interest obtained from a chaotic system, whose past is stored as a time series. The chaotic nature contained in some time series generated from real systems dificults the forecasting problem. Additionally, it is considered that the measurements stored may have a high noise level, contain outliers and also missing data.

In order to obtain these future estimates we propose a forecast algorithm based on fuzzy logic (FF). The general idea is that current situations in the time series may be resembling past situations. Then future values can be estimated based on the following values of these cases. The raised algorithm extracts the relevant information from a times series by means of delay vectors. This information becomes a set of fuzzy rules and throungh an inference process you get the forecasts.

Based on the results obtained it is observed that the main advantages of this forecast model are: i) has a great effectiveness with the execution time to perform the learning task (construction of the rule base), ii) it can process large amounts of data, iii) it is a incremental model, which implies that the learning stage does not need to be repeated to incorporate new information, iv) the model is robust to missing data, ignoring the delay vectors that are not complete, v) it can operate with outliers data, distributing the fuzzy sets so that these data would have the least contribution and vi) this model is versaltile, or sea, can work with time series of various kinds. The results obtained, in terms of precision, place it at a medium point with respect to the methods with which it is compared. This is mainly because it can not adequately handle the noise in the measurements and that the outliers data affects it considerably. Although your performance is not deficient, it should look obtain lower forecast errors. In contrast, ac FF is positioned as the most efficient method in terms of the learning time. On the other hand, the results with massive data can easily process $1,000,000$ of data in a short time (about 23 minutes).

Capítulo 1

Introducción

"Todos los modelos son malos, algunos son útiles". George E. P. Box, Estadístico Británico.

En diversas áreas de la ciencia y la tecnología es necesario conocer el comportamiento y los cambios que presentan ciertas magnitudes derivadas de un sistema o proceso específico. Para poder apreciar adecuadamente la forma en la que cambian las magnitudes de interés, éstas se deben medir y almacenar, por lo general en base al tiempo. Las razones para realizar lo anterior son muy variadas, aunque principalmente se hace para aplicar algún tipo de control sobre el sistema o sus variables, o bien, para intuir de alguna manera el futuro del proceso o la evolución que tendrán las magnitudes relevantes.

En el caso donde el interés principal es anticipar el comportamiento futuro de alguna o algunas magnitudes en particular, la finalidad es tomar decisiones en el instante actual tratando de prevenir lo que sucederá después. En algunos casos tratando de evitar que se lleguen a situaciones adversas en el sistema, por ejemplo, si la magnitud que se está analizando es la humedad en un invernadero, sería conveniente saber si en algún momento esta podría superar los límites donde los cultivos morirían. En otros casos, podría ser tratando de evaluar si la variable en cuestión tendrá un comportamiento del que se pueda obtener algún provecho; un ejemplo muy claro de esta situación es tratar de conocer a futuro el tipo de cambio entre dos monedas, digamos entre el euro y el dólar. Si se tuviera información que

sugiere que es buena idea comprar dolares en este momento, se cambiaría alguna suma en euros por dólares, ya que posteriormente se podrán vender a un valor superior al adquirido.

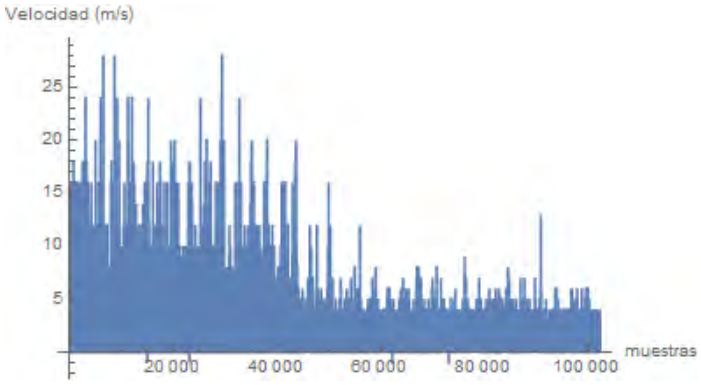
La diferencia sustancial entre realizar estimaciones futuras y control radica en como se lleva a cabo la toma de decisiones. Supongamos que se mide el nivel de líquido de una presa y se desea mantenerlo siempre inferior al 75 por ciento de la capacidad total. Un control en el sentido convencional sería medir el nivel de liquido de la presa y como acción de control abrir en mayor o menor grado las compuertas. Si ocurre que en un periodo considerable de tiempo llueve abundantemente, el control convencional se limitaría a drenar el agua excedente para el correcto funcionamiento. Al utilizar algún modelo de pronóstico, podría tenerse información que sugiera que debido la intensidad de las lluvias sin importar las acciones de control tomadas hasta ese momento se superarán las tres cuartas partes de la capacidad de la presa. Esto llevaría a buscar otros mecanismos de drenado o bien a utilizar los ya existentes de otra manera para evitar el desbordamiento.

Si se busca conocer el futuro del comportamiento de alguna magnitud es necesario contar con su historial de mediciones, esto da lugar al estudio de las series de tiempo. Formalmente una serie de tiempo es una secuencia de observaciones de una variable de interés ordenada cronológicamente, donde la variable es recopilada a intervalos de tiempo igualmente espaciados y se representa de manera general como se muestra en (1.1) [Douglas C. Montgomery, 2008].

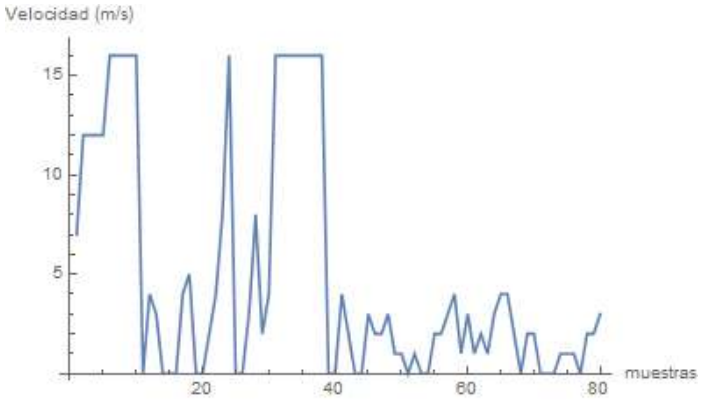
$$X = \{X_1, X_2, X_3, \ldots, X_N\} \tag{1.1}$$

donde N es la longitud de la serie de tiempo. Normalmente la serie de tiempo se denota como X y su i-ésima muestra como X_t, para $i \in \mathbb{N}$, donde $t = t_0 + (i-1)T$. El periodo de muestreo T (tiempo que transcurre entre tomar dos muestras adyacentes) está dado en segundos, en tanto que t_0 representa el tiempo inicial (en segundos) a partir del cual se comienzan a tomar mediciones.

Las series de tiempo se pueden obtener de ámbitos muy diversos. En la Figura 1.1(a) se observa la representación gráfica de una serie de tiempo de velocidad de viento y en la Figura 1.1(b) se presentan sus primeras 80 muestras. A continuación se mencionan las áreas más relevantes en las que se requiere el uso de series de tiempo.



(a) ***Serie completa*** *(*$102,272$ ***muestras****)*



(b) ***Primeras*** 80 ***muestras***

Figura 1.1: Serie de tiempo de velocidad de viento

- Industria e Ingeniería.- En la industria el control de procesos requiere estar preparado para posibles cambios que alteren la estabilidad y el funcionamiento óptimo del proceso; además también es necesario determinar condiciones seguras de operación. Por lo tanto, es común realizar pronósticos de las variables de mayor relevancia dependiendo del tipo de proceso. En la industria química se puede aplicar en variables tales como: presión, temperatura, calor, humedad y energía, entre otros. En la industria eléctrica se puede usar para predecir: potencia, voltaje, corriente, carga y demanda eléctrica. En la parte de generación: radiación solar, velocidad del viento, caudal del agua, temperatura, capacidad de generación, etcétera. En la ingeniería civil se puede usar como herramienta para predecir sismos, daños a estructuras, hacer estudios de

suelo o estudios en presas [Douglas C. Montgomery, 2008] y [Robert H. Shumway, 2011].

- Economía y Finanzas.- En el sector económico es donde puede tener mayor relevancia el obtener pronósticos, algunos casos financieros son rendimiento de inversiones como bonos, materias primas, acciones, previsiones de tipos de interés y de cambio de moneda. Instituciones financieras, organizaciones políticas y gobiernos están interesados en pronosticar variables como crecimiento del PIB, tasas de interés, inflación, producción y consumo [Douglas C. Montgomery, 2008] y [Robert H. Shumway, 2011].

- Demografía.- Las previsiones de población se hacen de manera rutinaria, midiendo y tratando de pronosticar al mismo tiempo, nacimientos, defunciones, migración, crecimiento de la población, crecimiento laboral, desempleo, con la finalidad de planificar servicios sociales, de salud, jubilaciones e incluso el lanzamiento de nuevos productos y tipos de servicios [Douglas C. Montgomery, 2008] y [Robert H. Shumway, 2011].

- Ciencias Medicas y Biológicas.- En el control de enfermedades puede usarse para conocer los casos registrados de determinadas enfermedades que podrían convertirse en epidemias. Prácticamente cualquier variable del cuerpo puede generar una serie de tiempo y dar nueva información para anticiparse al desarrollo de enfermedades. Algunos ejemplos son: la presión sanguínea, el ritmo cardiaco, niveles glucosa, minerales, etcétera. En este ambito, tienen una importancia especial las señales obtenidas a partir de procedimientos como: electroencefalograma (EEG), electromiograma (EMG) y electrocardiograma (ECG) [Douglas C. Montgomery, 2008] y [Robert H. Shumway, 2011].

1.1. Definición del problema

El problema que se requiere resolver es el siguiente. Supóngase que se cuenta con una serie de tiempo caótica que tiene la forma $X = \{X_1, X_2, X_3, \ldots, X_N\}$, la cual además puede tener presencia de datos atípicos, un alto nivel de ruido en las mediciones e incluso datos faltantes. Dados la dimensión de embebido (m), el tiempo de retardo (τ) y el número

de pronósticos (n) que se desean obtener, se deben calcular los pronósticos a futuro ($\hat{X}_{N+k}$). Estos se determinan exclusivamente a partir de la serie de tiempo y en algunos casos de los pronósticos previos a X_{N+k}, considerando que $k = 1, 2, \ldots, n$.

En esta tesis se ha desarrollado un sistema de pronóstico basado en lógica difusa y vectores de retardo. El cual contendrá un algoritmo de aprendizaje o entrenamiento y otro de pronóstico. En las subsecciones siguientes se explica de manera concisa con qué características se diseñó el sistema de pronóstico y de manera muy general la razón de elegir la lógica difusa para llevar a cabo este sistema.

1.1.1. Características deseables en un sistema de pronóstico

En la actualidad existe una gran cantidad de métodos y técnicas utilizadas para realizar pronósticos, sin embargo, hay propiedades deseables en cualquier modelo que se use para pronosticar. Al desarrollar este trabajo se pretende atender a estas características.

Debido a que las series de tiempo tienen un origen muy diverso, es atractivo que exista un modelo versátil que se pueda aplicar, con ciertas adecuaciones, indistintamente a series de tiempo de magnitudes físicas, económicas, biológicas, médicas, demográficas, entre otras. Algunos ejemplos de series de tiempo caóticas son la velocidad del viento, el tipo de cambio, un sistema de pendulo doble, entre otras.

Como segundo punto a considerar se busca que el modelo sea fácil de utilizar por parte de los usuarios, es decir, que no requiera de un conocimiento profundo de la teoría que encierra la creación de este modelo, pero al mismo tiempo que dé la posibilidad al usuario de conceptualizar de manera general el proceso de estimación, en este sentido se espera un modelo intuitivo.

Por otro lado es necesario que el modelo presente un desempeño aceptable en precisión, comparado con otros modelos existentes, es decir, que sea un modelo competitivo en precisión en el ámbito que se le utilice. Se requiere un modelo robusto ante las diferencias entre una aplicación y otra.

También se debe considerar la posibilidad de que el modelo sea eficiente en el tiempo que tarda en entregar los resultados, así como en el consumo de recursos para generar los pronósticos. Se espera que el algoritmo encargado de realizar el aprendizaje

tenga una complejidad computacional baja tanto en el uso de memoria como de tiempo.

Se busca adicionalmente que la cantidad de datos a procesar no sea una limitante para el modelo, en otras palabras se espera que el modelo trabaje adecuadamente con datos masivos, (en inglés Big-data). Esto es bastante deseable ya que permitiría realizar una única etapa de aprendizaje y por otro lado aportaría más información para los pronósticos. De la misma manera el modelo de pronóstico debe tener alguna forma de adaptarse e incorporar la información que ofrecen los nuevos datos que vayan midiéndose. Es decir, que sea un modelo incremental, el cual permitiría seguir obteniendo información conforme transcurra más tiempo, sin la necesidad de volver a realizar todo el proceso de aprendizaje. Un modelo incremental incorpora información nueva a la base de conocimiento, sin desechar la anterior y sin la necesidad de repetir la fase de aprendizaje.

Un sistema de pronóstico que pueda manejar datos masivos y al mismo tiempo que sea incremental puede trabajar indefinidamente y en tiempo real, procesando los nuevos datos recopilados e incorporandolos a su conocimiento a la par que genera pronósticos. Con esto se espera que se puedan hacer estimaciones del futuro con todo el pasado de la variable de interés y que cada situación anterior tenga una cierta contribución en la medida que se asemeje a la situación actual.

Finalmente se aspira a que el modelo pueda realizar pronósticos a diferentes horizontes de predicción. Esto permitiría que el modelo se pueda adaptar aún más a un problema específico sin dejar por eso de usarse en otros de manera similar. Lo anterior también favorece que se puedan hacer multiples análisis de la variable de interés sin necesidad de cambiar de modelo.

1.1.2. Motivación para usar lógica difusa

La lógica difusa ha tenido un gran impacto y aceptación en entornos donde se tiene conocimiento vago, impreciso, incierto, ambiguo, inexacto, o probabilístico por naturaleza. Trata de emular el pensamiento humano que frecuentemente conlleva información de este tipo, originada de la inexactitud inherente de los conceptos humanos y del razonamiento basado en experiencias anteriores similares pero no idénticas. En este sentido las series de tiempo y su pronóstico son un problema donde la información disponible puede presentar

esas características. De manera natural el ser humano usa su propia experiencia para intuir el futuro y al mismo tiempo para condensar la información que posee. Resulta bastante interesante trasladar esa forma de estimar lo que sucederá (partiendo de la información previa) de los seres humanos a un sistema de pronóstico autonomo. Además, de alguna manera, podría absorber el problema de definir límites razonables para la magnitud analizada. En una etapa de entrenamiento del sistema, la información contenida en la serie de tiempo podría servir para que el sistema clasifique las mediciones y al llegar una nueva determine a cuales situaciones anteriores se parece y en que grado.

1.2. Objetivos

A continuación se menciona el objetivo general, así como los objetivos específicos de esta tesis.

1.2.1. Objetivo general

Implementar un modelo basado en lógica difusa que pueda realizar pronósticos de series de tiempo. Este algoritmo debe transformar la información relevante de las series de tiempo en una base de reglas de inferencia utilizando relaciones difusas. Estas reglas sirven para comparar el grado de relación que existe entre las situaciones actuales y las anteriores y ponderarlas de manera que los pronósticos para el instante actual, se generen a partir de las situaciones pasadas que tengan mayor parecido.

1.2.2. Objetivos específicos

Los principales propósitos al desarrollar el presente trabajo se resumen de la siguiente manera:

- Buscar en la literatura actual trabajos donde se realizan pronósticos de series de tiempo usando de alguna forma los conceptos y teoría de lógica difusa.
- Desarrollar un mecanismo para convertir las series de tiempo en un conjunto de vectores de retardo que contengan la información relevante.

- Realizar un mapeo de la información contenida en los vectores de retardo en una base de reglas de inferencia difusa.

- Implementar un sistema de pronóstico que use la información actual y la base de reglas difusas para realizar estimaciones a uno y a n pasos a futuro.

- Realizar pronósticos en varias series de tiempo que presenten comportamiento caótico y observar su desempeño para los diversos casos de estudio.

- Llevar a cabo comparaciones de precisión utilizando algunas medidas de errores existentes para tal fin, así como también tiempo de ejecución. Estas comparaciones son aplicadas en algunos modelos de pronóstico de series de tiempo, incluyendo alguno que específicamente se use en series caóticas y el presentado en este proyecto.

- Diseñar un experimento que permita observar a grandes rasgos el comportamiento que el modelo presenta cuando se trabaja con Big data.

1.2.3. Justificación

Las principales razones para realizar este trabajo son que no existen modelos de pronóstico de series de tiempo que utilicen la lógica difusa de manera pura; la gran mayoría son híbridos entre otras técnicas y lógica difusa. Parece conveniente aplicar las ventajas que ofrece la lógica difusa en el problema de pronóstico, de esta manera se podría permitir a un modelo de pronóstico difuso aprender de manera similar a la forma en que razonamos los seres humanos, permitiría modelar de alguna manera la tendencia general de la series de tiempo.

La segunda razón que impulsa la implementación de este proyecto es que actualmente los modelos de pronósticos de series de tiempo basados en inteligencia artificial, tienden a ser muy costosos computacionalmente hablando, tanto en tiempo de procesamiento como en memoria requerida. Los modelos simples llegan a ser insuficientes en la precisión. Se pretende que el enfoque dado en este caso busque un punto intermedio entre ambas situaciones.

Otro hecho que motiva la realización de esta tesis es que existen pocos modelos que sean incrementales, es decir, que conforme se tengan nuevos datos, estos también aporten a la base de conocimiento, pero sin la necesidad de repetir el proceso de entrenamiento. Es muy común, por ejemplo en redes neuronales, que para incorporar nuevos datos al modelo, se tenga que reentrenar por completo.

Por último, un punto a considerar es que los modelos de pronóstico existentes tienden a ser muy limitados cuando se manejan datos masivos, se busca que el modelo implementado tenga una respuesta aceptable al procesar datos masivos.

1.3. Descripción de capítulos

Esta tesis se organiza de la siguiente manera, en el Capítulo 1 se aborda a grandes rasgos el pronóstico de series de tiempo, posteriormente se explica que problemas se tienen al realizar pronósticos y como esta metodología pretende darles solución. Después se mencionan los objetivos que se pretenden alcanzar durante el desarrollo del proyecto, adicionalmente se exponen de manera breve las razones principales que motivan que se lleve a cabo.

En el Capítulo 2, se muestran los trabajos que se han realizado para pronóstico usando lógica difusa, y las diferencias sustanciales con esta implementación. Aquí se abordan las series de tiempo difusas y las aplicaciones principales que han tenido. Posteriormente se exponen las tecnicas de pronóstico que están basadas en lógica difusa. Abordando los modelos que combinan la red neuronal artificial (en inglés Artificial Neural Network) (ANN) con la lógica difusa, después los mapas cognitivos difusos y finalmente los modelos de regresión difusos.

En el Capítulo 3, se presenta el estudio de las series tiempo y los modelos más comunes que existen para realizar pronósticos, teniendo, en consideración las series de tiempo caóticas. Se mencionan los modelos que comunmente se usan para pronosticar en series de tiempo que son lineales, finalmente se hace mención de métodos que se usan en series de tiempo no lineales, o bien, que presentan dificultades para estimar el futuro.

El Capítulo 4 se enfoca en la lógica difusa, desde su uso (aplicaciones), las diferen-

cias con la lógica clásica, la teoría de conjuntos, etcétera. Se abordan las operaciones que se realizan con dichos conjuntos y posteriormente los mecanismos de razonamiento y las implicaciones lógicas que son usadas para manipular el conocimiento dentro de este ámbito. En seguida se explican los métodos de defuzzificación más conocidos, haciendo énfasis en el método del centroide y su simplificación el método del centro de los conjuntos.

El Capítulo 5 explica detalladamente la metodología a seguir para desarrollar el modelo de pronóstico difuso, los algoritmos implementados, las características del modelo programado así como los problemas presentados al desarrollar el método.

El Capítulo 6 muestra las pruebas realizadas al sistema de predicción difuso, presentado comparaciones con algunos modelos representativos tales como: el autoregresivo integrado de media móvil, (en inglés Autoregressive Integrated Moving Average) (ARIMA), las ANN y algoritmo de pronóstico no lineal basado en vecinos cercanos (NN). Como segunda parte se evalúa el desempeño del modelo cuando se trabaja con datos masivos y se presentan comparaciones entre diferentes cantidades de datos para algunas series de tiempo.

En el Capítulo 7 se mencionan las conclusiones a las que se llegó después del desarrollo de la tesis y también se dan una serie de recomendaciones al usar este sistema. Finalmente se habla de manera general de las mejoras que pueden hacerse, las pruebas que sería buena idea adicionar y las posibles áreas de oportunidad a investigar en este sentido.

Conclusiones del capítulo

Hasta este punto se ha explicado el problema que se resuelve en esta tesis y las principales razones para abordar el problema de pronóstico de series de tiempo usando lógica difusa. También se mencionaron los objetivos que se han de cumplir durante el desarrollo de la tesis y finalmente se dió una descripción general del contenido de cada Capítulo.

Capítulo 2

Antecedentes y estado del arte

En esta Capítulo se mencionan los trabajos previos donde se usa la lógica difusa para el pronóstico de series de tiempo, la forma en como se han realizado estimaciones con modelos difusos se divide en dos grandes grupos. En el primero las series de tiempo en sí se consideran de naturaleza difusa, es decir, cada elemento de la serie se expresa de manera difusa. El segundo grupo considera las series de tiempo en su forma convencional, en tanto que el modelo de pronóstico es el que utiliza la lógica difusa para cumplir con su finalidad. Este último contiene técnicas de pronóstico que incorporan en su implementación la lógica difusa, pero no son puramente modelos difusos, como antecedentes de modelos difusos puros se menciona la perspectiva de los mapas cognitivos difusos.

2.1. Series de tiempo difusas y pronóstico

[Song und Chissom, 1993b] introducen el concepto de series de tiempo difusas, de la siguiente manera, definen una serie de tiempo (X), que es un subcojunto de $\mathbb{R}$ y contiene términos X_t, para $t \in \mathbb{N}$; como el universo de discurso en el cual los conjuntos difusos A_i son definidos para $i \in \mathbb{N}$ y A es la colección de los A_i. Luego A es llamada una serie de tiempo difusa en X.

La diferencia principal entre una serie de tiempo convencional y una difusa es que en las primeras los datos son números y en las segundas son términos lingüísticos o conjuntos difusos. Las series de tiempo difusas son utilizadas en ambientes ambiguos o

cuando los datos fueron recopilados de esa manera.

Un ejemplo para mostrar en que casos son usadas es el siguiente, imaginemos que se observa el clima en cierta región, bien podrían realizarse mediciones en alguna escala de temperatura, pero también se puede usar el lenguaje común para describir de alguna manera el comportamiento del clima, podríamos usar términos como bueno, muy bueno, caliente, muy frío, algo caliente o algo frío, entre otros. Es importante notar dos situaciones, la primera que para diferentes días existen diferentes rangos de temperatura y la segunda, que se usan términos distintos para describir el clima en días diferentes. En un día de verano utilizaremos palabras como caliente, muy caliente, etcétera; mientras que en invierno cambiaremos por frío, muy frío, algo frío, pero difícilmente se utilizará el término caluroso en invierno. De esta forma, termina siendo evidente que puede haber series de tiempo que de manera natural tengan asociados términos lingüísticos o conjuntos difusos. Claramente estos tipos de series de tiempo difieren de las convencionales y no pueden ser utilizados los mismos modelos [Song und Chissom, 1993b].

Considerando el enfoque de las series de tiempo difusas se han hecho trabajos relacionados con el pronóstico que abarcan apliaciones muy diversas. En [Song und Chissom, 1993a] se aplican las series de tiempo difusas en el problema de pronóstico de inscripciones en la Universidad de Alabama; usan datos recopilados desde $1,971$ a $1,990$, donde el máximo registrado fue de $19,328$ y el mínimo de $13,055$, su espacio de trabajo lo definen desde $13,000$ hasta $20,000$ y lo dividen en 7 intevalos de la siguiente manera: $u_1 = (13,000-14,000), u_2 = (14,000-15,000), u_3 = (15,000-16,000), u_4 = (16,000-17,000), u_5 = (17,000-18,000), u_6 = (18,000-19,000), u_7 = (19,000-20,000)$, posteriormente definen también 7 conjuntos difusos, "no many","no too many","many","many many","very many","too many","too many many", los cuales tienen las funciones de pertenencia siguientes, respectivamente:

$$\mu_{A_1}(X) = \begin{cases} 1 \text{ para } X \in u_1 \\ 0.5 \text{ para } X \in u_2 \\ 0 \text{ otro caso} \end{cases}$$

$$\mu_{A_2}(X) = \begin{cases} 1 \text{ para } X \in u_2 \\ 0.5 \text{ para } X \in u_1 \text{ o} X \in u_3 \\ 0 \text{ otro caso} \end{cases}$$

$$\mu_{A_3}(X) = \begin{cases} 1 \text{ para } X \in u_3 \\ 0.5 \text{ para } X \in u_2 \text{ o} X \in u_4 \\ 0 \text{ otro caso} \end{cases}$$

$$\mu_{A_4}(X) = \begin{cases} 1 \text{ para } X \in u_4 \\ 0.5 \text{ para } X \in u_3 \text{ o} X \in u_5 \\ 0 \text{ otro caso} \end{cases}$$

$$\mu_{A_5}(X) = \begin{cases} 1 \text{ para } X \in u_5 \\ 0.5 \text{ para } X \in u_4 \text{ o} X \in u_6 \\ 0 \text{ otro caso} \end{cases}$$

$$\mu_{A_6}(X) = \begin{cases} 1 \text{ para } X \in u_6 \\ 0.5 \text{ para } X \in u_5 \text{ o} X \in u_7 \\ 0 \text{ otro caso} \end{cases}$$

$$\mu_{A_7}(X) = \begin{cases} 1 \text{ para } X \in u_7 \\ 0.5 \text{ para } X \in u_6 \\ 0 \text{ otro caso} \end{cases}$$

Luego plantean de manera matricial las relaciones difusas existentes entre los términos usando el modelo de primer orden y la definición de serie difusa invariante en el tiempo, definiciones 3 y 10 de [Song und Chissom, 1993b] respectivamente. Lo anterior genera relaciones del tipo $A_1 \rightarrow A_1, A_1 \rightarrow A_2, ..., A_2 \rightarrow A_1, ...$ (aunque no necesariamente existen todas las combinaciones); con esto llegan a un modelo $A_t = A_{t-1} \bigcirc R$, donde A_t es la cantidad de inscripciones en el año actual (de manera difusa), A_{t-1} es la cantidad de ins-

cripciones en el año anterior y $\bigcirc R$ representa la composición max-min de las relaciones difusas, expresada de manera matricial. Finalmente hacen una comparación de su modelo con uno de regresión lineal y según los autores obtienen un desempeño mejor en trece de los veinte años utilizados.

En [Song und Chissom, 1994] presentan una continuación del trabajo hecho en [Song und Chissom, 1993a], donde definen más formalmente el trabajo previo y la diferencia sustancial es que utilizan una serie difusa variante con el tiempo, definición 10 de [Song und Chissom, 1993b]. Posteriormente agregan como métodos para convertir sus valores difusos en términos numéricos el método del centroide y una red neuronal. Los resultados obtenidos los comparan con los de la versión anterior. Posteriormente en [Chen, 1996], se presenta una modificación al método usado en [Song und Chissom, 1994], donde después de obtener las relaciones difusas de los términos $A_1 \rightarrow A_1, A_1 \rightarrow A_2, ...$; éstas se clasifican en grupos, que contienen todos los posibles conjuntos difusos que estan enseguida para cada conjunto difuso. Después siguiendo criterios preestablecidos hacen el cálculo de los pronósticos, utilizando operaciones aritméticas más simples para calcular la composición de las relaciones. En [Yu, 2005] se presenta una modificación a lo mostrado por [Chen, 1996] donde se pondera las veces que una relación difusa ha aparecido de manera que los grupos de relaciones no solo tendrán su miembros como implicaciones sino que se tiene una medida de cuantas veces se presentó una implicación similar y la defusificación se hace considerando todos los consecuentes y cuantas veces aparece cada uno.

El pronóstico de series de tiempo difusas se ha enfocado en hacer modificaciones a los algoritmos planteados por [Song und Chissom, 1993a], [Song und Chissom, 1994], [Chen, 1996] y [Yu, 2005], a continuación se mencionan algunas de las más relevantes, para dar un panorama general de hacia donde se han enfocado los esfuerzos de predicción de series de tiempo difusas; en la mayoría de los casos tratan de optimizar, de alguna manera, el algoritmo propuesto por [Chen, 1996]. Como primer caso en [Huarng und Yu, 2006] se utiliza una red neuronal para encontrar las relaciones difusas entre los conjuntos usados, de manera que al realizar el pronóstico se obtiene el conjunto de salida por medio de la red neuronal. [Egrioglu u. a., 2010] y [Egrioglu u. a., 2011] proponen usar el algoritmo de [Chen, 1996] optimizando la longitud de los intervalos, tratando de minimizar el error medio

cuadrado, por medio de la función "fminbnd" de MATLAB. los resultados son comparados con los artículos predecesores y presentan mejores resultados. Mientras que en [Huang u. a., 2011] usan optimización por enjambre de partículas (en inglés Particle Swarm Optimization) (PSO) aplicado para obtener el pronóstico, el cual será una combinación ponderada de la información que ofrece la ultima implicación y la que ofrecen de manera general todas las implicaciones anteriores. [Amjad u. a., 2012] presentan un modelo híbrido en el cual usan PSO para obtener las contribuciones de las implicaciones y algoritmos genéticos (en inglés Genetic Algorithms) (GA), para optimizar la longitud de los intervalos.

[Sulandari und Yudhanto, 2015] presentan modificaciones a [Chen, 1996] y [Yu, 2005], que adaptan la técnica de promedios móviles para ponderar la contribución de cada conjunto presente en las implicaciones. [Sachdev und Sharma, 2015] utilizan algoritmos genéticos (en inglés Genetic Algorithms) (GA) para optimizar el rango y los intervalos de la serie de tiempo difusa aplicado al problema de abastecimiento (stock). [Efendi u. a., 2015] presentan un modelo de inversión que permite ponderar la contribución de cada relación difusa en su grupo y lo aplican al problema de abastecimiento así como en demanda de energía eléctrica. [Ismail u. a., 2015] usan series de tiempo difusas para el pronóstico en demanda de energía eléctrica en el caso particular de Taiwan. Otro ejemplo del uso de la series de tiempo difusas se tiene en [Efendi u. a., 2016], donde las usan para pronosticar en casos donde los datos no son estacionarios. Como casos de prueba usan la demanda de energía eléctrica y tipos de cambio. Finalmente en [Cheng u. a., 2016], modifican el método de agrupamiento de las implicaciones, usando el Algoritmo de k-medias.

2.2. Modelos difusos para pronóstico de series de tiempo

La diferencia principal de los modelos difusos que se utilizan para pronóstico de series de tiempo y las series de tiempo difusas radica en que los primeros usan la lógica difusa para construir el modelo de pronóstico, mientras que las segundas mapean la serie de tiempo convencional a una representación difusa. Las técnicas basadas en lógica difusa que se usan para pronóstico más comunes son modelos neuro-difusos, mapas cognitivos difusos y combinaciones de modelos de regresión con lógica difusa. A continuación se presentan

algunos trabajos hechos en cada uno de los temas mencionados.

2.2.1. Modelos neuro-difusos

[Jang, 1993] presenta la implementación de un modelo neuro-difuso llamado red adaptativa basada en un sistema de inferencia difusa (en inglés Adaptative Network Based Fuzzy Inference System) (ANFIS), el cual es una red adaptativa en la que las funciones usadas en sus nodos se dividen dependiendo de la capa. En la capa uno se usa como función en cada nodo la pertenencia del valor de entrada a la función de membresía predefinida (triangular, trapezoidal, campana, etc), en la capa dos se usa como función de los nodos, el producto o alguna otra forma de realizar una intersección de los datos de entrada, obteniendo una fuerza de activación. Posteriormente en la capa tres se usa una proporción que normaliza todas las fuerzas de activación, en la capa cuatro se hace la multiplicación de la fuerza de activación normalizada con una combinación lineal de las variables de entrada. En la quinta capa se hace la suma de todas las señales obtenidas en la capa anterior. En ese mismo artículo se presentan algunos usos para las redes adaptativas basadas en inferencia difusa, como modelar algunas funciones no lineales y hacer pronósticos en series de tiempo caóticas. Algunos ejemplos de pronóstico de series de tiempo usando modelos ANFIS se pueden apreciar en [Sfetsos, 2000], [Kurian u. a., 2006], [Firat und Güngör, 2008] y [Yadav und Balakrishnan, 2014]; donde hacen predicciones en series de tiempo de velocidad de viento, hidrológicas, iluminación diurna interior y tráfico de redes inalámbricas, respectivamente.

Por otro lado en [Kim und Kasabov, 1999] se presenta un enfoque llamado sistema hybrido de inferencia difusa (en inglés Hybrid Fuzzy Inference System) (HyFIS), el cual es un tipo especial de red neuronal, donde la primera capa captura los datos. La segunda capa pasa los datos por funciones de membresía asociadas a los términos lingüísticos que se usen. En la tercera capa se hace una intersercción de los valores obtenidos en la capa anterior, considerándose la parte IF de las reglas. Posteriormente en la cuarta capa se hace la unión de los consecuentes o parte THEN de las reglas. Finalmente en la quinta capa se hace la defusificación de los valores usando algún método para tal fin. Cabe mencionar que los pesos asignados a las conexiones de las capas 3 y 4 representan las fuerzas de activación de las reglas y son elegidos inicialmente al azar en un intervalo predefinido, para posteriormente

aplicar el método de descenso de gradiente para optimizar estos valores. Podemos encontrar trabajos donde se haya usado el concepto de modelos hibridos de inferencia difusa para pronóstico de series de tiempo en [Kim und Kasabov, 1999], [Sánchez u. a., 2007] y [Arango und Velasquez, 2014]. En el primer caso lo utilizan para pronóstico en series de tiempo de presencia de CO_2 y series de tiempo caóticas, en el segundo caso se comparan varios métodos neuro-difusos entre ellos HyFIS, en estimación de concentración de polen en el aire, finalmente en el tercer caso también se comparan diversos metodologías neuro-difusas en la predicción del índice de tipo de cambio en Colombia.

Un tercer enfoque de los modelos neuro-difusos es mostrado en [Nauck und Kruse, 1999], donde plantean el uso de una técnica llamada sistema neuro-difuso de aproximación de funciones (en inglés Neuro Fuzzy Systems for Function Approximation) (NEFPROX). En esta metodología se tiene un tipo especial de red con tres capas, donde la primera representa las variables de entrada y solo capturan los datos, la capa intermedia u oculta representa las reglas difusas, las cuales calculan los grados en que se cumple cada regla y la tercera capa tiene los valores de salida donde se hace la defuzzificación de los términos. Cada conexión entre la capa de entrada y la oculta es etiquetada con términos lingüísticos y presentan los pesos de los antecedentes de las reglas, en tanto que las conexiones entre la capa de salida y la oculta representan los pesos de los consecuentes de las reglas difusas. [Nauck und Kruse, 1998], [Nauck und Kruse, 1999] y [Sánchez u. a., 2007] muestran su aplicación en el pronóstico de series de tiempo. En los dos primeros casos se usa para la serie de tiempo de Mackey-Glass, que es una serie caótica sintética, el último se enfoca como ya se mencionó anteriormente en la predicción de la concentración de polen en el aire.

Existen más trabajos donde utilizan los conceptos de lógica difusa aplicados en la creación de redes adaptativas y redes neuronales. En ese sentido, con las referencias anteriormente mencionadas solo se pretende mostrar a grandes rasgos los enfoques que existen y su aplicación en los pronósticos de series de tiempo. En los artículos [Riid und Rüstern, 1998], [Sánchez u. a., 2007], [Efendigil u. a., 2009] y [Xing u. a., 2017] se pueden encontrar comparaciones de varios modelos neuro-difusos y son a la vez un punto de partida para buscar alguna técnica en particular, ya que en estos artículos se trata de hacer un compendio de lo que se desarrolló hasta el momento de la publicación de dichos trabajos.

2.2.2. Mapas cognitivos difusos

Los mapas cognitivos ([Axelrod, 2015]) representan relaciones causales entre dos o más términos, son un tipo de grafo donde se muestran las relaciones de consecuencia entre sucesos. [Kosko, 1986] y [Kosko, 1992] presentan una versión difusa de los mapas cognitivos originales, donde se consideran que las relaciones causales entre dos eventos tienen un determinado grado de activación, usando términos lingüísticos tales como: tiene poca relación, contribuye en gran medida y están estrechamente ligados, entre otros. En este trabajo definen la causalidad de eventos como una combinación de implicaciones, pero consideran que una relación de causalidad es más general que una implicación, de este modo si un concepto C_i tiene como consecuencia otro C_j plantean que para representarse mediante implicaciones se tendrían que cumplir dos condiciones, primero que $Q_i \longrightarrow Q_j$, donde los términos Q representan una correlación positiva y como segunda condición que $\sim Q_i \longrightarrow \sim Q_j$ donde los términos $\sim Q$ significan una correlación negativa o inversa. Esto último se plantea debido a que puede haber relaciones causales en las que conforme aumenta un concepto su consecuente tiene una relación directa y aumenta también, o bien si dismimuye uno el otro de igual manera; pero también pueden darse los casos inversos que el incremento en un concepto haga que decremente el otro, o al contrario, que cuando drecremente el primero el segundo incremente. Al finalizar el artículo, presenta una teoría asociada con los mapas cognitivos difusos y como modelar las relaciones causales por medio de conjuntos difusos y álgebra.

Algunas aplicaciones de estos modelos en el pronóstico de series de tiempo se puede apreciar en [Stach u. a., 2008], donde consideran que cada punto de la serie de tiempo tiene dos medidas de interes, su magnitud, la cual se fuzzifica, y el cambio entre un punto y otro el cual también se fuzzifica. Para contruir el mapa usan todas las posibles relaciones existentes considerando el número de conjuntos difusos tanto para la magnitud como el cambio, así que el número de nodos viene dado por el producto de la cantidad de conjuntos en cada caso. Porteriormente usando algoritmos genéticos de codificación real (en inglés Real Coding Genetic Algorithms) (RCGA), realizan un proceso de optimización para determinar cual es el mapa que más se ajusta a la serie de tiempo en cuestión. Sus resultados son comparados

con las series de tiempo de inscripciones en la universidad de Alabama presentado en [Song und Chissom, 1993a] y con la serie de tiempo abastecimiento mostrada en [Yu, 2005].

En [Song u. a., 2010a] y [Song u. a., 2010b] se presenta un modelo híbrido entre un mapa cognitivo difuso y una red neuronal. La red neuronal se diseña con cuatro capas, la primera capa simplemente recibe los datos de entrada de la serie de tiempo y la segunda capa toma esos datos fusificandolos en base a los conjuntos difusos planteados para el mapa cognitivo difuso. En esta etapa cada variable de entrada pasa por un grupo de conjuntos difusos que dividen el universo de discurso y finalmente esta capa se conecta a la siguiente por una serie de pesos. En la tercera capa se aplican las relaciones de causalidad del mapa cognitivo difuso y se defusifica. La última capa entrega los valores numéricos. La técnica presentada en estos artículos es aplicada en la predicción de series de tiempo usando series caóticas tales como Mackey-Glass y Lorenz.

[Homenda u. a., 2014] y [Pedrycz u. a., 2016] presentan otro enfoque donde modelan series de tiempo usando mapas cognitivos difusos. Estos no realizan un proceso de fusificación, aprendizaje, pronóstico y defusificación, sino que basan su trabajo en una técnica de ventana móvil, en la cual se toman n ventanas (de un tamaño fijo) de la serie de tiempo, donde n representa el número de nodos en el mapa cognitivo difuso. Posteriormente se crean tres matrices, una de pesos que contiene las relaciones entre todos los nodos del mapa, otra de activaciones que contiene las ventanas desplazadas en uno de la serie de tiempo y finalmente el pronóstico se guarda en otra matriz de objetivos, en la que cada elemento $y_{i,j}$ se calcula como una suma ponderada de los de la columna i de la matriz de pesos por los valores de la columna j de la matriz de activación. [Homenda u. a., 2014] la aplica en tres series de tiempo: lluvias anuales en Londres de 1813 a 1912, número de nacimientos por mes en Nueva York de 1946 a 1959 y los anillos de los arboles Campito para medir su crecimiento de los años 1907 a 1960. [Pedrycz u. a., 2016] usan este modelo con unas adecuaciones en la optimización de algunos parámetros, para predecir usando dos series de tiempo: temperatura mensual de una mina de cobre en grados Celsius en los años de 1933 a 1976 y flujo diario medio del río Oldman de 1988 a 1991.

2.2.3. Modelos de regresión difusa

Existen diversos modelos para hacer pronósticos de series de tiempo que se basan en regresión. [Homenda u. a., 2014] mencionan los más comunes, posteriormente en el Capítulo 3 de esta tesis se mencionan a fondo los modelos de regresión. A continuación se mencionan algunos trabajos encontrados en el estado del arte que implementan una versión difusa de varios de estos modelos de regresión. Si bien no se incluyen trabajos sobre todos los modelos de regresión que tienen una versión difusa se mencionan estos como referencia de una línea más de investigación donde se han hecho pronósticos usando de alguna manera la lógica difusa.

[Park u. a., 1995], aunque no presentan propiamente un modelo de predicción, usan la idea de un modelo ARMA combinado con lógica difusa aplicado al control de un sistema que consta de un carro y un péndulo invertido. [Yang und Huang, 1998] presentan la versión difusa de un modelo autorregresivo de media móvil con términos exógenos (en inglés Autoregressive Moving Average Exogenous) (ARMAX), donde forman reglas de implicación difusas a partir de los términos que intervienen en un modelo ARMAX, es decir, la parte regresiva, la parte de media móvil y los términos exógenos, usando metaheurísticas para determinar los parámetros óptimos de su modelo. Eso lo aplican en el pronóstico de datos de carga horaria del sistema Taipower en 1992 en cargas típicas del sistema en primavera, verano, otoño e invierno, así como sus registros asociados de variables meteorológicas, incluyendo temperatura, humedad relativa, velocidad del viento y precipitación.

Una versión difusa de ARIMA es presentada en [Tseng u. a., 2001], quienes dividen su técnica en tres fases. En la primera parte ajustan un modelo ARIMA usando la información disponible en las observaciones, los datos de entrada no se consideran difusos. El resultado de esta fase da los parámetros óptimos del modelo y los residuos se consideran Ruido blanco Gaussiano, el cual es usado como dato de entrada para la fase dos donde se minimiza el grado de fusificación y calculan el centro y ancho de cada conjunto difuso. Posteriormente se eliminan los datos de los límites superior e inferior cuando existen datos atípicos y se formula nuevamente el modelo de regresión difusa. Como ejemplos de aplicación usan una serie de tiempo del tipo de cambio al contado de dolár de Taiwan (NTD) a

dolár de Estados Unidos (USD) entre el banco principal de Taiwan y sus clientes, la cual consta de 40 mediciones tomadas de agosto a septiembre de 1996.

[Tseng und Tzeng, 2002] presentan un modelo estacional ARIMA difuso, en el cual primero se identifica el modelo, usando las ACF y PACF. Después se determinan los parámetros desconocidos, luego se hacen pruebas de bondad de ajuste a los residuos y finalmente se hacen pruebas de pronóstico en los datos. En este caso la parte difusa consiste en que las observaciones se consideran números difusos, que tienen un centro y un ancho determinados. El modelo se aplica en el pronóstico de dos series, el valor total de producción de la industria de maquinaria en Taiwán, y el volumen de ventas de refrescos. Se hace el símil con el modelo convencional. En [Popov und Bykhanov, 2005] hacen algo similar usando modelos autorregresivo general condicional con heteroscedasticidad (en inglés General Autoregressive Conditional Heteroscedasticity) (GARCH), donde modelan la volatilidad de series de tiempo usando la lógica difusa y después comparan la versión difusa con la versión convecional del modelo GARCH. Su técnica es aplicada a una serie de tiempo del Índice Industrial Dow Jones, tomando valores del periodo febrero de 2004 a febrero de 2005 (264 observaciones).

Un método que se decide mencionar más a fondo por su naturaleza es basado en vecinos cercanos en combinación con la lógica difusa, presentado en [Singh, 1998]. En este caso su idea básica es predecir el punto X_{N+1} de una serie de tiempo a partir de los puntos más parecidos a X_N, para determinar cuales puntos del pasado son parecidos usan una función de membresía la cual se calcula como $\mu(X_t) = \frac{1}{1+(d(X_t-X_N)/F_d)^{F_e}}$, posteriormente se normaliza la membresía dando como resultado una $\mu'(X_t)$. A continuación presentan dos criterios de decisión en el primer enfoque simplemente si $\mu'(X_t) \geq \lambda$ se dice que es un vecino cercano. En la segunda forma se crea un vector auxiliar de cambios donde cada elemento b_t es 0 si $X_{t+1} < X_t$, 1 si $X_{t+1} > X_t$ o 2 si $X_{t+1} = X_t$, entonces el criterio de decisión para saber si un vecino es cercano es $\mu'(X_t) \geq \lambda$ así como $b_{t-1} = b_{n-1}$ y $b_{i-2} = b_{n-2}$ y $\ldots b_{i-j} = b_{n-j}$, donde j, λ, F_d, F_e son parámetros que se eligen por prueba y error. Los datos de ventas para el pronóstico provienen de una empresa de fabricación ABX que fabrica equipos de control ubicada en el suroeste de Inglaterra. Se han seleccionado los datos de ventas para cuatro productos diferentes A, B, C y D; el total de datos es de 70 meses para

cada serie y se dividen en 50 de entrenamiento 20 de validación o predicción. Este método se menciona debido a que tiene cierta relación con la técnica que se implementa a lo largo de esta tesis.

Finalmente se mencionan en conjunto el trabajo presentado en [Flores u. a., 2015a], [Flores u. a., 2015b] y [Flores u. a., 2016b], donde presentan una técnica de pronóstico basada también en el método de vecinos cercanos, aunque aquí el enfoque es bastante diferente de lo mencionado en [Singh, 1998], ya que sí se usan propiamente conjuntos difusos en tanto que en [Singh, 1998] tan solo usan una idea muy vaga de estos. Estas tres referencias podrían considerarse la base para desarrollar el presente trabajo. Cabe mencionar que lo presentado a lo largo de esta tesis difiere incluso de estos trabajos en el sentido de que aquí se usan exclusivamente los conceptos de lógica difusa para hacer el pronóstico. En los artículos anteriores se mezclan algunos conceptos del método de vecinos cercanos, existen algunas similitudes pero el enfoque de pronóstico es diferente. En [Flores u. a., 2016a] presentan un análisis de complejidad sobre su versión difusa de NN, donde determinan que la complejidad computacional en tiempo para la fase de aprendizaje es lineal ($O(N)$) con respecto a la longitud de la serie de tiempo (N) en tanto que el pronóstico se realiza en tiempo constante ($O(1)$). Esto sirve como base para lo desarrolado a lo largo de esta tesis, por las similitudes que existen entre ambas técnicas la complejidad computacional debe ser la misma.

Conclusiones del capítulo

Hasta este punto, se han explicado a grandes rasgos los trabajos realizados en el área de pronósticos de series de tiempo usando modelos inspirados en lógica difusa. Mencionando las series de tiempo difusas, los mapas cognitivos difusos, técnicas que combinan la lógica difusa con métodos de inteligencia artificial y también procedimientos hybridos entre modelos estadísticos y lógica difusa. Finalmente se hace énfasis en las técnicas que usan una versión difusa del método NN. Si bien, existe un gran número de contribuciones que utilizan la lógica difusa no se observa ninguno que la use puramente para el problema de pronóstico.

Capítulo 3

Análisis de series de tiempo y modelos de pronóstico

En este capítulo se presenta la teoría del pronóstico de series de tiempo, se mencionan los enfoques más comúnmente usados (entre ellos los modelos ARIMA), y se abordan a grandes rasgos las redes neuronales y su uso en el pronóstico. Finalmente se menciona la predicción en series de tiempo caóticas, el análisis de espacio de fase (donde se introducen los conceptos de la dimensión de embebido y el tiempo de retardo). También se hace alusión al método de vecinos cercanos, que usa las herramientas de espacio de fase para construir un modelo de estimación de estados futuros a partir de la información relevante de la serie de tiempo.

3.1. Análisis de las series de tiempo

El análisis de las series de tiempo se puede realizar a partir de tres enfoques distintos. El primero se hace en el dominio del tiempo, el segundo en el dominio de la frecuencia y el tercero es la representación en espacio de estados. La mayoría de los métodos utilizados para pronóstico se basan en la representación en el dominio del tiempo, así que es en la que nos centraremos en esta tesis.

Las series de tiempo surgen a partir de sistemas dinámicos, sistemas que la salida

depende de la entrada en el instante actual y en instantes anteriores. Según [Ogata, 1996], [Alligood u. a., 2006] y [Moctezuma, 2015] los sistemas dinámicos se clasifican con respecto a las siguientes perspectivas. En base al tiempo se dividen en continuos y discretos, en los primeros las variables toman valores para cada instante de tiempo mientras que en los segundos solo se tienen valores para instantes de tiempo específicos, por lo anterior se puede decir que las series de tiempo siempre tienen una representación discreta.

Una segunda forma de clasificar los sistemas dinámicos es a partir de si son o no variantes en el tiempo. Es decir, si depende explícitamente del tiempo, en este sentido es evidente que las series de tiempo lo son. La siguiente manera de catalogar sistemas dinámicos es considerando si son lineales o no. Se dice que un sistema es lineal si cumple el principio de superposición, a grandes rasgos que un sistema sea lineal implica que su respuesta es una combinación ponderada de sus variables de entrada. Por lo tanto, un sistema lineal generalmente se puede ver como la combinación de efectos de sistemas más sencillos. Con respecto a lo anterior existen series de tiempo que surgen tanto de sistemas lineales como no lineales.

Otra forma de agrupar los sistemas es en base a si son causales o no. Se dice que un sistema es causal cuando la salida del sistema no depende de instantes futuros. Las series de tiempo surgen generalmente de sistemas causales. La última forma de dividir los sistemas dinámicos es en determinísticos y estocásticos, en los primeros los estados futuros están completamente determinados por una entrada y un estado inicial. Por el contrario, en los segundos existe intervención del azar, en la mayoría de los casos las series de tiempo surgen de sistemas deterministas, pero también se considera cierta parte aleatoria para modelar por ejemplo incertidumbre en las mediciones.

Las propiedades de los sistemas dinámicos dan lugar a los atributos que presentan las series de tiempo, a continuación se abordan estas cualidades.

3.1.1. Características de las series de tiempo

El análisis de series de tiempo debe considerar sus características más importantes. Según [Chafield, 1975], [Perez, 2010], [Brockwell und Davis, 2013] y [Homenda u. a., 2014] las series de tiempo se descomponen en cuatro componentes principales:

- Tendencia Secular.- Son los cambios que presenta la serie de tiempo a largo plazo en la media, se considera el resultado de fuerzas que actúan de manera persistente que afectan el crecimiento o reducción de la misma.

- Variaciones cíclicas.- Son variaciones que se ajustan a un periodo, independientes a las variaciones estacionales y la tendencia secular. Se pueden ver como los cambios a medio plazo, por ejemplo los ciclos comerciales, los cuales dependen de la prosperidad, recesión, depresión, etcétera.

- Variaciones estacionales.- Representan fluctuaciones cíclicas relacionadas con el calendario. Aquí se consideran los efectos de las estaciones o fenómenos que se repiten año tras año, por ejemplo fiestas o costumbres que se presentan con una regularidad y que afectan a la variable que se mide. Se pueden considerar como cambios a corto plazo.

- Variaciones Residuales.- Son alteraciones que se producen por efectos aleatorios o muy difíciles de cuantificar, también pueden ser considerados los cambios sistemáticos. Normalmente estos cambios son modelados por medio de probabilidad.

El enfoque clásico de análisis utiliza estas cuatro propiedades para plantear un modelo que las combina aditiva o multiplicativamente, por lo que se deben obtener cada una de las componentes.

Existen algunas técnicas que cumplen con esta tarea, [Perez, 2010] plantea que para estimar la tendencia se pueden usar el método gráfico de los puntos medios, el método de las medias escalonadas, el método de las medias móviles o bien un ajuste por mínimos cuadrados con alguna función predeterminada (lineal, polinomial, logarítmica, etcétera). Para el cálculo de la componente cíclica se encuentran procedimientos como el método del ciclo medio o el de los residuos, esta componente es la más difícil de estimar. Para obtener la componente residual generalmente se plantea una hipótesis sobre la distribución de los datos. Existen diversas pruebas que permiten verificar la validez de la hipótesis, es decir, que tanto la distribución seleccionada se ajusta a los datos; se tienen estadísticos de prueba como el z, t y los valores p; las pruebas más comunes son: Prueba de Kolmogórov-Smirnov y Chi cuadrada de Pearson (para mayor información consultar [Walpole, 2012]). Finalmente

la componente estacional se obtiene quitando el efecto de las demás componentes, con sustracción si el modelo para combinarlas fue aditivo o división si fue multiplicativo.

3.1.2. Series de tiempo estacionarias

Las series de tiempo estacionarias son un tipo muy importante de series. Se dice que una serie de tiempo es estrictamente estacionaria si sus propiedades no se ven afectadas por un cambio en el origen del tiempo. Su definición formal es como se observa en la Definición 1.

Definición 1 *Una serie de tiempo se conoce como estacionaria si su comportamiento probabilístico es idéntico para una colección de puntos* $\{X_1, X_2, \ldots, X_N\}$*, que para otro con un desplazamiento en el tiempo* $\{X_{1+i}, X_{2+i}, \ldots, X_{N+i}\}$*. Esto es:*

$$P\left(X_1 \leq c_1, X_2 \leq c_2, X_3 \leq c_3, \ldots, X_N \leq c_N\right) = P\left(X_{1+i} \leq c_1, X_{2+i} \leq c_2, \ldots, X_{N+i} \leq c_N\right)$$

Que la serie de tiempo sea estacionaria implica un tipo de equilibrio estadístico o estabilidad en los datos. Consecuentemente, la serie temporal tiene una media constante definida por la media muestral; de la misma manera la varianza está dada por la varianza muestral. Por otra parte, una serie de tiempo débilmente estacionaria, es un proceso de varianza finita tal que la función de la media es acotada y no depende del tiempo.

La mayoría de los métodos de análisis consideran que la serie de tiempo es estacionaria, y en general lo que se busca es convertir una serie de tiempo no estacionaria en una que sí lo sea. [Chafield, 1975], [Douglas C. Montgomery, 2008], [Robert H. Shumway, 2011], [Brockwell und Davis, 2013].

3.1.3. Autocovarianza, autocorrelación y autocorrelación parcial

La dependencia entre dos observaciones dadas X_s y X_t (de una serie de tiempo, X) puede ser evaluada numéricamente, utilizando las nociones de covarianza y correlación según lo planteado en [Douglas C. Montgomery, 2008], [Robert H. Shumway, 2011] y [Brockwell und Davis, 2013] con las siguientes definiciones.

Definición 2 *Suponiendo que la varianza de X es finita, se define la autocovarianza como el momento producto con respecto de la media, expresado en* (3.1)*, para todos los t y s. Se denomina autocovarianza ya que tanto t como s se toman de la misma serie de tiempo.*

$$\gamma_k(s,t) = cov\,(X_s, X_t) = E\left[(X_s - \bar{X}_s)(X_t - \bar{X}_t)\right] \tag{3.1}$$

donde $\bar{X}_t$ y $\bar{X}_s$ son las medias para X_t y X_s, respectivamente.

Si se define que $s = t + k$, denominando a k como un retardo y se calcula la covarianza para todos los posibles retardos sobre la serie de tiempo ($k = 0, 1, 2, \ldots, N-1$), el conjunto de todos estos valores $\gamma_k(s,t)$ se conoce como función de autocovarianza.

En el caso donde $s = t$ (implica que $k = 0$) la expresión de la covarianza se reduciría a la varianza $\gamma_0(t,t) = cov\,(X_t, X_t) = var(X_t) = E\left[(X_t - \bar{X}_t)^2\right] = \sigma^2$.

Definición 3 *El coeficiente de autocorrelación, para el retardo k, se define a partir de la varianza y las covarianzas como se muestra en* (3.2)*.*

$$\rho_k = \frac{cov(X_s, X_t)}{var(X_t)} = \frac{\gamma_k(s,t)}{\gamma_0(t,t)} \tag{3.2}$$

donde la colección de los valores ρ_k para todas los valores de k posibles se conoce como función de autocorrelación (ACF)

La autocovarianza mide la dependencia lineal que existe entre los puntos de la misma serie, observada en diferentes momentos. Las series muy suaves exhiben funciones de autocovarianza que permanecen grandes incluso cuando el retardo k es grande, o sea que t y s están muy alejados entre sí. Por otra parte, las series agudas tienen funciones de autocovarianza que son casi cero, para separaciones grandes. Así para las series de tiempo consideradas estacionarias la función de autocovarianza $\gamma_k(s,t)$ depende de s y t solo a través de su diferencia $|s-t|$. El último concepto a mencionar es el de función de autocorrelación parcial y se explica en la Definición 4.

Definición 4 *La función de autocorrelación parcial de un proceso estacionario, X, denotada por ϱ_k es como se observa en* (3.3)*.*

$$\varrho_1 = Corr\left(X_{t+1}, X_t\right) = \rho_1$$
$$\varrho_k = Corr\left(X_{t+k} - \hat{X}_{t+k}, X_t - \hat{X}_t\right), k \geq 2 \tag{3.3}$$

donde $\hat{X}_{t+k}$ *y* $\hat{X}_t$ *son estimadores de regresión que hacen una combinación lineal de los puntos* $\{X_{t+1}, \ldots, X_{t+k-1}\}$ *y minimizan el error cuadrático medio (en inglés Mean Square Error) (MSE). La función de autocorrelación parcial entre* X_{t+k} *y* X_t *es la ACF con la dependencia lineal de* X_{t+1} *hasta* X_{t+k-1} *eliminada, es decir, la ACF entre* X_{t+1} *y* X_{t+k} *que no se explica por retrasos de* 1 *hasta* $k-1$*. Si el proceso es Gaussiano, entonces* $\varrho_k = Corr\left(X_{t+k}, X_t | X_{t+1}, \ldots, X_{t+k-1}\right)$*; esto es,* ϱ_k *es el coeficiente de correlación entre* X_{t+k} *y* X_t *en la distribución bivariada condicional de* (X_{t+k}, X_t) *en* $\{X_{t+1}, \ldots, X_{t+k-1}\}$*.*

3.2. Enfoques clásicos para pronóstico de series de tiempo

La mayoría de los modelos convencionales usados en el análisis y la predicción de series de tiempo se basan en las características mencionadas en el apartado anterior, y normalmente consideran que las series de tiempo son generadas a partir de sistemas lineales. A continuación se explican los modelos tradicionalmente usados.

3.2.1. Modelos autoregresivos, integrados y de medias móviles

En las series de tiempo, es deseable permitir que la variable de interés sea influenciada por los valores pasados de la variable independiente (el tiempo) y posiblemente por sus propios valores pasados. Si el presente se puede modelar plausiblemente en términos de sólo los valores pasados de las variables conocidas, tenemos la perspectiva atractiva de que la predicción será posible. [Robert H. Shumway, 2011]

Modelos autoregresivos (AR)

La idea en la que se basan los modelos autoregresivos es que el valor actual de la serie de tiempo X_t puede ser representado como una función de los p valores anteriores de la misma serie de tiempo $X_{t-1}, X_{t-2}, \ldots, X_{t-p}$, donde p representa el número de pasos

que se consideran en el pasado para obtener el valor actual. Esta idea se origina al observar la función de autocorrelación (ACF), la cual en muchos casos muestra que existe cierta correspondencia entre los mismos valores de la serie de tiempo. De esta manera ahora parece razonable plantear un modelo autoregresivo.

Definición 5 *Un modelo AR de orden p, denotado* $\boldsymbol{AR}(p)$*, tiene la forma que se muestra en* (3.4).

$$X_t = \phi_1 * X_{t-1} + \phi_2 * X_{t-2} + \cdots + \phi_p * X_{t-p} + w_t \tag{3.4}$$

donde X es una serie estacionaria según lo presentado en la Definición 1, y los términos ϕ *son constantes diferentes de cero. Se considera que w es Ruido blanco Gaussiano con media* $E(w) = 0$ *igual a cero y varianza* σ_w^2*. También se supone que la serie de tiempo X tiene media cero. En el caso donde la serie de tiempo tuviera media diferente de cero digamos* $\bar{X}$*,* (3.4) *tomaría la forma de* (3.5).

$$X_t = \alpha + \phi_1 * X_{t-1} + \phi_2 * X_{t-2} + \cdots + \phi_p * X_{t-p} + w_t \tag{3.5}$$

donde $\alpha = \bar{X}(1 - \phi_1 - \phi_2 - \cdots \phi_p)$

Por comodidad se define el operador de retroceso para k pasos hacia atrás como se muestra en (3.6). Ahora expresando (3.4) en base al operador de retroceso se obtiene (3.7).

$$B^k X_t = X_{t-k} \tag{3.6}$$

$$\left(1 - \phi_1 B - \phi_2 B^2 \cdots \phi_p B^p\right) X_t = w_t \tag{3.7}$$

$$\phi(B) X_t = w_t$$

donde $\phi(B) = \left(1 - \phi_1 B - \phi_2 B^2 \cdots \phi_p B^p\right)$ se conoce como el operador de autoregresión.

Un proceso autoregresivo de primer orden, $\boldsymbol{AR(1)} \Rightarrow X_t = \phi_1 X_{t-1} + w_t$, puede ser presentado como un proceso lineal según (3.8), donde su función de autocovarianza se calcula por medio de (3.9) y su ACF es como se observa en (3.10), planteado en [Chafield, 1975], [Douglas C. Montgomery, 2008], [Robert H. Shumway, 2011] y [Brockwell und Davis, 2013].

$$X_t = \sum_{j=0}^{\infty} \phi^j w_{t-j} \tag{3.8}$$

$$\gamma_k(s,t) = cov(X_s, X_t) = \frac{\sigma_w^2 \phi^k}{1-\phi^2}, k \geq 0. \tag{3.9}$$

$$\rho_k = \frac{\gamma_k(s,t)}{\gamma_0(t,t)} = \phi^k, k \geq 0. \tag{3.10}$$

Modelos de media móvil (MA)

La media móvil sirve para poder seguir los cambios que ocurren en una serie de tiempo. Si se quisiera usar la media de la serie de tiempo como estimador de los datos a futuro, se observaría que esta sólo da una tendencia general de la serie de tiempo. Esto se debe a que los datos más antiguos dan una aportación igual que los recientes. Es lógico pensar que los datos más recientes aportan mayor información sobre la dinámica del sistema. Por lo anterior se puede calcular la media pero sólo de los n datos más recientes y usarse como estimador del punto siguiente en la serie de tiempo. A continuación se da una definición formal del modelo de media móvil, el cual se usa para calcular el valor actual de la serie de tiempo a partir de los términos de error (incertidumbre) w .

Definición 6 *Un modelo de media móvil de orden q (**MA(q)**) se define por* (3.11)*. Este modelo se puede escribir usando el operador de retroceso que se mostraba en* (3.6)*, como se observa en* (3.12)*.*

$$X_t = w_t + \theta_1 w_{t-1} + \theta_2 w_{t-2} + \cdots + \theta_q w_{t-q} \tag{3.11}$$

$$X_t = \theta(B) w_t \tag{3.12}$$

en donde $\theta(B)$ es conocido como operador de media móvil y está dado por (3.13)*.*

$$\theta(B) \;=\; 1 + \theta_1 B + \theta_2 B^2 + \cdots + \theta_q B^q \tag{3.13}$$

Considerando el modelo de media móvil de primer orden (***MA*(1)**) que tiene la forma $X_t \;=\; w_t + \theta_1 w_{t-1}$, se pueden calcular su función de autocovarianza y su ACF como

se aprecia en (3.14) y (3.15). Aquí se debe notar que $\rho_1 \leq 0.5$ para cualquier valor de θ_1, también que X_t está correlacionado con X_{t-1}, pero no con X_{t-2}, X_{t-3}, etcétera; lo cual contrasta con el modelo $\boldsymbol{AR(1)}$.

$$\gamma_k(s,t) = \begin{cases} (1+\theta_1^2)\sigma_w^2 & \text{si } k = 0 \\ \theta_1\sigma_w^2 & \text{si } j = 1 \\ 0 & \text{si } k > 1 \end{cases} \tag{3.14}$$

$$\rho_k = \begin{cases} \frac{\theta_1}{(1+\theta_1^2)} & \text{si } k = 1 \\ 0 & \text{si } k > 1 \end{cases} \tag{3.15}$$

Una última consideración que se menciona antes de adentrarse formalmente en los modelos ARMA, es el hecho de que un modelo $\boldsymbol{MA(1)}$ se considera invertible en el sentido de que se tiene la misma función de autocovarianza para θ_1 y $1/\theta_1$. Como ejemplo considere el par de parámetros $\sigma_w^2 = 1$ y $\theta_1 = 5$ que tienen la misma función de autovarianza que el par $\sigma_w^2 = 25$ y $\theta_1 = 1/5$. Supongase que el primer modelo tiene la forma $X_t = w_t + \frac{w_{t-1}}{5}$ para w_t que es una variable aleatoria Independiente e idénticamente distribuida (iid) que sigue una distribución normal con media cero y varianza 25 (i.e., $N(0, 25)$); así que el segundo modelo tendría la forma $X_t = w_t + 5w_{t-1}$ para la variable w_t iid con distribución normal $N(0, 1)$, los cuales son modelos equivalentes. Para mayor información se puede consultar [Chafield, 1975], [Douglas C. Montgomery, 2008], [Robert H. Shumway, 2011] y [Brockwell und Davis, 2013].

Modelo autoregresivo de media móvil (ARMA)

El modelo ARMA combina las características de los modelos AR y MA, su planteamiento formal se da en la Definición 7.

Definición 7 *Un modelo autorregresivo de media móvil (en inglés Autoregressive Moving Average) (ARMA) tiene la estructura de* (3.16).

$$X_t = \phi_1 X_{t-1} + \cdots + \phi_p X_{t-p} + w_t + \theta_1 w_{t-1} + \cdots + \theta_q w_{t-q} \tag{3.16}$$

donde los términos ϕ y θ son diferentes de cero y $\sigma_w^2 > 0$. Los parámetros p y q son llamados el orden autoregresivo y de media móvil, respectivamente, de modo que el modelo se puede representar como ARMA(p,q). Si X_t tiene media diferente de cero, digamos $\bar{X}$, se adiciona el término $\alpha = \bar{X}(1 - \phi_1 - \cdots - \phi_p)$ y el modelo queda como se muestra en (3.17).

$$X_t = \alpha + \phi_1 X_{t-1} + \cdots + \phi_p X_{t-p} + w_t + \theta_1 w_{t-1} + \cdots + \theta_q w_{t-q} \tag{3.17}$$

nuevamente se asume que w es Ruido blanco Gaussiano con media cero $E(w) = 0$ y varianza σ_w^2.

Para explicar las propiedades de los modelos ARMA es más conveniente expresarlos usando los operadores AR y MA definidos en (3.7) y (3.12). De esta manera (3.16) toma la forma de (3.18).

$$\phi(B)X_t = \theta(B)w_t \tag{3.18}$$

Existen tres problemas principales que se presentan en los modelos ARMA. 1) existe redundancia de parámetros, 2) la parte autoregresiva puede depender del futuro (no es causal) y 3) los modelos de medias móviles no son únicos. A continuación se dan algunas definiciones para explicar estos problemas.

A partir de (3.18) se puede demostrar que un proceso $ARMA(1,1)$ que tiene la forma $X_t = w_t$, conocido como de Ruido blanco Gaussiano, es equivalente a otro que tuviera la forma $X_t - 0.5X_{t-1} = w_t - 0.5w_{t-1}$, donde la solución sigue siendo $X_t = w_t$. Sin embargo en el segundo caso se está ocultado el hecho de que el proceso también es Ruido blanco Gaussiano. Al usar el operador de retardo expresado en (3.6) y los modelos AR denotados por (3.7) y MA que se observan en (3.12)) se puede llegar a que

$$X_t = (1 - 0.5B)^{-1}(1 - 0.5B)X_t = (1 - 0.5B)^{-1}(1 - 0.5B)w_t = w_t$$

lo cual indica que nada ha cambiado. Esto es conocido como parámetros de redundancia o sobreparametrización y es útil cuando se busca determinar si los datos están correlacionados o no. En el caso de no eliminar los parámetros de redundancia se podría pensar que si lo están aunque realmente no existe correlación entre los datos.

Considerando la Definición 8 se puede saber si un modelo ARMA es causal y en base a la Definición 9 se tiene forma de conocer si es invertible.

Definición 8 *Un modelo $ARMA(p,q)$ es causal, si una serie de tiempo X que tiene almacenados valores para $t = 0, \pm 1, \pm 2, \ldots, \pm N$ puede ser escrita como un proceso unilateral como se observa en* (3.19).

$$X_t = \sum_{j=0}^{\infty} \psi_j w_{t-j} = \psi(B) w_t, \tag{3.19}$$

donde $\psi(B) = \sum_{j=0}^{\infty} \psi_j B^j$ y $\sum_{j=0}^{\infty} |\psi_j| < \infty$; con $\psi_0 = 1$. En otras palabras, un modelo ARMA es causal sii $\phi(z) \neq 0$ para $|z| \leq 1$. Así los coeficientes del proceso lineal serían $\psi(z) = \sum_{j=0}^{\infty} \psi_j z^j = \frac{\theta(z)}{\phi(z)}$.

Definición 9 *Un modelo $ARMA(p,q)$ es invertible, si la serie de tiempo X puede ser escrita según* (3.20).

$$\pi(B) X_t = \sum_{j=0}^{\infty} \pi_j X_{t-j} = w_t, \tag{3.20}$$

donde $\pi(B) = \sum_{j=0}^{\infty} \pi_j B^j$ y $\sum_{j=0}^{\infty} |\pi_j| < \infty$; con $\pi_0 = 1$. Dicho de otra manera, un modelo ARMA es invertible sii $\theta(z) \neq 0$ para $|z| \leq 1$ y los coeficientes del proceso estarían dados por $\pi(z) = \sum_{j=0}^{\infty} \pi_j z^j = \frac{\phi(z)}{\theta(z)}$.

La última parte que se explica sobre el modelo ARMA es cómo realizar pronósticos. Basándose en lo planteado en [Robert H. Shumway, 2011], considérese X estacionaria y con media cero, de acuerdo con la Definición (3.1.2); además el modelo es invertible y causal, donde w es Ruido blanco Gaussiano ($N(0, \sigma_w^2)$). Se desea calcular X_{N+k} a partir de $X = \{X_1, X_2, \ldots, X_N\}$, para $k = 1, 2, \ldots, n$. La predicción se debería calcular según (3.21), como la esperanza condicional dadas las observaciones con las que se cuenta. Sin embargo si se considera que N es lo suficientemente grande se pueden aproximar los valores $\hat{X}_{N+k}$ por los términos $\tilde{X}_{N+k}$, los cuales se calculan según (3.22) y son la esperanza condicional considerando una serie de tiempo de tamaño infinito.

$$\hat{X}_{N+k} = E\left[X_{N+k}|X_N, \ldots, X_1\right] \tag{3.21}$$

$$\tilde{X}_{N+k} = E\left[X_{N+k}|X_N, X_{N-1}, \ldots, X_1, X_0, X_{-1}, \ldots, -\infty\right] \tag{3.22}$$

Recordando que X se supuso causal ($X_{N+k} = \sum_{j=0}^{\infty} \psi_j w_{N+k-j}$, $\psi_0 = 1$) e invertible ($w_{N+k} = \sum_{j=0}^{\infty} \pi_j X_{N+k-j}$, $\pi_0 = 1$) entonces 3.22 genera (3.23) y (3.24).

$$\tilde{X}_{N+k} = \sum_{j=0}^{\infty} \psi_j \tilde{w}_{N+k-j} = \sum_{j=k}^{\infty} \psi_j w_{N+k-j} \tag{3.23}$$

$$\tilde{X}_{N+k} = -\sum_{j=1}^{k-1} \pi_j \tilde{X}_{n+m-j} - \sum_{j=m}^{\infty} \pi_j X_{N+k-j} \tag{3.24}$$

donde $\tilde{w}_t = E\left[w_t|X_N, X_{N-1}, \ldots, X_1, X_0, X_{-1}, \ldots, -\infty\right] = w_t$ para $t \leq N$ y $\tilde{w}_t = 0$ para $t > N$.

Adicionalmente $E\left[X_t|X_N, X_{N-1}, \ldots, X_1, X_0, X_{-1}, \ldots, -\infty\right] = X_t$ para $t \leq N$. Por lo tanto las predicciones se pueden calcular usando (3.24), empezando con el pronóstico de OSA, es decir $k = 1$ y continuando para $k = 2, 3, \ldots, n$. Para obtener el MSE se usa (3.23) de la cual se desprende que la diferencia $X_{N+k} - \tilde{X}_{N+k}$ es $\sum_{j=0}^{n-1} \psi_j w_{N+n-j}$.

$$P_{N+k} = E\left[X_{N+k} - \tilde{X}_{N+k}\right] = \sigma_w^2 \sum_{j=0}^{n-1} \psi_j^2 \tag{3.25}$$

Para mayor información sobre los modelos ARMA se pueden consultar [Douglas C. Montgomery, 2008], [Robert H. Shumway, 2011] y [Brockwell und Davis, 2013].

Modelo autoregresivo integrado de media móvil (ARIMA)

Es común observar procesos donde las magnitudes de las variables que observamos no tienen un nivel constante, sin embargo exhiben un comportamiento homogéneo con el tiempo. En este sentido, hasta ahora se ha hecho la suposición de que las series de tiempo son estacionarias, según lo planteado en la subsección 3.1.2. Ahora se tomará en consideración que las series de tiempo pueden no ser estacionarias.

Tomando como partida $X_t = X_{t-1} + w_t$, que es un modelo $AR(1)$ con $\phi_1 = 1$, se puede deducir que $\Delta X_t = w_t$ (que representa la derivada discreta) y este proceso es

estacionario. En general una serie de tiempo no estacionaria se puede representar como la combinación de dos componentes, una de tendencia variable o no estacionaria (μ_t) y una componente estacionaria con media cero (Y_t). Esto se representa de manera genérica como se observa en (3.26).

$$X_t = \mu_t + Y_t \Rightarrow Y_t = \Delta^d X_t \tag{3.26}$$

donde $\mu_t = \beta_0 + \beta_1 t$ y Y_t es estacionaria. Si se diferencia dicho proceso se tendrá un proceso estacionario, de esta manera se tiene:

$$\Delta X_t = X_t - X_{t-1} = \beta_1 + Y_t - Y_{t-1} = \beta_1 + \Delta Y_t$$

Otra forma de llegar a la misma expresión es tomando en cuenta que μ_t expresada en (3.26) es una variable estocástica y varía lentamente de acuerdo con una paso aleatorio. Es decir, $\mu_t = \mu_{t-1} + v_t$ donde v_t (paso aleatorio) es estacionario y sigue una distribución normal. En este caso $\Delta X_t = v_t + \Delta Y_t$. A partir de esto se puede pensar que pasa si $\mu_t = \mu_{t-1} + v_t$ y a su vez $v_t = v_{t-1} + e_t$ donde se supone que e_t es otra variable aleatoria y al mismo tiempo un proceso estacionario, entonces $\Delta X_t = v_t + \Delta Y_t$ no es estacionario pero $\Delta^2 X_t = e_t + \Delta^2 Y_t$ si lo es. Esto conduce a pensar que si μ_t es un polinomio de k-ésimo orden, $\mu_t = \sum_{j=0}^{k} \beta_j t^j$, entonces la serie diferenciada $\Delta^k Y_t$ es estacionaria.

Los modelos autoregresivo integrado de media móvil, (en inglés Autoregressive Integrated Moving Average) (ARIMA), son una ampliación de los modelos ARMA que incluyen diferenciación. A continuación se precisa formalmente este concepto en la Definición 10.

Definición 10 *Un proceso X_t se dice que es $\boldsymbol{ARIMA}(p, d, q)$ si $\Delta^d X_t = (1-B)^d X_t$ es $\boldsymbol{ARMA}(p, q)$. En general el modelo se representa como se observa en 3.27.*

$$\phi(B)(1-B)^d X_t = \theta(B) w_t \tag{3.27}$$

Si $E(\Delta^d X_t) = \mu$, el modelo toma la forma $\phi(B)(1-B)^d X_t = \delta + \theta(B) w_t$ donde $\delta = \mu(1 - \phi_1 - \cdots - \phi_p)$.

Debido a la no estacionareidad, se debe tener cuidado al hacer diferenciaciones en los pronósticos. En este sentido los aspectos relevantes de este problema, se pueden manejar más adecuadamente a través de modelos en espacio de estado. Se hace énfasis en que

$Y_t = \Delta^d X_t$ es un modelo ARMA, debido a lo anterior se pueden usar los conceptos mencionados sobre estos modelos, para obtener los pronósticos de Y_t. Estos pronósticos a su vez conducen a estimaciones para X_t. Por ejemplo si se define $d = 1$, dados los pronósticos $\hat{Y}_{N+k}$ para $k = 1, 2, \ldots, n$ se tiene que $\hat{Y}_{N+k} = \hat{X}_{N+k} - \hat{X}_{N+k-1}$. Así que la predicción en X se calcula como se observa en (3.28).

$$\hat{X}_{N+k} = \hat{Y}_{N+k} + \hat{X}_{N+k-1} \tag{3.28}$$

con condiciones iniciales $X_{N+1} = Y_{N+1} + X_N$.

Aunque es un poco más complicado obtener los errores de predicción P_{N+k}, (3.25) es una buena aproximación cuando N es grande. Así el error cuadrático medio (en inglés Mean Square Error) se calcula como se observa en (3.29).

$$P_{N+k} = \sigma_w^2 \sum_{j=0}^{n-1} \psi_j^{*2}, \tag{3.29}$$

donde ψ_j^{*2} es el coeficiente de z^j en $\psi^*(z) = \theta(z)/\phi(z)(1-z)^d$.

Para finalizar, se explican algunos pasos básicos para ajustar modelos ARIMA a datos de series de tiempo. Estos pasos implican graficar los datos, realizar algunas transformaciones, identificar los órdenes de dependencia del modelo, hacer una estimación de parámetros y finalmente se lleva a cabo un diagnóstico para elegir un modelo. En primer lugar, como con cualquier análisis de datos, debemos construir un diagrama de tiempo de los datos e inspeccionar el gráfico para detectar cualquier anomalía. Si la variabilidad en los datos crece con el tiempo, será necesario transformar los datos para estabilizar la varianza.

Después de transformar adecuadamente los datos, el siguiente paso es identificar los parámetros preliminares, es decir, valor del orden autorregresivo (p), el orden de diferenciación (d) y el orden de media móvil (q). A partir de la gráfica de la serie de tiempo se puede distinguir si es necesario realizar una diferenciación. Si se requiere diferenciar, entonces se diferencia una vez y se revisa el gráfico de tiempo para ΔX_t, el cual nos da información de si es necesaria una diferenciación adicional. Este proceso se repite hasta que se observe que ya no es necesario hacerlo una vez más. Se debe tener cuidado que no se hagan diferenciaciones excedentes ya que esto introduce dependencia entre datos que realmente

no existe. Por ejemplo, $X_t = w_t$ es una serie sin correlación, pero $\Delta X_t = w_t - w_{t-1}$ es un modelo de media móvil de orden uno ($MA(1)$).

Además de las gráficas de tiempo, la ACF puede ayudar a indicar si la diferenciación es necesaria. Un lento decaimiento es una indicación de que puede ser necesaria la diferenciación. Cuando los valores preliminares de d han sido encontrados, el siguiente paso es tomar las gráficas de ACF y PACF, para $\Delta^d X_t$ y se usa como guía la Tabla 3.1. Lo anterior para estimar los valores preliminares de p y q. Tomando en cuenta que si $p = 0$ y $q > 0$ la ACF se "corta" (deja de tener valores significativos) después del retraso q, en tanto que la PACF disminuye. Por otro lado si $q = 0$ y $p > 0$, el PACF se "corta" después del retraso p, y el ahora la ACF disminuye. Finalmente si $p > 0$ y $q > 0$, tanto la gráfica de la función de autocorrelación como la de la función de autocorrelación parcial irán en descenso. No es necesario ser tan preciso en esta etapa de la construcción del modelo, incluso en muchos casos se pueden tener valores evidentes para p y q.

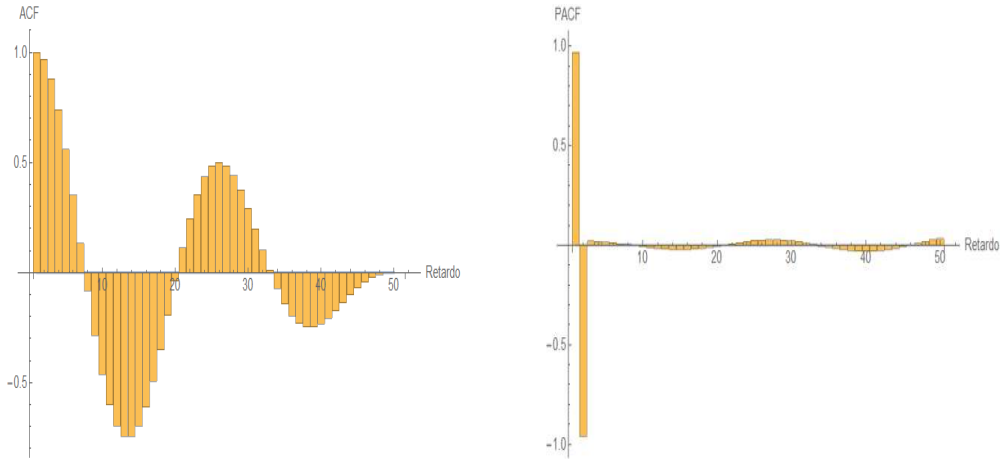
	$AR(p)$	$MA(q)$	$ARMA(p,q)$
ACF	Disminuye	"Corta" después de retraso q	Disminuye
PACF	"Corta" después del retraso p	Disminuye	Disminuye

Tabla 3.1: Tendencia de las ACF y PACF para los modelos AR,MA y ARMA

Para ilustrar lo anterior considere que se tienen 50 muestras (que corresponden a dos ciclos de la función) de una serie de tiempo generada a partir de una función senoidal con frecuencia angular igual 2π. Entonces la ACF y la PACF tienen la forma que se observa en las Figuras 3.1(a) y 3.1(b), donde se aprecia que el proceso más adecuado para modelar esta serie de tiempo es un AR ya que la ACF disminuye y la PACF se corta después de un retraso $p = 2$.

3.2.2. Redes neuronales artificiales para pronóstico de series de tiempo

Las aplicaciones de una red neuronal artificial (en inglés Artificial Neural Network) (ANN) se realizan en diversas áreas del conocimiento. las ANN sirven para modelar sistemas, pero con la característica de que se tiene un modelo implícito para representarlos. Debido a



(a) *ACF* **(b) *PACF***

Figura 3.1: ACF y PACF para una función senoidal

lo anterior también tienen aplicación en el pronóstico ya que pueden capturar la dinámica de las series de tiempo, incluso cuando los datos se originaron de un proceso complejo, por ejemplo de sistemas no lineales. A continuación se da un panorama general de como construir una red neuronal para la estimación de situaciones futuras en series temporales. Existe una gran variedad de arquitecturas de ANN, sin embargo solo se menciona la Feedforward, que es la más utilizada en pronóstico de series de tiempo.

Concepto y construcción

Una red neuronal artificial (en inglés Artificial Neural Network) es una estructura diseñada para resolver ciertos tipos de problemas, tratando de emular la forma en que el cerebro humano lo haría. La forma general de una red neuronal es una "caja negra"(un modelo implícito) que es usada para modelar problemas con alta dimensionalidad y con comportamientos no lineales ([Douglas C. Montgomery, 2008]). Las ANN tienen una amplia gamma de aplicaciones entre las cuales destacan: procesamiento de leguaje natural, compresión de imágenes, reconocimiento de caracteres, reconocimiento de patrones en imágenes, problemas de combinatoria, pronóstico de series de tiempo, modelado de sistemas y filtrado de ruido, entre otros.

Existen dos tipos básicos de redes neuronales artificiales las de una capa y las conocidas como multicapa. En cualquier caso, las entradas del proceso se reciben en la capa inicial, posteriormente al ser transferidas a la(s) capa(s) oculta(s) son ponderadas por coeficientes $\boldsymbol{W}$, que corresponden a la fortaleza de las conexiones entre neuronas. Las neuronas de la capa oculta cumplen con dos funciones, 1) combinan de alguna manera todas las entradas ponderadas por lo coeficientes de las conexiones, 2) aplican una función conocida como de activación, la cual permite mapear los términos a un solo valor que se encuentre dentro de un rango preestablecido. En este sentido la función de activación más común es la función sigmoide o sigmoidal, que se define como $Y = \frac{1}{1+e^{-X}}$; esta función mapea siempre al rango de valores $[0, 1]$. En seguida los resultados de las neuronas de la capa oculta pasan a la capa de salida, la cual puede tener una o múltiples neuronas, en este punto también se pondera la contribución de cada neurona de la capa oculta para generar las salidas correspondientes. En esta notación X representa las entradas y Y las salidas de la red neuronal artificial (en inglés Artificial Neural Network).

El algoritmo de propagación hacia atrás es usado para entrenar redes multicapa, su aceptación se debe principalmente a que permite calcular los pesos de todas las capas ocultas. En este método a la red se le presentan parejas de patrones, una entrada dada con su respectiva salida deseada. Por cada nueva pareja se ajustan los pesos, que parten de valores iniciales, de manera que disminuya el error entre la salida real y la respuesta de la red. El algoritmo de retropropagación no es otra cosa que un método de descenso de gradiente donde se minimiza una función de costo, presentando individualmente todos los pares de aprendizaje del error; así los parámetros son modificados iterativamente. Desde el punto de vista de minimización el problema es no lineal en los parámetros. Con la intención de evitar un sobreajuste, se debe eligir una red no demasiado grande. Para mayor información sobre el algoritmo de propagación hacia atrás y en general sobre las ANN se pueden consultar [Bishop, 1995], [Olabe, 1998] y [Duda u. a., 2012].

Aplicación en series de tiempo

Según [Kantz und Schreiber, 2004] en el entorno del pronóstico de series de tiempo, las ANN son generalmente usadas en lugar de construir un modelo formal. Es decir se busca

desarrollar un sistema de conocimiento subyacente que podría ser requerido al plantearse un procedimiento analítico. Si se puede tener una respuesta satisfactoria con una red neuronal pierde sentido buscar un modelo estadístico para cumplir con este objetivo.

Las redes neuronales más usadas como predictores son las conocidas como Feed-forward. De esta manera se usan p variables de entrada $X_1, X_2, \ldots, X_p$ para obtener una o más variables de salida. Las variables de entrada pueden ser al igual que en los modelos estadísticos, datos del pasado de la serie de tiempo, pero también variables externas. En la Figura 3.2 se muestra una estructura de una red neuronal con una sola capa oculta, donde se toma algunos valores del pasado de la serie de tiempo y de otras variables externas. En este caso la red neuronal funge como un predictor un paso a futuro (en inglés One Step Ahead) (OSA) ya que la salida solamente es un punto de la serie de tiempo.

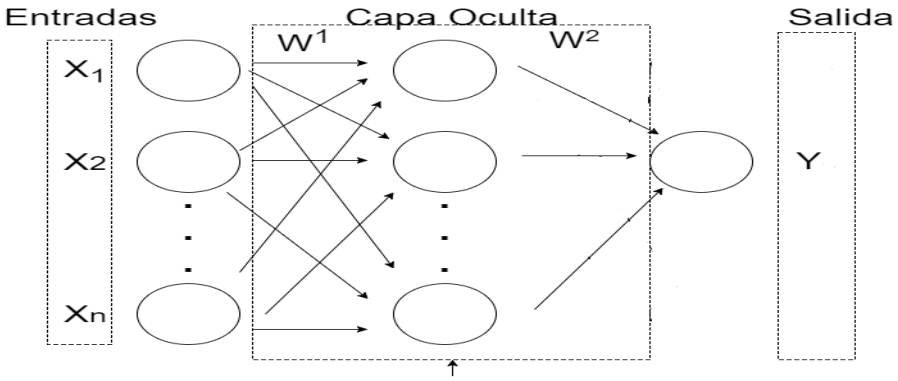


Figura 3.2: Ejemplo de una red neuronal de una capa aplicada en pronóstico de series de tiempo

En el caso de que se quisiera diseñar una red que calcule n pronósticos a futuro basta con definir ese mismo número de salidas en la red. La estructura de la red neuronal se diseña en base a la serie de tiempo con la que se trabaje. Se pueden tomar ciertas ideas de los modelos clásicos con la finalidad de determinar cuales serán las entradas de la red. Por ejemplo, basándose en un modelo AR se pueden considerar los p valores pasados de la serie de tiempo, o considerando los modelos ARMA se pueden tomar los p valores anteriores de la serie de tiempo y los q datos del error de pronóstico. En cuanto al número de capas ocultas y las interconexiones que existirán entre sí, generalmente se diseña a mano. Se puede empezar usando redes simples con una sola capa oculta, pero no existen reglas determinadas para la

selección de la arquitectura de la red.

Una forma muy atractiva de realizar estimaciones a futuro usando una red neuronal es aplicando el concepto de vectores de retardo. Este tema se aborda más adelante en la Subsección 3.3.3. Pero la idea básica es tomar solo los puntos del pasado que aporten información relevante para calcular el pronóstico. Más información sobre el uso de las ANN en el pronóstico de series de tiempo se puede encontrar en [Kantz und Schreiber, 2004] y [Douglas C. Montgomery, 2008].

3.3. Series de tiempo caóticas y pronóstico

El análisis estadístico presentado en el apartado anterior no es el más adecuado para tratar series de tiempo que se originan a partir de sistemas no lineales. En esta tesis mencionaremos únicamente la parte que se refiere a series de tiempo que se originan de sistemas caóticos. A continuación se mencionan conceptos generales de los sistemas caóticos y posteriormente se aborda la perspectiva del espacio de fase para analizar series con presencia de caos. Finalmente se introduce el método de vecinos cercanos que hace uso de las técnicas presentadas en el análisis del espacio de fase y representa uno de los métodos de pronóstico más efectivos usados en sistemas caóticos.

3.3.1. Sistemas caóticos

Una aparente paradoja es que el caos es determinista, generado por reglas fijas que no implican por sí mismas elementos de cambio. En principio, el futuro está completamente determinado por el pasado. Sin embargo en la práctica se amplifican las pequeñas incertidumbres, al igual que los errores diminutos de las mediciones y estos se acumulan en cálculos, con el efecto de que aún cuando el comportamiento es predecible a corto plazo, es impredecible a largo plazo. A continuación en las Definiciones 11, 12 y 13 se exponen (de manera general) las condiciones para decir que un sistema es caótico y en la Definición 14 se precisa lo que implica la presencia de caos en un sistema. [Olivares Caballero, 1994].

Definición 11 *Un conjunto* $\boldsymbol{D}$ *se dice que es denso en un espacio normado* $\boldsymbol{X}$ *si para cada elemento* $x \in \boldsymbol{X}$ *y cada* $\epsilon > 0$ *existe* $\boldsymbol{d} \in \boldsymbol{D}$ *con* $\|x - d\| < \epsilon$. *Equivalentemente, un*

subconjunto $\boldsymbol{D'}$ *de* $\boldsymbol{X}$ *es denso en* $\boldsymbol{X}$ *si la cerradura de* $\boldsymbol{D}$ *es* $\boldsymbol{X}$*), es decir, que* $\boldsymbol{D}$ *tiene intersección con todos los conjuntos abiertos diferentes de cero que pertenecen a* $\boldsymbol{X}$*.*

Como ejemplo para comprender lo que significa que un conjunto sea denso, tomemos como referencia a los números reales. Entre dos números cualquiera pertenecientes a este conjunto habrá también una cantidad infinita de números, es decir, tienen la misma cardinalidad.

Definición 12 *Un mapeo* $f: J \to \boldsymbol{J}$ *se dice que es topológicamente transitivo si para cualquier par de conjuntos abiertos* $\boldsymbol{U}, \boldsymbol{V} \subset \boldsymbol{J}$ *existe* $k > 0$ *tal que* $f^k(\boldsymbol{U}) \cap \boldsymbol{V} \neq \emptyset$*. Donde* $f^k()$ *denota aplicar iterativamente la función* f*,* k *veces.*

Que un mapeo sea topológicamente transitivo quiere decir que puntos que se encuentran en una vecindad arbitrariamente pequeña se mueven a otra. Esto se cumple bajo iteración en sistemas discretos o transcurrido un tiempo en sistemas continuos.

Definición 13 *Un mapeo* $f: J \to \boldsymbol{J}$ *posee dependencia sensitiva a las condiciones iniciales si existe* $\boldsymbol{\delta} > 0$ *tal que, para cualquier* $x \in \boldsymbol{J}$ *y cualquier vecindad* $\boldsymbol{V}$ *de* x*, existe* $y \in \boldsymbol{V}$ *tal que* $|f^n(x) - f^n(y)| > \boldsymbol{\delta}$*. Donde* $f^n()$ *denota aplicar iterativamente la función* f*,* n *veces para con* $n \geq 0$*.*

Intuitivamente, un mapeo tiene dependencia a las condiciones iniciales si existen puntos arbitrariamente cercanos a x (una vecindad) que eventualmente se separan al menos $\boldsymbol{\delta}$ al transcurrir el tiempo. En realidad no todos los puntos cercanos a x necesitan separarse, pero al menos uno de los que se encuentre en la vecindad lo hace. En la Figura 3.3 se muestra el sistema de Lorenz, en el cual se puede ver que después de algún tiempo la diferencia (ante condiciones iniciales distintas) se hace muy evidente. En el caso de la señal azul tuvo condiciones iniciales $x = 0.0, y = 1.0, z = 0.0$ y la naranja $x = 0.3, y = 1.2, z = 0.1$ entre más tiempo transcurre, el sistema tiende a mostrar comportamientos cada vez más distanciados. Para este caso se integró el sistema hasta un tiempo límite de 100 segundos y con pasos de integración de 0.1 segundos.

Definición 14 *Sea* $\boldsymbol{J}$ *un conjunto. La función* $f: J \to \boldsymbol{J}$ *se dice que es caótica en* $\boldsymbol{J}$ *si:*

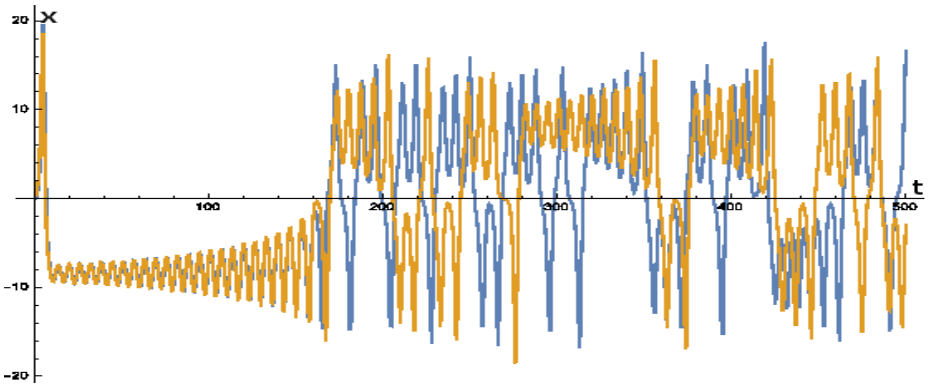


Figura 3.3: Ejemplo de sensibilidad a las condiciones iniciales para el sistema de Lorenz

1 f tiene dependencia sensitiva a las condiciones iniciales.

2 f es topológicamente transitiva.

3 El conjunto de puntos periódicos de f es denso en $\boldsymbol{J}$.

Asociado a la función f, existe un sistema dinámico el cual se dice caótico si la función f lo es.

En general, se puede decir que en un sistema caótico residen tres características principales. No es predecible debido a la dependencia de las condiciones iniciales. No puede ser descompuesto en subsistemas debido a la transitividad topológica. Aún así tiene una componente de regularidad debido a que los puntos periódicos son densos. [Peitgen u. a., 2006], [Alligood u. a., 2006]

Exponentes de Lyapunov

Considerando la sensibilidad a las condiciones iniciales es fácil notar que conforme transcurre el tiempo dos trayectorias que seguiría un sistema pueden cambiar considerablemente. Un sistema predominantemente periódico tendría una evolución que difiere muy poco. En contraparte un sistema con presencia de caos tiene una divergencia muy rápida, esta separación es exponencial. El exponente de este incremento es característico para el sistema subyacente a los datos y cuantifica la fuerza del caos. Se llama el exponente de Lyapunov y es usado para medir el nivel de caos presente en un sistema.

Existen varios tipos de exponentes de Lyapunov para sistemas dinámicos que se pueden cosiderar en el espacio de fase. El de mayor utilidad es conocido como máximo exponente de Lyapunov, el cual se define como se observa en la Definición 15.

Definición 15 *Dados dos vectores S_1 y S_2 (en el espacio de fase) que tienen una distancia $\| S_1 - S_2 \| \delta_0 \ll 1$. Se denota δ_n la distancia a un tiempo n a futuro entre dos trayectorias que emergen de esos puntos, es decir, $\delta_n = \| S_{1+n} - S_{2+n} \|$. Entonces λ_m es determinado por* (3.30) *y es conocido como el máximo exponente de Lyapunov.*

$$\delta_n \simeq \delta_0 \exp^{\lambda_m n},\ \delta_n \ll 1,\ n \gg 1. \tag{3.30}$$

Si λ_m es positivo significa que la divergencia exponencial entre dos trayectorias cercanas existe y por lo tanto el sistema tiene presencia de caos, en el caso donde $\lambda_m = \infty$ se dice que la señal es ruido. Para mayor información se pueden consultar [Kantz und Schreiber, 2004]. En el Capítulo 6 se muestran los máximos exponentes de Lyapunov para los diferentes casos de estudio.

3.3.2. Análisis del espacio de fase

El espacio de fase o diagrama de fase es una representación de los estados de un sistema. La dimensión del espacio de fase está dada por la dinámica del sistema. Esta representación permite observar la evolución del sistema, pero también puede ayudar a determinar que tipo de sistema se analiza. La dinámica de los sistemas continuos se modela mediante ecuaciones diferenciales, en el caso de los sistemas discretos es mediante las ecuaciones de diferencia. En cualquiera de los dos casos cuando se encuentra la solución de dichas ecuaciones, se obtiene una secuencia de puntos que parten desde las condiciones iniciales a un conjunto de valores que se les conoce como trayectoria o flujo en el caso continuo y órbita en el caso discreto.

A partir de las trayectorias u órbitas se pueden plantear los conceptos de la Definición 16. Estos conjuntos límite determinan la dinámica del sistema, al analizarlos se

pueden conocer características importantes del sistema entre las que podrían encontrarse si es lineal, estable o caótico, entre otras.

Definición 16 *Un punto y, es un punto límite de x, si para toda vecindad $\boldsymbol{V}$ de x, y entra repetidamente a $\boldsymbol{V}$ cuando $t \to \infty$. El conjunto $\boldsymbol{L}(x)$ de todos los puntos límite de x es llamado el conjunto límite de x.*

En base a la definición anterior se conocen actualmente cuatro tipos de conjuntos límite.

- Puntos de equilibrio: Es un objeto de dimensión cero en espacio de fase, relacionado con sistemas que tienden al reposo después de un transitorio. Un punto x^* es un punto de equilibrio si y sólo si $f(x^*) = 0$. Los puntos de equilibrio se dan cuando el máximo exponente de Lyapunov es menor a cero $\lambda_m < 0$.

- Ciclos límite: Es una trayectoria u órbita cerrada, representada por un objeto unidimensional asociado con un estado estable periódico. En este caso el exponente de Lyapunov es nulo, es decir, $\lambda_m = 0$.

- Soluciones cuasi-periódicas: En donde dos o más periodicidades pueden ser identificadas y relacionadas por números irracionales. Para dos periodicidades el conjunto límite es una superficie conocida como toroidal de dos dimensiones.

- Atractores extraños: Es un objeto geométrico en el espacio de fase, en el cual convergen trayectorias caóticas. El atractor de un sistema caótico no es un objeto geométrico simple, como un círculo o un toroide. Sino que es un poco más complicado geométricamente, en general tienen formas de fractales, los cuales poseen dimensión fraccionaria. EN este caso existe presencia de caos por lo tanto $0 < \lambda_m < \infty$.

Sistemas caóticos comunes

A continuación se mencionan algunos sistemas que son muy conocidos en el contexto del caos. Si bien son sistemas sintéticos (no tienen una interpretación real o son modelos simplificados de fenómenos reales), son de gran utilidad en el ámbito de series de

tiempo. Antes de abordarlos se mencionan algunos sistemas reales que presentan también este comportamiento; entre los cuales destacan el sistema de un péndulo doble, el tiempo meteorológico, las señales de los EEG y ECG ó también algunas dinámicas de poblaciones.

El primer sistema sintético con presencia de caos es conocido como de Lorenz (ver [Lorenz, 1963]). El cual es un sistema en tiempo continuo, constituido por tres ecuaciones diferenciales. Este sistema surgió como una simplificación de la dinámica de la atmósfera terrrestre. El sistema de Lorenz tiene la forma de (3.31), en donde para los valores $\sigma_L = 10, \rho_L = 28, \beta_L = 8/3$ exhibe un comportamiento caótico.

$$\begin{aligned} \frac{dx}{dt} &= \sigma_L(y - x) \\ \frac{dy}{dt} &= x(\rho_L - z) - y \\ \frac{dz}{dt} &= xy - \beta_L z \end{aligned} \tag{3.31}$$

El segundo sistema es en tiempo discreto y está dado por dos ecuaciones de diferencias (ver [Hénon, 1976]). Depende de dos parametros α_L y β_L, donde estas constantes toman los valores 1.4 y 0.3, respectivamente. El sistema se conoce como de Henon y viene descrito por (3.32).

$$\begin{aligned} x_{n+1} &= 1 - \alpha_H x_n^2 + y_n \\ y_{n+1} &= \beta_H x_n \end{aligned} \tag{3.32}$$

El tercer sistema que se menciona es llamado de Rossler (ver [Rössler, 1976]), es generado por un sistema en tiempo continuo, formado por un conjunto de tres ecuaciones diferenciales no lineales expresadas en (3.33). Para los valores $\alpha_R = 0.1, \beta_R = 0.1, \gamma_R = 14$ este sistema tiene un comportamiento caótico.

$$\begin{aligned} \frac{dx}{dt} &= -y - z \\ \frac{dy}{dt} &= x + \alpha_R y \\ \frac{dz}{dt} &= \beta_R + z(x - \gamma_R) \end{aligned} \tag{3.33}$$

El último sistema que se menciona es conocido como de Mackey-Glass (ver [Glass und Mackey, 2010]), es un sistema no lineal con una ecuación diferencial de retardos, que tiene la forma de (3.34). Para los valores $\beta_M = 0.2, \gamma_M = 0.1, \tau_M = 17, n_M = 10$ exhibe comportamiento caótico.

$$\frac{dx}{dt} = \beta_M \frac{x(t-\tau_M)}{1 + x^{n_M}(t-\tau_M) - \gamma_M x} \tag{3.34}$$

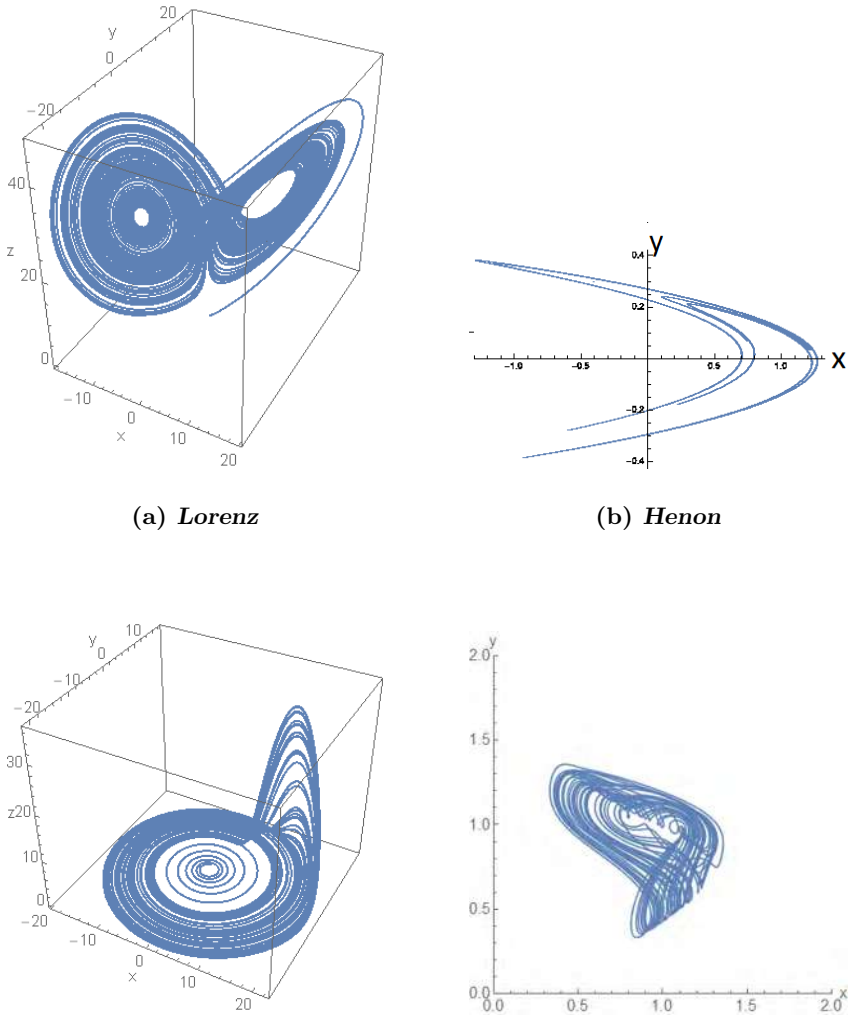
En las Figuras 3.4(a), 3.4(b), 3.4(c), 3.4(d) se observa la representación en espacio de fase para Lorenz, Henon, Rossler y Mackey-Glass. En esta última su representación se obtiene al graficar $x(t)$ contra $x(t-\tau_M)$.

En la Figura 3.4 se aprecian las representaciones en espacio de fase para los sistemas caóticos sintéticos más conocidos. Se puede observar en todos los casos que las trayectorias son atractores extraños en donde la evolución del sistema orbita determinadas regiones en el espacio de fase. La presencia de caos en estos sistemas hace que la trayectorias en el espacio de fase nunca vuelvan a tomar un valor que ya se haya presentado y esto se puede ver en las cuatro subfiguras.

Tiempo de retardo y algoritmo de información mutua

Habiendo subrayado la importancia del espacio de fase para el estudio de sistemas con propiedades determinísticas, debemos afrontar el primer problema: lo que observamos en un experimento no es un objeto de espacio de fase sino una serie de tiempo, muy probablemente sólo una secuencia de medidas escalares. Por lo tanto, tenemos que convertir las observaciones en vectores de estado. Este es el problema importante de la reconstrucción del espacio de fase que técnicamente se resuelve mediante el método de los retardos (o construcciones relacionadas). [Kantz und Schreiber, 2004]

Las series de tiempo pueden ser vistas como una secuencia de medidas escalares de alguna variable, las cuales dependen del estado actual del sistema que las genera. Por lo general se toman muestras a intervalos fijos de tiempo. Es decir, apreciamos el sistema a través de una función de medición $s(x(t))$ y nuestras observaciones pueden presentar variaciones aleatorias denotada por η_t. Esto se representa según(3.35).



(a) ***Lorenz*** **(b)** ***Henon***

(c) ***Rossler*** **(d)** ***Mackey-Glass***

Figura 3.4: Diagramas en espacio de fase para las series sintéticas

$$X_t = s(x(t)) + \eta_t \tag{3.35}$$

donde X_t representa la observación que se almacena en la serie de tiempo, η_t es la incertidumbre que contiene la medición y x_t representa el estado del sistema para el tiempo t.

En [Kantz und Schreiber, 2004] se plantea que la reconstrucción del espacio de fase se puede hacer por medio de retardos, considerando m dimensiones en el espacio de fase. Se supone que a partir de la variable dada se obtendrán las demás variables en el espacio de fase. Entonces se pueden formar los vectores S_t, dados por (3.36), donde la diferencia de tiempo entre una muestra y otra es denominada τ y se conoce como el tiempo de retardo. Estas diferencias denotan las derivadas (por medio de las cuales se obtienen las otras variables de estado). Por ejemplo, se la dimensión real del sistema es 3 y de alguna manera se calcula ese valor, al hacer la reconstrucción por medio de los vectores de retardo realmente se están calculando las variables de estado representadas por las diferencias $X_t - \tau$ y $X_t - 2\tau$.

$$S_t \;=\; \left\{X_{t-(m-1)\tau}, X_{t-(m-2)\tau}, \ldots, X_{t-\tau}, X_t\right\}. \tag{3.36}$$

Aquí se debe notar que para $\tau > 1$ solamente la ventana de tiempo cubierta para cada vector es incrementada, mientras que el número de vectores construidos para la serie de tiempo es aproximadamente el mismo. Lo anterior debido a que se crea un vector por cada observación, así que bajo circunstancias generales el atractor formado por los vectores de retardo, será equivalente al atractor que tendría el sistema original.

El tiempo de retardo se interpreta (en la reconstrucción del espacio de fase) como la escala de tiempo interna del sistema dinámico, es decir, se considera que el sistema cada determinado tiempo tiene un cambio lo suficientemente significativo como para capturar el estado en el que se encuentra. Es más difícil obtener una buena estimación del tiempo de retardo τ , que encontrar la dimensión de embebido. Ya que sistemas con la misma m pero diferente τ son equivalentes en el sentido matemático para datos sin ruido. Si τ es pequeño en comparación con la escala de tiempo interna del sistema, elementos sucesivos

de los vectores de retardo estarán fuertemente correlacionados. Por el contrario, si el valor es muy grande elementos sucesivos serán demasiado independientes y formarán una nube de puntos muy grande en el espacio $\mathbb{R}^m$. Una aproximación para calcular este parámetro se puede obtener por medio de la función de autocorrelación (ACF), según la Definición 3, en donde τ se encuentra en el primer cero de la función. Una posible limitación al determinar τ por medio de la ACF es que sólo considera la correlación lineal del sistema.

La técnica más utilizada para calcular un valor apropiado del tiempo de retardo es conocido como de información mutua. El método de información mutua se basa en la información que se conoce acerca de $X_{t+\tau}$ si conocemos X_t. En el intervalo de exploración de los datos se crea un histograma de resolución ϵ para la distribución de probabilidad de los datos. Denotando por P_i la probabilidad de que la señal asuma un valor dentro del i-ésimo compartimiento del histograma. Asimismo $P_{i,j}$ la probabilidad conjunta de que X_t esté en el compartimiento i y $X_{t+\tau}$ esté en el compartimiento j. Entonces la información mutua para el tiempo de retardo τ, se cálcula como se aprecia en (3.37) [Kantz und Schreiber, 2004].

$$I_\epsilon(\tau) \;=\; \sum_{i,j} P_{i,j}(\tau) ln(P_{i,j}(\tau)) - 2\sum_{i} P_i ln(P_i). \tag{3.37}$$

La forma que toma (3.37) se obtiene considerando el desarrollo planteado en [Fraser und Swinney, 1986] y además que se está calculando la entropía de una sola variable con respecto a ella misma. En el caso donde $\tau = 0$, las probabilidades conjuntas son $P_{i,j} = P_i\delta_{i,j}$ y la expresión se vuelve la Entropía de Shannon de la distribución de los datos. Es costumbre usar una resolución fija para particionar los datos, aún así existen algoritmos alternativos que usan particiones adaptables. En cualquier caso, el número de particiones debe ser relativamente grande ya que en muchas ocasiones no existe el límite cuando $\epsilon \to 0$. Para valores pequeños de τ, $I_\epsilon(\tau)$ será grande. Si se decrementa relativamente rápido en el límite superior de τ, X_t y $X_{t+\tau}$ no tendrán nada que ver el uno con el otro. Entonces $P_{i,j}$ puede ser factorizado como P_iP_j y la información mutua se vuelve cero. El primer mínimo de $I(\tau)$ denota el tiempo de retardo donde $X_{t+\tau}$ agrega la máxima información al conocimiento de X_t, en otras palabras la redundancia es menor.

La información mutua mide la dependencia general de dos variables, en tanto que la ACF mide la dependencia lineal, por lo que esta última está más limitada. La información mutua mide la incertidumbre de una variable si se conoce otra; este concepto está estrechamente ligado a la entropía de Shannon o de la información. La entropía mide la cantidad de información contenida en un mensaje y el aporte de información de cada símbolo al mensaje completo, pero al mismo tiempo mide la incertidumbre presente en una señal.

Al calcular de manera práctica este parámetro, normalmente se recurre a hacerlo visualmente por medio de gráficas que en su eje horizontal tengan un rango de valores posibles para τ. En tanto que en su eje vertical tengan el valor que regresa la función de entropía. Así se siguen dos criterios para calcular τ, el primero que se elija el primer mínimo de la función en el intervalo dado. La segunda que cuando se tiene una función que decrece monotónicamente el valor más adecuado para τ es uno. Para más información se pueden consultar [Fraser und Swinney, 1986] y [Kantz und Schreiber, 2004].

Dimensión de embebido y algoritmo de falsos vecinos cercanos

La interpretación de la dimensión de embebido en el espacio de fase es el número de variables de estado que intervienen en el sistema en cuestión. [Kantz und Schreiber, 2004] plantean que una vez que se ha obtenido τ, se debe calcular un valor adecuado para m. El algoritmo más utilizado para calcular la dimensión de embebido, se conoce como de falsos vecinos cercanos (en inglés False Nearest Neighbors) (FNN) (ver [Kennel u. a., 1992]).

Si se asume que la dinámica en el espacio de fase es representada por un campo vectorial, entonces los estados vecinos deberían estar sujetos a casi la misma evolución del tiempo. Así dos trayectorias que surjan de puntos muy cercanos, en un futuro cercano deberían ser similares. Lo anterior incluso cuando el caos incluye una divergencia exponencial entre ambos. La idea básica es buscar para los puntos en el conjunto de datos que son vecinos en el espacio de embebido, pero que no deben ser vecinos ya que su futura evolución temporal es demasiado diferente. Supongase que la dimensión de embebido adecuada para un sistema es m_0, si se analizan los mismos datos pero en una dimensión inferior, la transición de m_0 a m es una proyección, eliminando ciertos ejes del sistema coordenado. Por lo tanto, algunos puntos (de los cuales sus coordenadas han sido eliminadas) diferirán

fuertemente, pero por la proyección se convierten en vecinos cercanos (falsos).

Entonces por cada punto en la serie de tiempo se toman sus vecinos más cercanos en la dimensión m y se calcula la razón de las distancias entre esos puntos en las dimensiones m y $m+1$. Si la razón es más grande que un umbral preestablecido r se dice que los vecinos son falsos. Este umbral tiene que ser lo suficientemente grande para permitir la divergencia exponencial debida al caos determinista. Si se denota la desviación estándar de los datos por σ y se usa la máxima norma, el número de falsos vecinos cercanos se calcula como se describe en (3.38).

$$X_{fnn}(r) = \frac{\sum_{t=1}^{N-m-1} \Theta\left(\frac{|S_t^{(m+1)} - Sk_t^{(m+1)}|}{|S_t^{(m)} - Sk_t^{(m)}|} - r\right)\Theta\left(\frac{\sigma}{r} - |S_t^{(m)} - Sk_t^{(m)}|\right)}{\sum_{t=1}^{N-m-1} \Theta\left(\frac{\sigma}{r} - |S_t^{(m)} - Sk_t^{(m)}|\right)} \tag{3.38}$$

donde $Sk_t^{(m)}$ representa los vecinos cercanos de S_t en la dimensión m y la función Θ representa un escalón unitario. La primera función escalón del numerador es la unidad si los vecinos más cercanos son falsos ó si la distancia es incrementada en razón de un factor más grande que r cuando se incrementa la dimensión de embebido. Mientras que la segunda función escalón elimina a todos los pares cuya distancia inicial era mayor que σ/r. Estos por definición no pueden ser falsos vecinos, ya que en promedio hay suficiente espacio para distanciarse más lejos de σ. Estos son candidatos inválidos para el método, así que se eliminan, lo cual también se observa en la normalización.

Contrario a lo que se podría suponer, puede haber vecinos cercanos falsos aún cuando se está en la dimensión de embebido correcta. Lo anterior, debido al ruido en las mediciones; adicionalmente si hay más datos disponibles también podría incrementar la razón de falsos vecinos cercanos. Por último, al obtener la dimensión de embebido mediante este algoritmo, comúnmente se gráfica todo el rango de valores que puede tomar m contra el porcentaje de falsos vecinos cercanos. Finalmente se puede escoger el primer mínimo de la función, o bien, se toma el valor en el cual existe el mayor descenso en el porcentaje de falsos vecinos ([Kennel u. a., 1992], [Kantz und Schreiber, 2004]).

3.3.3. Algoritmo de pronóstico no lineal basado en vecinos cercanos

Como se mencionó en las secciones anteriores, con la finalidad de reconstruir el espacio de fase se deben estimar los valores de m y τ. De esta manera, se construyen a partir de estos valores, vectores de retardo. Basándose en la idea de los vectores de retardo (que contienen de alguna manera la información relevante de la serie de tiempo) se puede plantear un método de pronóstico. A continuación se menciona la adaptación hecha por [Kantz und Schreiber, 2004] del algoritmo de clasificación conocido como vecinos cercanos (consultar [Cover und Hart, 1967] o [Duda u. a., 2012]), en la estimación de valores futuros de una serie de tiempo. Este algoritmo es llamado algoritmo de pronóstico no lineal basado en vecinos cercanos (NN).

Con la finalidad de estimar el valor futuro del sistema X_{N+1}, dado el presente X_N, se busca una lista de todos los estados pasados X_t con $t < N$ para el más cercano a X_N en base a alguna regla predefinida. En el caso donde un estado (supongase un estado X_{n_0}) que se presenta en el tiempo t_0, es similar al presente X_N se dice que son cercanos en el espacio de fase. De esta manera se puede garantizar que X_{t_0+1} es cercano a X_{N+1}.

Si se observa el sistema por un largo periodo de tiempo, existirán estados en el pasado que son arbitrariamente cercanos al estado presente. Así la predicción $\hat{X}_{N+1} = X_{t_0+1}$ será relativamente cercana al valor real del estado donde se encontrará el sistema. El algoritmo resultante contará con m y τ como parámetros de entrada y el esquema de predicción tomará cada medición disponible $X_1, X_2 \ldots, X_N$ y calculará su vector de retardo como se expresa en (3.36) correspondiente. Sin embargo, para $t < (m-1)\tau$ no se pueden calcular sus respectivos vectores de retardo, porque no se cuenta con mediciones anteriores a $t = 1$.

Usando el enfoque de pronóstico OSA, para calcular la predicción $\hat{X}_{N+1}$, buscamos los vectores $S_{t_0} \in \mathscr{U}_R(S_N)$ (vectores más cercanos en el vecindario de S_N) y se usa X_{t_0+1} como predictor, tomando en cuenta que $t_0 < N$. Una última consideración es que al ser capturados los datos han sido contaminados con algún tipo de ruido; asumiendo que es un Ruido blanco Gaussiano con desviación estándar σ, se puede suponer que cualquier par de datos que tiene una variación de a lo más σ uno respecto al otro, son el mismo valor o estado. Con lo anterior se puede plantear formalmente el algoritmo de pronóstico no lineal

basado en vecinos cercanos (NN) (Algoritmo 1). Donde m podría calcularse con el método de falsos vecinos cercanos expresado en (3.38)) y τ por medio de la técnica de información mutua según (3.37). Adicionalmente se define un parámetro ϵ_r que determina el radio para el cual dos vectores de retardo se consideran vecinos.

Algoritmo 1 $NN\,(X, m, \tau, \epsilon_r)$

1: $N \leftarrow longitud(X)$
2: $S \leftarrow S_t = \left\{X_{t-(m-1)\tau}, X_{t-(m-2)\tau}, \ldots, X_{t-\tau}, X_t\right\}\ \forall t \in [(m-1)\tau, N-1]$
3: $X_{t_0} \leftarrow$**nulo**
4: $N_0 \leftarrow 0$
5: **para** $S_t \in S$, $(t \in [(m-1)\tau, N-1])$ **hacer**
6: **si** $\|S_N - S_t\| < \epsilon_r$ **entonces**
7: $\mathscr{U}_R(S_N) \leftarrow S_t$
8: $X_{t_0+1} \leftarrow X_{t_0+1} + X_{t+1}$
9: $N_0 \leftarrow N_0 + 1$
10: **fin si**
11: **fin para**
12: **si** $N_0 > 0$ **entonces**
13: $\hat{X}_{N+1} = \frac{X_{t_0}}{N_0}$
14: **si no**
15: $\hat{X}_{N+1} = X_N$
16: **fin si**
17: **devolver** $\hat{X}_{N+1}$

El algoritmo 1 recibe la serie de tiempo, en la Línea 2 la convierte en un conjunto de vectores de retardo (usando los valores de m y τ que se le proporcionan). Posteriormente, en las Líneas 5 a 11 verifica cuales vectores son vecinos cercanos del vector S_N (asociado a X_N), se almacenan en un arreglo $\mathscr{U}_R(S_N)$ y se incrementa el número de vecinos. Finalmente, en las Líneas 12 a 16 el pronóstico de X_{N+1} se cálcula como el promedio de los N_0 valores contenidos en X_{t_0+1} esto considerando que existen vecinos cercanos. Si no existe ningún vector que se asemeje a S_N, el pronóstico ($\hat{X}_{N+1}$) viene dado simplemente por X_N.

Si se desea hacer pronóstico a n pasos a futuro, el procedimiento a seguir es muy parecido. La diferencia radica en que no se toma el valor inmediato para cada vector contenido en $\mathscr{U}_R(S_N)$, o sea, cada término X_{t_0+1}. En su lugar se tomarían los n valores posteriores para cada vecino cercano. De esta manera cada pronóstico X_{N+k} (con $k = 1, 2, \ldots, n$) se calcula como el promedio de los términos X_{t_0+k} para cada vecino cercano contenido en $\mathscr{U}_R(S_N)$.

Para entender la representación que tienen m y τ en un problema real, considere que se tiene una serie de tiempo de temperatura, en la cual se tomarón las mediciones cada 10 minutos y supongamos que $m = 6$ y $\tau = 2$. Esto nos indica que un vector de retardo estaría compuesto por 6 tomando la actual y cinco anteriores, las cuales están separadas una de otra 20 minutos, así que un vector de retardo contendría de forma condensada la dinámica del sistema de los últimos 80 minutos. Realmente la dimensión de embebido nos sirve para determinar cuantos puntos del pasado son necesarios tomar para modelar un estado del sistema. En tanto que el tiempo de retado se usa para saber que tan distanciados deben estar esos puntos, de manera que aporten la información relevante más cercana que exista.

Existen modificaciones al algoritmo NN, en las cuales no se calculan m y τ en su lugar estos parámetros son optimizados mediante técnicas heurísticas basadas principalmente en inteligencia artificial. En este sentido se podrían usar GA, PSO o optimización por evolución diferencial (en inglés Diferencial Evolution) (DE). El procedimiento a seguir usando estas metaheurísticas es parecido. Por ejemplo, cuando se usa DE lo que se hace es crear un grupo de individuos iniciales, donde cada uno representa una instancia del algoritmo NN y tienen valores asignados para m, τ y ϵ_r. Posteriormente se evalua el desempeño de estos individuos (esto se hace por medio de alguna medida de error) y se pueden aplicar sobre ellos mutaciones y recombinaciones para obtener nuevos individuos. En este procedimiento se irán combinando los individuos de la población y seleccionando a los mejores para al final regresar el individuo que tuvo un mejor desempeño. Esto implica que se llegaría a un vector solución que contiene valores óptimos para m, τ y ϵ_r. Esta forma de obtener estimaciones futuras es conocida como algoritmo de pronóstico no lineal basado en vecinos cercanos optimizado con evolución diferencial (NNDE) y es planteada en [De La Vega u. a.,

2014].

Conclusiones del capítulo

En este capítulo se dió una descripción de los conceptos más relevantes relacionados con las series de tiempo. Entre ellos, las componentes de una serie de tiempo, las series de tiempo estacionarias, la función de autocovarianza, la ACF, la PACF, éstas constituyen las herramientas básicas usadas para hacer un análisis de las series de tiempo. Posteriormente, se abordaron los enfoques lineales más usados en el pronóstico, los modelos AR, MA, ARMA y ARIMA. Donde destacaban los modelos ARIMA por ser los más generales. En general estos modelos se basan en tomar puntos anteriores de la serie de tiempo y hacer una combinación de ellos para estimar los puntos posteriores. Después se mencionaron brevemente las ANN por su relevancia en el pronóstico, donde destaca el hecho de que son usadas en problemas no lineales y no requieren un modelo explícito del problema. Finalmente se presentó el análisis en espacio de base, donde se introdujeron conceptos como series de tiempo caóticas, la dimensión de embebido (m), el tiempo de retardo (τ) y el algoritmo NN así como su modificación NNDE. Este algoritmo resalta por un lado al ser intuitivo, ya que se basa en la idea de que situaciones actuales pueden paracerse a situaciones anteriores. Por otro lado, es bastante robusto debido a que parte de la teoría del espacio de fase y puede modelar sistemas dinámicos que no necesariamente son lineales.

Capítulo 4

Lógica difusa y teoría de conjuntos difusos

4.1. Introducción a la lógica difusa

Los seres humanos constantemente toman decisiones a partir de la información que tienen disponible, aunque en muchos casos esta es deficiente. Por ejemplo, no siempre está completa, en otras ocasiones es imprecisa o tiene cierta incertidumbre. Sin embargo, la manera en que razonamos permite trabajar con información limitada y aún así se obtienen resultados aceptables en un tiempo relativamente corto. La lógica difusa se desarrolla como una herramienta que pretende capturar esta capacidad que existe en el razonamiento humano (procesar información inexacta). Surge como una extensión de la lógica clásica o nítida, buscando remediar las limitaciones que tiene de forma inherente el expresar el conocimiento de forma tajante.

4.1.1. Transición de la lógica clásica a la lógica difusa

La lógica clásica surge de la necesidad del ser humano de clasificar objetos y conceptos. Se basa en conjuntos conocidos como nítidos, a partir de los cuales captura el conocimiento y expresa relaciones entre términos, con la finalidad de realizar inferencias o razonamientos. Según [Klir und Yuan, 1995] estos conjuntos se pueden definir mediante tres

formas, 1) se listan todos los elementos del conjunto ($A = \{a_1, a_2, \ldots, a_n\}$), 2) se evalúa si se satisface una propiedad de membresía ("x_1 tiene la propiedad P", $A = \{x_1 \,|\, P(x_1)\}$), 3) mediante una función indicadora. Esta función denota de forma condensada si un elemento dado pertenece o no al conjunto en cuestión, $\chi_A : x_1 \to \{0, 1\}$.

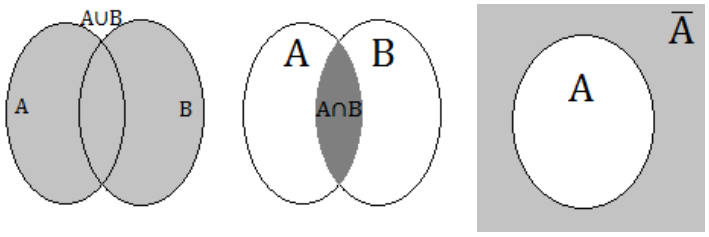
$$\chi_A(x_1) = \begin{cases} 1 \text{ para } x_1 \in A \\ 0 \text{ para } x_1 \notin A \end{cases}$$

Usando cualquiera de las tres formas se pueden expresar los conjuntos, para lograr lo anterior, se han definido una serie de operaciones entre los conjuntos clásicos (consultar [Klir und Yuan, 1995] ó [Chen und Pham, 2000]). De manera general son: la unión, para dos conjuntos representa todos los elementos que existen dentro del primero o del segundo. La intersección, que representa la región de elementos que pertenecen al mismo tiempo a dos conjuntos. El complemento representa todos los elementos que no pertenecen a un conjunto. Estas operaciones se expresan matemáticamente en (4.1),(4.2) y (4.3), respectivamente y se pueden apreciar visualmente en la Figura 4.1.

$$A \cup B = \{x_1 \,|\, x_1 \in A \vee x_1 \in B\} \tag{4.1}$$

$$A \cap B = \{x_1 \,|\, x_1 \in A \wedge x_1 \in B\} \tag{4.2}$$

$$\bar{A} = \{x_1 \,|\, x_1 \notin A\} \tag{4.3}$$



(a) ***Unión*** **(b)** ***Intersección*** **(c)** ***Complemento***

Figura 4.1: Operaciones en Conjuntos Clásicos

Con el objetivo de inferir hechos a partir de otros, considerados como sus antecedentes, estas operaciones de conjuntos se modifican de manera que se puedan usar como

conectivas entre enunciados. Si se parte de tres enunciados p_A, q_B y r_C (estos enunciados pueden ser proposiciones simples o compuestas). Se define $\neg p_A$ como la negación de p_A, que está asociada al complemento, también se pueden definir las operaciones $p_A \vee q_B$ y $p_A \wedge q_B$; la primera relaciona los enunciados de manera que se puede dar p_A, q_B o ambos. La segunda hace que la relación ligue los enunciados de manera que solo cuando ambos son verdaderos la relación lo es. Estas operaciones, conocidas como disyunción y conjunción, son equivalentes a la unión y a la intersección de conjuntos, respectivamente.

Existen otras conectivas usadas en lógica clásica, pero las de nuestro interés ahora son las tres mencionadas así como la condicional material o implicación (consultar [Castillo, 1999] para mayor información). Como ejemplos de la forma que pueden tomar las implicaciones se tiene $p_A \rightarrow r_C$, $p_A \wedge q_B \rightarrow r_C$, $p_A \wedge \neg q_B \rightarrow r_C$, o cualquier otra combinación de estos operadores. Las conectivas se usan como funciones de verdad, es decir, reciben las proposiciones como términos que solo pueden pertenecer a los conjuntos verdadero o falso y de manera similar dichas funciones solo devuelven valores verdaderos o falsos. Para poner de relieve como se formulan proposiciones en la lógica convencional y posteriormente como se hacen inferencias, tómese en cuenta el siguiente argumento.

Se denomina p_A a la proposición "Hoy es un día de verano" y q_B al enunciado "Los días de verano llueve". Estas dos sentencias son llamadas premisas y se conoce como conclusión a la aseveración que se puede obtener a partir de las premisas y al conjunto de premisas y conclusiones se le conoce como argumento, así la conclusión $p_A \wedge q_B \rightarrow r_C$ que se puede obtener sería "Hoy llueve".

Hasta este punto se puede pensar que la lógica clásica es suficiente al modelar el lenguaje, sin embargo existen diversos casos en que queda limitada. Una parte sumamente importante del lenguaje y el razonamiento humano es que se ponderan de alguna forma las expresiones y clasificaciones que se hacen. Por ejemplo, si se consideran las siguientes proposiciones "hoy hace mucho calor", "este año será bueno para la economía" y "Juan es muy alto"; en todos los casos se usa un término que hasta cierto punto cuantifica un hecho. Esto también permite algo de subjetividad al calificador, ya que no existe un límite perfectamente definido para decir que alguien es "muy alto" o no hay una temperatura preestablecida que indique "mucho calor". Por lo anterior, se observa que la lógica convencional puede

aceptar estas premisas, sin embargo, no puede modelar esa incertidumbre en los conceptos. Es decir, con la teoría clásica las premisas y conclusiones se limitan a ser completamente verdaderas o totalmente falsas.

4.1.2. Aplicaciones de la lógica difusa

Las principales aplicaciones de la lógica difusa se enfocan en situaciones tales como:

- En procesos complejos, si no existe un modelo de solución que use conocimiento objetivo, o bien si no existe un modelo sencillo.
- Cuando se debe introducir la experiencia de un operador experto y esté basada en conceptos imprecisos.
- Cuando ciertas partes del sistema que se analiza son desconocidas o no se pueden medir de forma fiable.
- Cuando se quieran controlar determinadas variables de un sistema, pero al mismo tiempo eso implique que se modifiquen desfavorablemente otras.
- Cuando se representen conceptos imprecisos, o se opere con cierta incertidumbre en el lenguaje.

Se puede decir que la lógica difusa se usa con dos finalidades principales, la primera es tratar situaciones que involucran una alta complejidad, en donde posiblemente no se tenga entendido completamente el comportamiento del sistema. La segunda en situaciones donde se requiera tener una solución rápida, pero que aún así sea aproximada. Por lo mencionado anteriormente, las aplicaciones más importantes de la lógica difusa se han dado en áreas multidisciplinarias. En el control de sistemas ha tenido una amplia aplicación en vehículos (automóviles, trenes, aviones y helicópteros), también en el control de plantas de generación de energía, procesos térmicos y en procesos industriales. Otro ámbito importante en el que se le ha dado uso es en la predicción y optimización. Como ejemplo de lo anterior, en el Capítulo 2 (revisión del estado del arte) se observó que se usa ampliamente (aunque no puramente) en el pronóstico. También se ha usado en entornos computacionales como son el

reconocimiento de patrones, sistemas de información y visión computacional, por mencionar algunos.

4.1.3. Conceptos preliminares de lógica difusa

La lógica difusa es una forma de lógica multivaluada, en la que se utilizan conjuntos, los cuales admiten un grado de pertenencia a ellos ([Acosta, 2006]). Este grado de pertenencia está dado por un valor dentro del rango $[0, 1]$. Puede considerarse una generalización de la lógica convencional, ya que la lógica clásica denota la pertenencia a un conjunto solo por dos posibles valores $\{0, 1\}$. En la Figura 4.2 se puede apreciar de forma gráfica la diferencia entre los conjuntos difusos y los clásicos. En los conjuntos tradicionales se dice que un elemento pertenece (puntos en verde), o no pertenece al conjunto (puntos en rojo). En cambio, en los conjuntos difusos existe un grado de pertenencia, los puntos en verde simbolizan elementos que pertenecen completamente al conjunto y los puntos en rojo los que tienen una pertenencia nula. Sin embargo los puntos en verde olivo, amarillo y naranja simbolizan precisamente ese grado de pertenencia, entre más parecidos son al verde tienen una pertencia mayor, de la misma forma entre más cercano al rojo tiene una pertenencia menor.

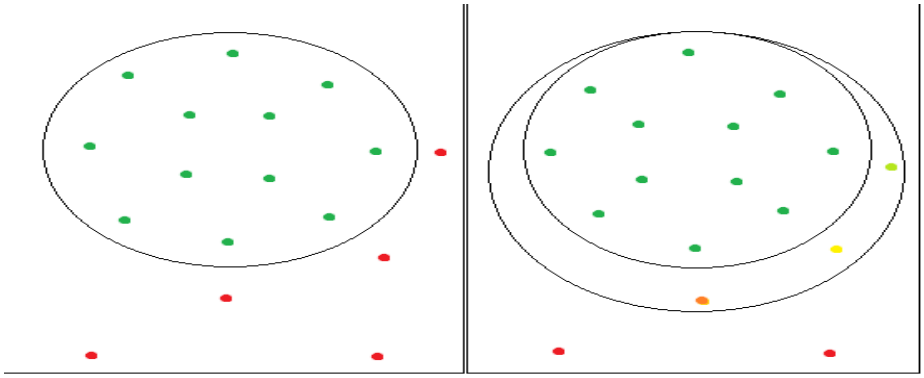


Figura 4.2: Comparación conjuntos clásicos $\{0, 1\}$ y difusos $[0, 1]$

En analogía con la lógica clásica, aquí también se pueden representar los conjuntos de diversas maneras, solo que ahora cada elemento debe quedar expresado por dos términos. El primero denota el elemento y el segundo su pertenencia al conjunto en cuestión. Antes

de introducir este concepto se darán algunas definiciones útiles, basadas en [Klir und Yuan, 1995], [Chen und Pham, 2000] y [Ross, 2009].

Definición 17 *Variable Lingüística. Es aquella noción o concepto que se calificará de forma difusa, se le aplica el término lingüística porque es definida mediante el lenguaje. Por mencionar algunos ejemplos tenemos la edad, la altura, la temperatura, el error, etcétera.*

Definición 18 *Universo de Discurso (U). Es el rango de valores donde reside la variable lingüística, es decir, todos los posibles valores que pueden tomar los elementos que poseen la propiedad expresada por la variable lingüística. Por ejemplo para una variable lingüística "edad de una persona", serían las edades comprendidas entre* 0 *y* 120 *años.*

Definición 19 *Valor Lingüístico. Son las diferentes clasificaciones que se efectúan sobre la variable lingüística, en otras palabras, son las posibles divisiones que se hacen sobre el universo de discurso. Para ilustrarlo considere como variable lingüística la temperatura, así se podrían definir los valores lingüísticos, "muy alta" , "alta" "normal" , "baja" , "muy baja"* .

Definición 20 *Conjunto Difuso. Es un valor lingüístico junto con una función que denote la pertenencia a dicho conjunto. El valor lingüístico podría considerarse como el nombre del conjunto, y la función de pertenencia se define como aquella asociación entre cada elemento del universo de discurso y el grado con el que pertenece al conjunto difuso.*

Tomando en cuenta la definiciones anteriores, se puede introducir la notación de (4.4), en donde U es el universo de discurso, x_1 es la variable lingüística y $\mu_A(x_1)$ es la función de membresía o pertenencia del conjunto A. Valores grandes en la función de membresía denotan un alto grado de pertenencia a ese conjunto y por el contrario, valores pequeños de membresía representan poca relación con el conjunto. Para ilustrar completamente las definiciones que se acaban de mencionar, la Figura 4.3 muestra su representación gráfica.

$$A = \{(x_1, \mu_A(x_1))\}, \forall\, x_1 \in U \tag{4.4}$$

Por último se define un sistema difuso, el cual se entiende como un grupo de elementos que actúan en conjunto con la finalidad de procesar información, usando como

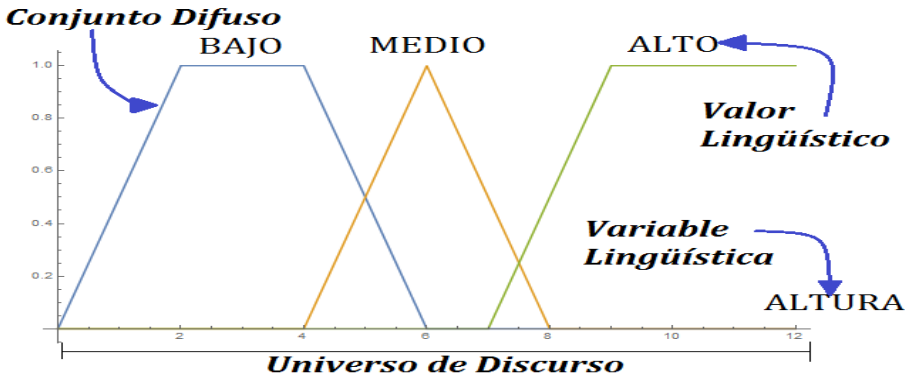


Figura 4.3: Definiciones generales de lógica difusa

base la lógica difusa ([Acosta, 2006]). Muchos libros y artículos científicos (entre ellos [Lee, 1990], [Klir und Yuan, 1995], [Chen und Pham, 2000] y [Orchard, 2004]) distinguen cuatro partes básicas de un sistema difuso. La etapa de fusificación, el motor de inferencia, la base de conocimiento y la etapa de defusificación. A continuación se explica cual es la función de cada una de ellas. En la Figura 4.4 se muestra la estructura de un sistema difuso, donde se aprecia como interactúan estas partes.

- Etapa de Fusificación. En esta fase se convierten los valores provenientes del proceso o señal que se desea tratar, a una representación difusa, generalmente los datos de entrada son números reales y se evalúa la pertenencia de estos a un grupo de conjuntos difusos definidos previamente, usualmente por un experto.

- Base de Conocimiento. Esta parte del sistema contiene una colección de reglas difusas que representan el conocimiento que algún experto posee sobre el comportamiento de las variables provenientes del exterior, derivadas del proceso que se analiza. En algunos casos estas reglas son generadas por un sistema automático. Adicionalmente, la base de conocimiento cuenta con un repertorio de los conjuntos difusos asociados para cada variable de entrada.

- Motor de Inferencia. Es la sección encargada de realizar el razonamiento a partir de las reglas que se tienen, es decir evalúa cuales reglas se cumplen, y en que grado. En general su salida es un agrupación de conjuntos difusos asociados a las variables de

salida.

- Etapa de Defusificación. Aunque esta fase no es necesaria en todos los procesos, comúnmente cuando se trata con magnitudes reales, el resultado que entrega el mecanismo de razonamiento es aún inapropiado para usarse e interpretarse. En este paso se transforman estos términos difusos nuevamente a un valor numérico, llamado salida nítida o salida neta.

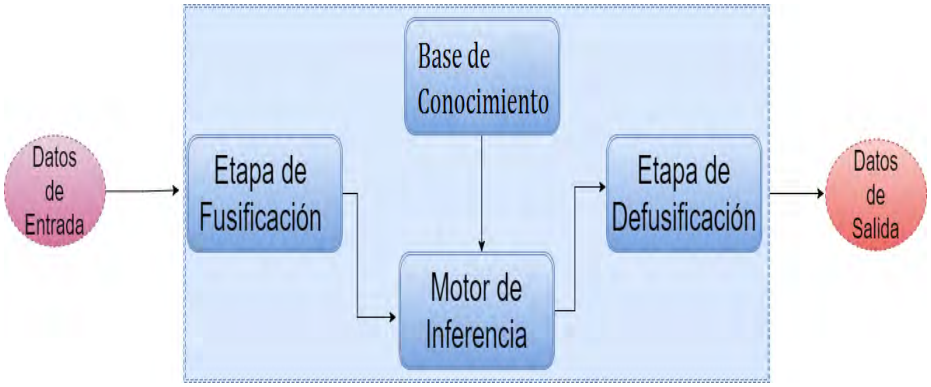


Figura 4.4: Estructura General de un sistema difuso

4.2. Teoría de conjuntos difusos

La Definición 20 describe un conjunto difuso; ahora se explicarán algunos conceptos teóricos que son útiles para denotar sus propiedades (Definiciones 21 a 27). También se mencionan las formas que toman los conjuntos difusos y finalmente las operaciones que pueden realizarse sobre ellos.

Definición 21 *Corte Alfa. Dado un conjunto difuso A para la variable lingüística x_1, con el universo de discurso U_1, se conoce como corte alfa al conjunto de elementos que pertenecen al conjunto A con un grado de membresía mayor o igual que alfa (α), como se observa en* (4.5)*. Similarmente se conoce como corte alfa estricto cuando esa condición se limita a valores mayores que alfa (α).*

$$A_\alpha = \{(x_1, \mu_A(x_1)) \in U_1 \,|\, \mu_A(x_1) \geq \alpha\} \tag{4.5}$$

Definición 22 *Soporte. Es el conjunto de elementos que tienen grado de pertenencia estrictamente mayor que cero, es decir, el corte alfa estricto de nivel cero expresado por* (4.6).

$$Soporte(A) = \{x_1/\mu_A(x_1) \in U_1 \,|\, \mu_A(x_1) > 0\} \tag{4.6}$$

Definición 23 *Núcleo. Es el conjunto de elementos que tiene grado de pertenencia igual a uno, o sea, el corte alfa de nivel uno como se observa en* (4.7).

$$\text{Núcleo(A)} = \{x_1 \in U_1 \,|\, \mu_A(x_1) = 1\} \tag{4.7}$$

Definición 24 *Altura. La altura de un conjunto difuso es el valor más grande de la función de pertenencia asociada.*

Definición 25 *Conjunto difuso normalizado. Se dice que un conjunto difuso está normalizado si su núcleo contiene algún elemento como se observa en* (4.8).

$$\exists\, x_1 \,|\, \mu_A(x_1) = 1 \tag{4.8}$$

Definición 26 *Punto de Cruce. Es el elemento x_1 de U_1 para el cual $\mu_A(x_1) = 0.5$.*

Definición 27 *Conjunto difuso unitario (singleton). Es un conjunto cuyo soporte es un único punto tal que la función de pertenencia es uno, es decir, el soporte coincide con el núcleo y tienen un único punto.*

$$x_1|\mu_A(x_1) = 1 = \text{Soporte(A)} = \text{Núcleo(A)}$$

En la Figura 4.5 se muestra la representación gráfica del núcleo, soporte, punto de cruce y la altura.

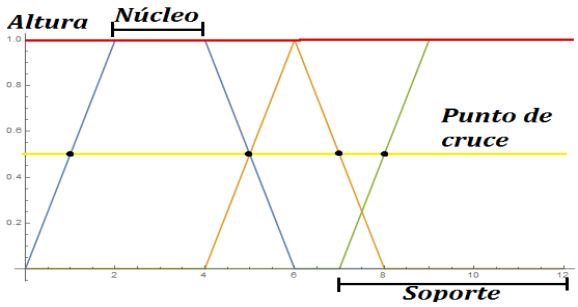


Figura 4.5: Características de los conjuntos difusos

4.2.1. Funciones de membresía

En la Definición 20 se mencionó lo que es una función de membresía, ahora se abordarán las formas posibles que pueden tomar dichas funciones. Si bien, se puede considerar que la forma de una función de membresía puede ser cualquiera, en la práctica existen algunos perfiles predefinidos. Estas formas dependen completamente de la aplicación y del criterio del diseñador del sistema difuso. Las funciones de membresía más conocidas son la triangular (Λ), trapezoidal (Π), monotónamente creciente (Γ), monótonamente decreciente (L), función $\mathbb{S}$, función $\mathbb{Z}$ (se cálcula como $\mathbb{Z} = 1 - \mathbb{S}$), Gaussiana ($\mathbb{G}$) y Campana General ($\mathbb{B}$). Las cuales se muestran en las Definiciones 28 a 35, respectivamente. Para apreciar más claramente éstas funciones se observan sus formas en la Figura 4.6. Algunas de estas funciones de pertenencia se pueden encontrar en [Klir und Yuan, 1995] y [Ross, 2009].

Definición 28 *Una función triangular (Lambda o Λ) se define mediante tres parámetros $a_\Lambda, m_\Lambda, b_\Lambda$ y está dada por* (4.9)*. Siendo como se observa en la Figura 4.6(a).*

$$\Lambda(x_1) = \begin{cases} 0 \text{ para } x_1 \leq a_\Lambda \\ \frac{x_1 - a_\Lambda}{m_\Lambda - a_\Lambda} \text{ para } a_\Lambda < x_1 \leq m_\Lambda \\ \frac{b_\Lambda - x_1}{b_\Lambda - m_\Lambda} \text{ para } m_\Lambda < x_1 \leq b_\Lambda \\ 0 \text{ para } x_1 > b_\Lambda \end{cases} \tag{4.9}$$

Definición 29 *Una función trapezoidal (PI o Π) está determinada por cuatro parámetros $a_\Pi, b_\Pi, c_\Pi, d_\Pi$ y es como se expresa en* (4.10)*. La cual tiene la forma que se muestra en la*

Figura 4.6(b).

$$\Pi(x_1) = \begin{cases} 0 \text{ para } x_1 \leq a_\Pi \\ \frac{x_1 - a_\Pi}{b - a_\Pi} \text{ para } a_\Pi < x_1 \leq b_\Pi \\ 1 \text{ para } b_\Pi < x_1 \leq c_\Pi \\ \frac{d_\Pi - x_1}{d_\Pi - c_\Pi} \text{ para } c_\Pi < x_1 \leq d_\Pi \\ 0 \text{ para } x_1 > d_\Pi \end{cases} \tag{4.10}$$

Definición 30 *La función monótomamente creciente (Gamma o* Γ*) recibe como argumentos los parámetros* a_Γ, m_Γ *y se enuncia matemáticamente como se observa en* (4.11)*. En tanto que su forma es como se muestra en la Figura 4.6(c).*

$$\Gamma(x_1) = \begin{cases} 0 \text{ para } x_1 \leq a_\Gamma \\ \frac{x_1 - a_\Gamma}{m_\Gamma - a_\Gamma} \text{ para } a_\Gamma < x_1 < m_\Gamma \\ 1 \text{ para } x_1 \geq m_\Gamma \end{cases} \tag{4.11}$$

Definición 31 *La función monótomamente decreciente (*$\mathbb{L}$*) está definida mediante dos entradas* $a_\mathbb{L}, m_\mathbb{L}$ *y está determinada por* (4.12)*, mientras que su forma se visualiza en la Figura 4.6(d).*

$$\mathbb{L}(x_1) = (1 - \Gamma(x_1)) = \begin{cases} 1 \text{ para } x_1 \leq m_\mathbb{L} \\ \frac{a_L - x_1}{a_L - m_L} \text{ para } m_\mathbb{L} < x_1 < a_\mathbb{L} \\ 0 \text{ para } x_1 \geq a_\mathbb{L} \end{cases} \tag{4.12}$$

Definición 32 *La función denotada por* $\mathbb{S}$*, se expresa en* (4.13) *y recibe dos parámetros* $a_\mathbb{S}, c_\mathbb{S}$*. Su representación gráfica se observa en la Figura 4.6(e).*

$$\mathbb{S}(x_1) = \begin{cases} 0 \text{ para } x_1 \leq a_\mathbb{S} \\ 2\left(\frac{x_1 - a_\mathbb{S}}{c_\mathbb{S} - a_\mathbb{S}}\right)^2 \text{ para } a_\mathbb{S} \leq x_1 \leq \frac{a_\mathbb{S} + c_\mathbb{S}}{2} \\ 1 - 2\left(\frac{x_1 - c_\mathbb{S}}{c_\mathbb{S} - a_\mathbb{S}}\right)^2 \text{ para } \frac{a_\mathbb{S} + c_\mathbb{S}}{2} \leq x_1 \leq c_\mathbb{S} \\ 1 \text{ para } x_1 \geq c_\mathbb{S} \end{cases} \tag{4.13}$$

Definición 33 *La función conocida como $\mathbb{Z}$ es simplemente la función inversa de la función $\mathbb{S}$ (la cual tiene parámetros $a_{\mathbb{Z}}, c_{\mathbb{Z}}$) y se expresa como en* (4.14)*. Su gráfica se muestra en la Figura 4.6(f).*

$$\mathbb{Z}(x_1) = \begin{cases} 1 \text{ para } x_1 \leq c_{\mathbb{Z}} \\ 1 - 2\left(\frac{x_1 - c_{\mathbb{Z}}}{a_{\mathbb{Z}} - c_{\mathbb{Z}}}\right)^2 \text{ para } c_{\mathbb{Z}} \leq x_1 \leq \frac{c_{\mathbb{Z}} + a_{\mathbb{Z}}}{2} \\ 2\left(\frac{x_1 - c_{\mathbb{Z}}}{a_{\mathbb{Z}} - c_{\mathbb{Z}}}\right)^2 \text{ para } \frac{c_{\mathbb{Z}} + a_{\mathbb{Z}}}{2} \leq x_1 \leq a_{\mathbb{Z}} \\ 0 \text{ para } x_1 \geq a_{\mathbb{Z}} \end{cases} \tag{4.14}$$

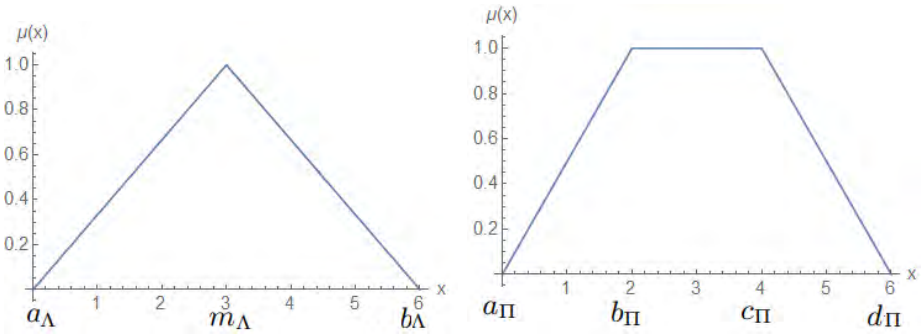
Definición 34 *La función Gaussiana ($\mathbb{G}$) es similar a la usada en teoría de la probabilidad y se caracteriza mediante dos parámetros $E_{\mathbb{G}}[x_1], \sigma_{\mathbb{G}}$, su forma matemática se muestra en* (4.15)*. En la Figura 4.6(g) se observa su forma gráfica. Se debe notar que esta función queda completamente expresada en términos de $E_{\mathbb{G}}[x_1]$ y $\sigma_{\mathbb{G}}$, así que no es una función definida a trozos.*

$$\mathbb{G}(x_1) = e^{-0.5\left(\frac{x_1 - E_{\mathbb{G}}[x_1]}{\sigma_{\mathbb{G}}}\right)^2} \tag{4.15}$$

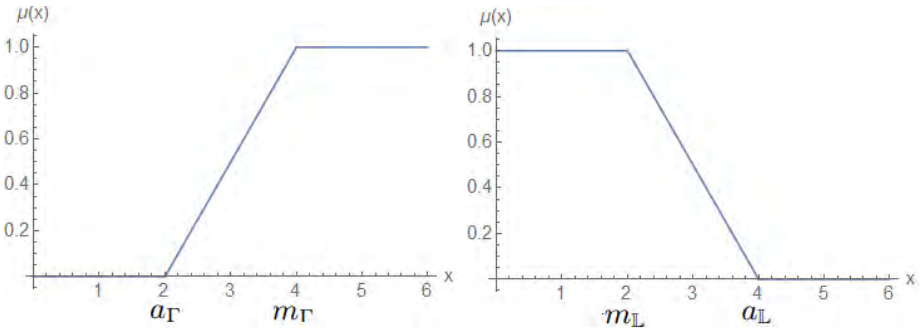
Definición 35 *La función Campana General ($\mathbb{B}$) recibe tres parámetros $a_{\mathbb{B}}, b_{\mathbb{B}}, c_{\mathbb{B}}$ y se expresa mediante* (4.16)*. La cual queda representada gráficamente como se aprecia en la Figura 4.6(h). Al igual que la función Gaussiana queda completamente expresada en términos de sus parámetros y tampoco es una función definida a trozos. Comúnmente $b_{\mathbb{B}} > 0$ en caso contrario la campana se invierte.*

$$\mathbb{B}(x_1) = \frac{1}{1 + \left|\frac{x_1 - c_{\mathbb{B}}}{a_{\mathbb{B}}}\right|^{2b_{\mathbb{B}}}} \tag{4.16}$$

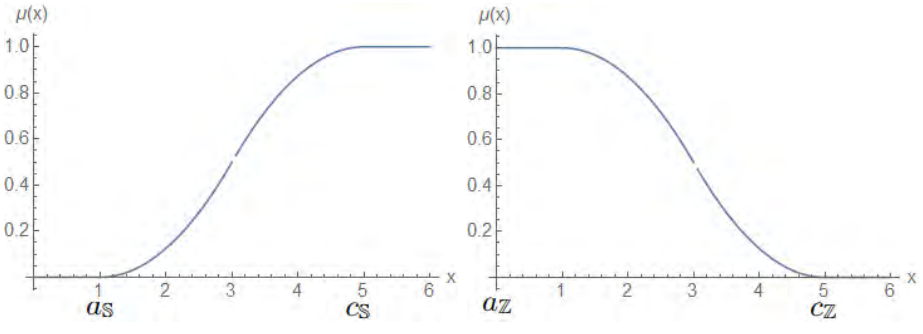
Se puede apreciar que estas funciones se asocian a conjuntos difusos normalizados (su altura es uno), aunque se pueden usar funciones que no lo sean, es más común trabajar con conjuntos normalizados.



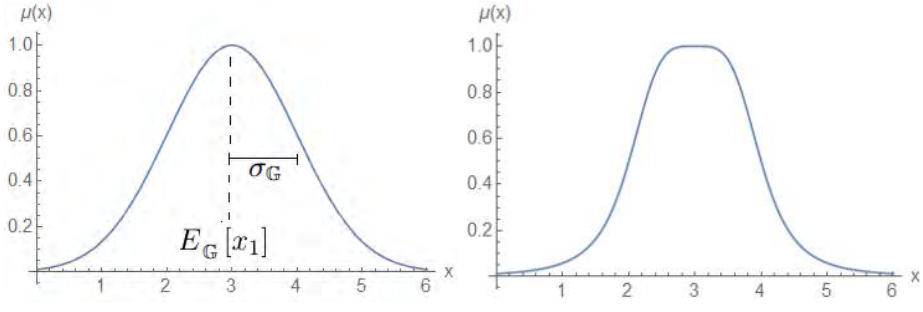
(a) ***Triangular*** $\Lambda(x_1)(a_\Lambda, m_\Lambda, b_\Lambda)$ **(b)** ***Trapezoidal*** $\Pi(x_1)(a_\Pi, b_\Pi, c_\Pi, d_\Pi)$



(c) ***Gamma*** $\Gamma(x_1)(a_\Gamma, m_\Gamma)$ **(d)** $\mathbb{L}(x_1)(m_\mathbb{L}, a_\mathbb{L})$



(e) $\mathbb{S}(x_1)(a_\mathbb{S}, c_\mathbb{S})$ **(f)** $\mathbb{Z}(x_1)(a_\mathbb{Z}, c_\mathbb{Z})$



(g) ***Gaussiana*** $\mathbb{G}(x_1)(E_\mathbb{G}[x_1], \sigma_\mathbb{G})$ **(h)** ***Campana General*** $\mathbb{B}(x_1)(a_\mathbb{B}, b_\mathbb{B}, c_\mathbb{B})$

Figura 4.6: Funciones de Membresía más comunes

4.2.2. Operaciones en Conjuntos Difusos

De forma similar que en la lógica clásica, en la difusa también existen operaciones que se pueden definir sobre los conjuntos. La notación es algo parecida a la de (4.1), (4.2) y (4.3). La diferencia radica en la interpretación que se les da, ya que estos conjuntos necesariamente tienen ligada una función de membresía, así que es sobre estas donde se aplican las operaciones. Comúnmente, se usa de manera indistinta la notación de las operaciones sobre los conjuntos o sobre las funciones de membresía. Basandose en [Ross, 2009], a continuación se exponen estas operaciones.

La unión en los conjuntos difusos se puede realizar por medio de funciones conocidas como t-conormas. Considérense dos conjuntos difusos A_1 y A_2, para la variable x_1 en un universo de discurso U_1. La unión en conjuntos difusos se expresa como se aprecia en (4.17). Las t-conormas deben cumplir una serie de condiciones, así que no cualquier función sirve para este propósito (para mayor información se puede consultar [Schweizer und Sklar, 2011]). Las tres t-conormas más populares son la función máximo, la función suma-producto y la suma drástica. La unión de conjuntos difusos usando estos tres esquemas se denota como se observa en (4.18), (4.19) y (4.20), respectivamente.

$$A_1 \cup A_2 \equiv \mu_{A_1 \cup A_2}(x_1) \equiv \mu_{A_1}(x_1) \cup \mu_{A_2}(x_1) \tag{4.17}$$

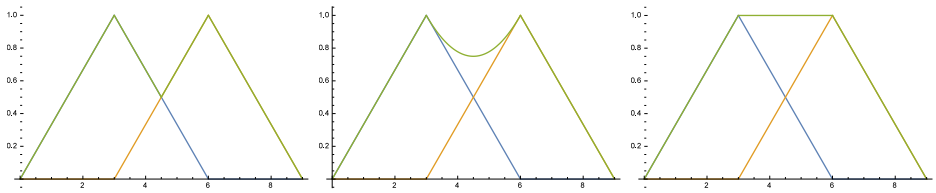
$$A_1 \cup A_2 = Max(\mu_{A_1}(x_1), \mu_{A_2}(x_1)) \tag{4.18}$$

$$A_1 \cup A_2 = \mu_{A_1}(x_1) + \mu_{A_2}(x_1) - (\mu_{A_1}(x_1) * \mu_{A_2}(x_1)) \tag{4.19}$$

$$A_1 \cup A_2 = \begin{cases} \mu_{A_1}(x_1) \text{ si } \mu_{A_2}(x_1) = 0 \\ \mu_{A_2}(x_1) \text{ si } \mu_{A_1}(x_1) = 0 \\ 1 \text{ otro caso.} \end{cases} \tag{4.20}$$

Para una mayor comprensión de estas funciones (las que fungen como el operador unión en los conjuntos difusos), se muestra en la Figura 4.7 su representación gráfica. En esta figura se usan como ejemplos funciones de membresía triangulares, la del conjunto A_1 está representado por la señal azul ($\mu_{A_1}(x_1)$), la del conjunto A_2 por la naranja ($\mu_{A_2}(x_1)$) y el resultado de la unión por la verde ($A_1 \cup A_2$). Donde la Figura 4.7(a), es la unión mediante la t-conorma del máximo, la Figura 4.7(b) es por medio de la suma-producto y la Figura

4.7(c), representa la unión bajo la t-conorma de la suma drástica.



(a) *Máximo* **(b) *Suma-Producto*** **(c) *Suma Drástica***

Figura 4.7: t-conormas para el operador unión en conjuntos difusos

Ahora, el operador de intersección de (4.21), considerando nuevamente los dos conjuntos difusos A_1 y A_2 se define por funciones llamadas t-normas. Estas también deben satisfacer ciertas restricciones (en [Schweizer und Sklar, 2011] se da una explicación detallada de en que consisten) y las más comunes son la función mínimo expresada en (4.22) el producto que se muestra en (4.23)) y el producto drástico que se aprecia en (4.24).

$$A_1 \cap A_2 \equiv \mu_{A_1 \cap A_2}(x_1) \equiv \mu_{A_1}(x_1) \cap \mu_{A_2}(x_1) \tag{4.21}$$

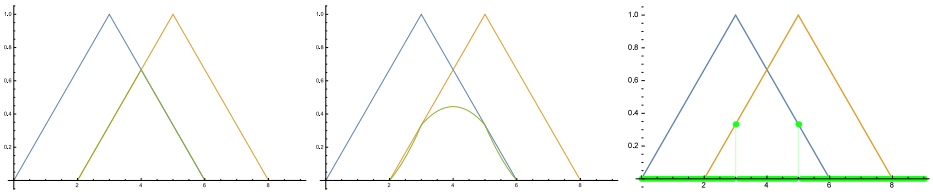
$$A_1 \cap A_2 = Min(\mu_{A_1}(x_1), \mu_{A_2}(x_1)) \tag{4.22}$$

$$A_1 \cap A_2 = \mu_{A_1}(x_1) * \mu_{A_2}(x_1) \tag{4.23}$$

$$A_1 \cap A_2 = \begin{cases} \mu_{A_1}(x_1) \text{ si } \mu_{A_2}(x_1) = 1 \\ \mu_{A_2}(x_1) \text{ si } \mu_{A_1}(x_1) = 1 \\ 0 \text{ otro caso.} \end{cases} \tag{4.24}$$

En algunas ocasiones la intersección también se denota como $I(\mu_{A_1}(x_1), \mu_{A_2}(x_1))$. En la Figura 4.8 se pueden visualizar las t-normas utilizadas como intersección. Donde se tiene que la Figura 4.8(a) es la función mínimo, la Figura 4.8(b) es la función producto y por último la Figura 4.8(c) representa la función del producto drástico. En los tres casos, la forma azul representa a la función de membresía $\mu_{A_1}(x_1)$ y la naranja a $\mu_{A_2}(x_1)$, en tanto que la verde es el resultado de calcular su intersección $(A_1 \cap A_2)$.

Por último, se mencionan las funciones que normalmente son usadas para denotar la operación de complemento. Considérese un conjunto difuso A_1, entonces la operación de complemento se puede expresar como se observa en (4.25). Las funciones que se usan de



(a) ***Mínimo*** **(b)** ***Producto*** **(c)** ***Producto Drástico***

Figura 4.8: t-normas para el operador intersección en conjuntos difusos

complemento, deben hacer un mapeo en el rango $[0, 1]$. Este mapeo se espera que invierta los límites, es decir, calcular el complemento de 0 debe regresar 1 y viceversa. También se desea que la función sea estrictamente decreciente y que al aplicarse dos veces consecutivas sobre un conjunto, el resultado sea ese mismo conjunto. Las tres funciones más utilizadas que cumplen con lo anterior son el complemento estándar expresado en (4.26)), el complemento de Yager que se observa en 4.27 y el de Sugeno ver (4.28).

$$\neg A_1 \equiv C(\mu_{A_1}(x_1)) \equiv \mu_{\bar{A}_1}(x_1) \tag{4.25}$$

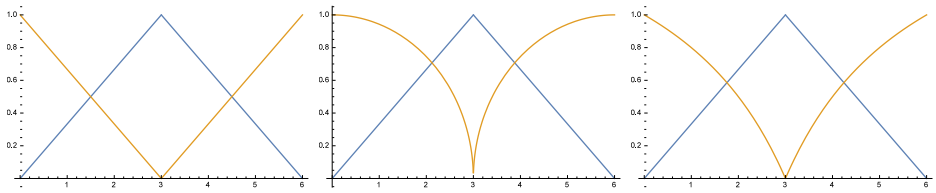
$$\neg A_1 = 1 - \mu_{A_1}(x_1) \tag{4.26}$$

$$\neg A_1 = (1 - \mu_{A_1}(x_1)^{w_y})^{1/w_y} \;\; w_y \in [0, \infty) \tag{4.27}$$

$$\neg A_1 = \frac{1 - \mu_{A_1}(x_1)}{1 - \lambda_c \mu_{A_1}(x_1)}, \; \lambda_c \in [0, 1]. \tag{4.28}$$

Con la finalidad de apreciar su comportamiento en la Figura 4.9 se muestran los resultados de aplicar éstas funciones. El complemento estándar se observa en la Figura 4.9(a), el de Yager en la Figura 4.9(b) y el de Sugeno en la Figura 4.9(c). Nuevamente se usan funciones de pertenencia triangulares, donde la azul representa a μ_{A_1}, mientras que la naranja es el resultado de la operación de complemento $\mu_{\bar{A}_1}$.

Lo anterior resume la teoría de conjuntos difusos, la cual es usada en la fase de fusificación que se mencionó en la introducción de este capítulo (como se visualizaba en la Figura 4.4). A continuación se explica la forma que toman los datos dentro del proceso de fusificación.



(a) *Estandar* **(b) *de Yager*** **(c) *de Sugeno***

Figura 4.9: Funciones usuales para calcular el complemento de un conjunto difuso

4.2.3. Proceso de Fusificación

Considerando que existen NV variables de entrada $(x_1, x_2, \ldots, x_{NV})$ y una variable de salida (y_1). Para cada x_i existirán NC_i conjuntos difusos en los que se dividirá el universo de discurso U_i. Así que, en la base de conocimiento habrá almacenado un arreglo de todos los conjuntos difusos, los cuales están asociados a cada una de las variables de entrada. En (4.29) se muestra la forma general que tiene el arreglo $\mathcal{A}_i$ asociado a la variable x_i.

$$\mathcal{A}_i = \{A_{i,1}, A_{i,2}, \ldots, A_{i,NC_i}\} \tag{4.29}$$

donde $i = 1, 2, \ldots, NV$ y cada término $A_{i,j}$ $(j = 1, 2, \ldots, NC_i)$ es un conjunto dentro del universo de discurso U_i. Adicionalmente cada conjunto $A_{i,j}$ tiene asociada una función de membresía $\mu_{A_{i,j}}(x_i)$. Note que no necesariamente existen el mismo número de conjuntos para cada variable de entrada. En la práctica el número de conjuntos para cada variable depende del problema en cuestión, aunque generalmente se considera que el arreglo de (5.2) tiene la misma cantidad de elementos para cada variable x_i. Por lo tanto se considera que para cada x_i habrá NC conjuntos difusos.

De la misma manera existirá un vector de conjuntos difusos $(\mathcal{B})$ para la única variable de salida y_1, el cual también tendrá NC conjuntos distribuidos en el universo de discurso V_1. Este vector tiene la forma que se observa en (4.30).

$$\mathcal{B} = \{B_1, B_2, \ldots, B_{NC}\}\,. \tag{4.30}$$

En la etapa de fusificación por cada variable de entrada (x_i), se obtendrá una lista de los valores que tomaron las funciones de membresía (asociadas a cada $A_j \in \mathcal{A}_i$), para el

valor específico (c_i) que tomó cada x_i. Sin embargo, no todos los conjuntos se activan ante la entrada $x_i = c_i$. Esto implica que se toma un subconjunto de $\mathcal{A}_i$ para cada variable de entrada x_i. El número de conjuntos que se activaron para cada variable x_i se denota como NC_{activ} y se calcula por medio de (4.31).

$$NC_{activ}(x) = \mid \left\{ \forall A_j \in \mathcal{A}_i \mid \mu_{A_{i,j}} \neq 0 \right\} \mid \tag{4.31}$$

donde $i = 1, 2, \ldots, NV$, $j = 1, 2, \ldots, NC$ y $\mid\mid$ expresa la cantidad de elementos contenidos en el arreglo (tamaño). Se debe cumplir que $NC_{activ} \leq NC$. La expresión indica que el número de conjuntos activados son los conjuntos totales menos los que tienen pertenencia igual a cero.

Así este proceso tiene la forma que se observa en (4.32). En la que $\boldsymbol{\mu}_{\mathcal{A}_i}(x_i = c_i)$ representa las membresías asociadas a los conjuntos $\mathcal{A}_i$, considerando que $\mathcal{A}_i$ contiene A_j conjuntos pero ahora $j = 1, 2, \ldots, NC_{activ}(x_i)$.

$$\begin{aligned} Fuzz(x_i = c_i) &= \boldsymbol{\mu}_{\mathcal{A}_i}(x_i = c_i) \\ &= \{\mu_{A_1}(x_i = c_i), \mu_{A_2}(x_i = c_i), \ldots, \mu_{A_{NC}}(x_i = c_i)\} \end{aligned} \tag{4.32}$$

donde el término $\boldsymbol{\mu}_{\mathcal{A}_i}(x_i = c_i)$ simboliza el arreglo de los NC valores que tomó la función de membresía para la i-ésima variable de entrada. Tomando en cuenta que $i = 1, 2, \ldots, NV$ y $Fuzz()$ expresa la operación de fusificar las variables de entrada.

En muchas ocasiones, solo se toma el conjunto (dentro de los que se activaron) que tiene mayor pertenencia. Así por cada entrada se obtiene únicamente un conjunto difuso y su membresía. Esta pertenencia es el máximo valor contenido en $\boldsymbol{\mu}_i$. Por consiguiente, la fusificación de las variables de entrada mostrada en (4.32) puede tomar la forma de (4.33).

$$\begin{aligned} Fuzz(x_i = c_i) &= \boldsymbol{\mu}_{Max(\mathcal{A}_i)}(x_i = c_i) \\ &= Max\left(\{\mu_{A_1}(x_i = c_i), \mu_{A_2}(x_i = c_i), \ldots, \mu_{A_{NC}}(x_i = c_i)\}\right) \end{aligned} \tag{4.33}$$

donde $\boldsymbol{\mu}_{Max(\mathcal{A}_i)}(x_i = c_i)$ representa el valor máximo contenido en $\boldsymbol{\mu}_i(x_i = c_i)$. Tomar exclusivamente el mayor valor por cada variable se hace con la finalidad de llevar a cabo

el menor número de operaciones. La cantidad de conjuntos que se hayan activado para la variable de entrada después se compara con las reglas difusas y puede volverse un proceso tardado entre más conjuntos se hayan activado en la fase de fusificación. Para más información ver [Lee, 1990], [Klir und Yuan, 1995], [Zadeh, 1996], [Chen und Pham, 2000], [Orchard, 2004], [Acosta, 2006] y [Ross, 2009].

4.3. Razonamiento difuso

Esta sección explica en qué consiste la inferencia difusa (razonamiento difuso), mencionando las proposiciones y las reglas difusas. El razonamiento difuso presenta muchas similitudes con la inferencia en la lógica clásica, aunque se diferencian principalmente en la combinación de los conjuntos. Asimismo se diferencian en la forma de interpretar las implicaciones, ya que, tomar directamente las formas de implicación que presenta la lógica convencional tiene poco interés práctico, quedando más como enfoques teóricos.

4.3.1. Proposiciones difusas

Tanto en la lógica tradicional como en la difusa, se conoce como proposición a un enunciado que se representa mediante variables lógicas. Existen dos tipos de proposiciones, las primeras, llamadas simples, asignan un valor a una variable difusa, así que estas necesariamente tienen asociado un conjunto difuso. El segundo tipo de proposición, conocida como compuesta, se obtiene por la agrupación de dos o más proposiciones simples. Esta concentración de proposiciones se hace utilizando las funciones de unión, intersección y complemento. Sin embargo, cuando se trata de proposiciones, los operadores $\cup, \cap, \neg$ suelen reemplazarse por $\vee, \wedge$ y $\neg$. Además se debe tomar en cuenta que las proposiciones enuncian que la variable lingüística tomó un valor lingüístico determinado por un conjunto difuso.

Considérense dos variables lingüísticas x_1 y x_2 que corresponden a dos proposiciones p_1 y p_2, asociadas a los conjuntos difusos A_1 y A_2, con universos de discurso U_1 y U_2, respectivamente. Aplicar las operaciones a las proposiciones conlleva combinar de la misma manera a las funciones de membresía de cada conjunto. Así que las operaciones $\vee, \wedge$ y $\neg$ en las proposiciones p_1 y p_2 realmente se hacen con los conjuntos A_1 y A_2, o bien, sobre sus

funciones de membresía $(\mu_{A_1}(x_1), \mu_{A_2}(x_2))$. Al igual que cuando se opera sobre conjuntos, estas operaciones se pueden aplicar sobre las funciones de pertenencia en su totalidad o sobre un punto específico de estas (en valores definidos para x_1, x_2). De esta manera la unión queda representada en (4.34), la intersección es como se observa en (4.35) y el complemento será como se aprecia en (4.36).

$$p_1 \vee p_2 \equiv (x_1 \textbf{ es } A_1) \vee (x_2 \textbf{ es } A_2) \tag{4.34}$$

$$p_1 \wedge p_2 \equiv (x_1 \textbf{ es } A_1) \wedge (x_2 \textbf{ es } A_2) \tag{4.35}$$

$$\neg p_1 \equiv \neg(x_1 \textbf{ es } A_1) \tag{4.36}$$

donde los operadores $\wedge, \vee$ y $\neg$ representan las conectivas del lenguaje **y**, **o**, y **no**, respectivamente. Por otro lado, la expresión x_i **es** A_j quiere decir que la variable x_i activó el conjunto A_j, o sea que el valor de la función de membresía asociada al conjunto A_j es mayor a cero $(\mu_{A_j}(x_i) > 0)$.

4.3.2. Reglas difusas e implicaciones

Una regla difusa es una relación difusa que se tiene entre los antecedentes y el consecuente; su fuerza representa el valor de la relación de pertencia. También puede ser visto como que tan fuertemente un experto cree en la regla. Las reglas difusas deben capturar relaciones de causa y efecto y normalmente son detectadas por un experto. Las relaciones causa efecto en la lógica difusa se pueden representan mediante implicaciones. Tomando dos proposiciones simples p_1 y q_1, se dice que p_1 implica q_1, si p_1 es un hecho considerado como antecedente y q_1 se considera como un hecho consecuente. La relación de implicación se denota mediante $p_1 \rightarrow q_1$. Usualmente, se busca que las implicaciones difusas representen relaciones en donde el consecuente es un tipo de respuesta ante lo presentado en el antecedente. Es decir, una implicación representa una relación causal.

Las reglas difusas toman una forma «**Si** *Antecedente*$_1$ **y** *Antecedente*$_2$ **y**... **y** *Antecedente*$_{NV}$ **Entonces** *Consecuente* », lo cual se expresa en términos matemáticos en (4.37). Normalmente se les denomina reglas **Si-Entonces** (en inglés, **If-Then**). En la cual $p_1, p_2, \ldots, p_{NV}$ son los antecedentes, $x_1, \ldots, x_{NV}$ representan las variables de entrada

(con los conjuntos difusos asociados $A_1, \ldots, A_{NV}$ en los universos de discurso $U_1, \ldots, U_{NV}$). Considerando además q_1 como el consecuente, para la variable de salida y_1 con su conjunto asociado B_1, en un universo de discurso V_1. Donde NV representa el número de variables de entrada y está directamente asociado al número de antecedentes. Aunque se puede pensar en múltiples consecuentes, no se incluirá esta situación ya que esto no es común en la práctica . Lo anterior de acuerdo con la mayoría de las fuentes bibliográficas (por ejemplo [Zadeh, 1996] y [Acosta, 2006]).

$$\begin{aligned} R &= \textbf{Si}\ (x_1\ \textbf{es}\ A_1)\ \textbf{y}\ (x_2\ \textbf{es}\ A_2)\ \textbf{y} \ldots \textbf{y}\ (x_{NV}\ \textbf{es}\ A_{NV})\ \textbf{Entonces}\ (y_1\ \textbf{es}\ B_1) \quad (4.37) \\ R &= p_1 \wedge p_2 \wedge \ldots \wedge p_{NV} \rightarrow q_1 \\ &\equiv (x_1 \triangleright A_1) \wedge (x_2 \triangleright A_2) \wedge \ldots \wedge (x_{NV} \triangleright A_{NV}) \rightarrow (y_1 \triangleright B_1). \end{aligned}$$

donde p_1 representa a $(x_1\ \textbf{es}\ A_1)$, etc. En tanto q_1 representa a $(y_1\ \textbf{es}\ B_1)$ y el símbolo $\triangleright$ representa la expresión **es**.

La implicación de dos proposiciones p_1, q_1 se puede expresar como en 4.38 y 4.39. No obstante, estas expresiones no corresponden con vinculaciones causa-efecto, así que no modelan reglas **Si-Entonces**.

$$p_1 \rightarrow q \equiv (\neg p_1) \vee q \tag{4.38}$$

$$p_1 \rightarrow q \equiv \neg\,(p_1 \wedge (\neg q)) \tag{4.39}$$

El significado más usado para la implicación se conoce como de Mamdani, ver [Mamdani, 1976]. En este enfoque la implicación se interpreta como cierta en el grado que el antecedente y el consecuente lo sean. La implicación de Mamdani toma como punto de partida la expresión presentada en (4.40), que considera una implicación como una intersección de los antecedentes y el consecuente. Lo anterior para un antecedente p_1, un consecuente q_1 que están ligados a los conjuntos difusos A_1 y B_1 con universos de discurso U_1 y V_1.

$$p_1 \wedge p_2 \wedge \ldots \wedge p_{NV} \rightarrow q_1 \equiv p_1 \wedge p_2 \wedge \ldots \wedge p_{NV} \wedge q_1. \tag{4.40}$$

Mamdani considera la implicación como una intersección ya que busca modelar relaciones de causa y efecto. Por ejemplo, supongase que se está interesado en el control de riego de un cultivo, se busca que las relaciones traten de modelar la toma de decisiones que se debe llevar acabo. Una regla en la que se puede pensar es **Si** " la «humedad» es 'baja' " **Entonces** " «la apertura de la válvula» debe ser 'grande' " . En este caso las reglas generadas llevan un tipo de control intrínseco. En otras ocasiones, simplemente se busca representar que algunos hechos conllevan a otros. Por ejemplo, si se trabaja en la generación de energía eléctrica por medio del viento, podrían surgir reglas difusas como **Si** " la «velocidad del viento» es 'baja' " **Entonces** " «la cantidad de energía generada» es 'baja' " . En cualquiera de los dos ejemplos se observa que en las relaciones **Si-Entonces** los antecedentes indican el grado de veracidad de la regla. Normalmente la fuerza con la que se activa el consecente se calcula como la intersección entre el grado de pertenencia de los antecedentes y la fortaleza con la que se activa la regla.

Tomando en cuenta la implicación tipo Mamdani de (4.40) y la estructura de una regla difusa con multiples antecedentes, ver (4.37), se puede obtener (4.41). En la cual las reglas quedan representadas como la intersección de todos los antecedentes y el consecuente. Estas expresiones son la base para realizar inferencias difusas, sin embargo se deben hacer ciertas consideraciones adicionales.

$$R = p_1 \wedge p_2 \wedge \ldots \wedge p_{NV} \wedge q_1 \tag{4.41}$$

$$\equiv (x_1 \triangleright A_1) \wedge (x_2 \triangleright A_2) \wedge \ldots \wedge (x_{NV} \triangleright A_{NV}) \wedge (y_1 \triangleright B_1) \tag{4.42}$$

Otro enfoque usando para la creación de reglas es conocido como de Takagi-Sugeno ([Acosta, 2006]). En el cual se considera el consecuente, como el resultado de mapear los valores que tomaron las variables de entrada (por medio de una función que genere un valor real). En la práctica esta forma de asumir las reglas es usada cuando se trabaja en sistemas de control. En especial cuando se desea trabajar en diferentes regiones de operación, pero para cada una de estas regiones existe un modelo matemático conocido. Así que las reglas sirven para combinar que tanto se trabaja sobre una región u otra y la respuesta

del controlador es una combinación de algunas funciones predefinidas. En esta tesis no se utiliza este enfoque, limitándose a usar el de Mamdani.

4.3.3. Inferencia difusa

En el proceso de inferencia se deben determinar los grados de activación, así como las formas que tomarán los conjuntos difusos de los consecuentes (que dependen directamente de estos grados de activación). Como se mostró en la Figura 4.4, la inferencia se realiza después de un proceso de fusificación, a partir de una base de conocimiento. Esta base de conocimiento alberga un conjunto de reglas, así como un grupo de conjuntos difusos para las variables de entrada y otro para las variables de salida, ver (4.29) y (4.30).

En la Figura 4.10 se muestran los pasos que se siguen en la fase de inferencia, donde primero se hace una verificación de las reglas activadas, se calculan las fuerzas de activación asociadas a cada regla, se combinan las formas de los conjuntos consecuentes con las fuerzas de activación y finalmente se realiza una composición entre los diferentes conjuntos difusos (obtenidas de las reglas que se activaron) para la variable de salida. Al final de este proceso se obtiene un conjunto difuso de salida, que pasará posteriormente a una etapa de defusificación.

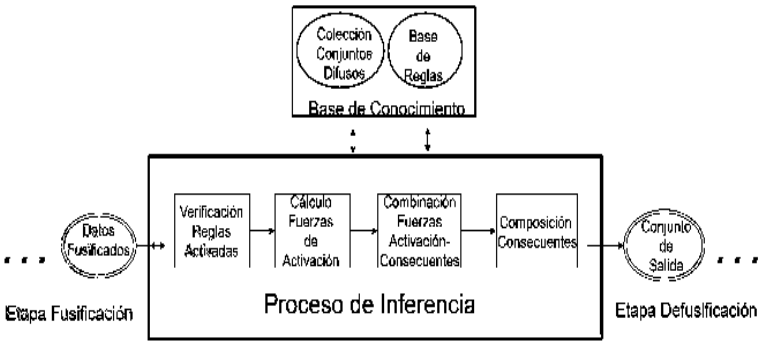


Figura 4.10: Fases del Proceso de Inferencia Difusa

Activación de las reglas

Los datos fusificados se comparan con las reglas que se tienen predefinidas, para ver cuales se pueden activar. Ya sea que los datos contengan las pertenencias a cada uno de los conjuntos que se activaron ver (4.32), o solo las pertenencias de los conjuntos con mayor grado de activación como se mostraba en (4.33).

Para verificar que reglas se activan considérese una base de reglas $\mathcal{R}$ con NR reglas contenidas, ver (4.43). Donde cada regla R_k tiene un antecedente (p_i) por cada variable de entrada x_i. Los antecedentes de cada regla tienen asociados conjuntos difusos que están fijos. O sea, que cada regla R_k tiene un conjunto predefinido por cada antecedente, que corresponde al estado o relación que modela esta regla.

$$\mathcal{R} = \{R_1, R_2, \ldots, R_{NR}\} \tag{4.43}$$

De esta manera que una regla se active implica que una situación predeterminada se cumplió. Para determinar esto se hace una comparación entre los datos de entrada mostradas en (4.32) y (4.33) con la base de reglas. Esta comparación requiere evaluar si los conjuntos asociados a los datos fusificados coinciden con los conjuntos de los antecedentes de cada regla. Es necesario mencionar que el conjunto asociado a cada antecedente debe ser alguno de los NC_{activ} conjuntos activados (en la fusificación) para la variable x_i. De esta manera, las reglas activadas ($\mathcal{R}_{activ}$) son un subconjunto de la base de reglas ($\mathcal{R}$). Estas reglas activadas $\{R_1, R_2, \ldots, R_{NR_{activ}}\}$ son las posibles combinaciones de todos los conjuntos difusos asociados a cada variable de entrada. Cada regla R_k tiene la forma de (4.37) considerando un antecedente por cada variable de entrada x_i.

Fuerzas de activación

Una vez que se tienen las reglas que se activaron, se procede a calcular la fuerza de activación de cada regla. Esta fuerza de activación expresa qué tanto se cumple la regla considerando los datos de entrada actuales. Es una forma de ponderar en que grado se activa (fuerza de activación) cada regla, en base a los datos recibidos. Para calcular el arreglo de las fuerzas activadas ($\mathcal{F}$) se debe hacer la intersección de las membresías asociadas a cada

conjunto difuso por cada antecedente (para cada una de las reglas). Así las fuerzas de activación tienen la forma que se aprecia en (4.44).

$$\mathcal{F} = \{F_1, F_2, \ldots, F_{NR_{activ}}\} \tag{4.44}$$

donde $F_k = \mu_{A_{k,1}}(x_1 = c_1) \wedge \mu_{A_{k,2}}(x_2 = c_2) \wedge \ldots \wedge \mu_{A_{k,NV}}(x_{NV} = c_{NV})$.

Estas fuerzas de activación normalmente están relacionadas con los conjuntos consecuentes de las reglas que se activan; se usan para determinar la forma que tomarán los conjuntos contenidos en los consecuentes. Sin embargo, es posible que en la lista de reglas activadas existan algunas que tengan el mismo consecuente. Por lo que se debe encontrar una colección de conjuntos reducida, en la cual se eliminen los conjuntos repetidos en los consecuentes. Así que el número de conjuntos reducidos (NC_{red}) se calcula como se muestra en (4.45).

$$\mathcal{B}_{red} = \{B_1, B_2, \ldots, B_{NC_{red}}\} \tag{4.45}$$

donde el número de conjuntos reducidos se calcula como todos los conjuntos asociados a un consecuente (de una regla activada) tales que sean diferentes al resto de los conjuntos de los consecuentes ver (4.46)).

$$NC_{red} = \mid \{\forall B_{k_1} \in q_{k_1} \mid B_{k_1} \neq B_{k_2}\} \mid \tag{4.46}$$

para $k_1 = 1, 2, \ldots, NR_{activ}$ y $k_2 \neq k_1 \in [1, N\ R_{activ}]$.

También se debe calcular la combinación de las fuerzas de activación de las reglas que tienen el mismo consecuente. El resultado de este procedimiento genera NC_{red} términos que se conocen como fuerzas de activación resultantes, denotadas por $\mathcal{F}_{res}$. La forma de este arreglo es como se observa en (4.47).

$$\mathcal{F}_{red} = \{F_1, F_2, \ldots, F_{NC_{red}}\} \tag{4.47}$$

donde cada fuerza de activación resultante (F_l para $l = 1, 2, \ldots, NC_{red}$) se obtiene de hacer la intersección de todas las fuerzas de activación asociadas a cada conjunto reducido (B_l).

Combinación consecuente-fuerza de activación

Una vez que se tienen las fuerzas de activación resultantes se hace una combinación de los conjuntos reducidos de los consecuentes presentados en (4.45)) con su fuerza de activación resultante. Esta combinación se hace obteniendo la intersección de las funciones de membresía asociadas a los conjuntos reducidos ($\mathcal{B}_{red}$) con las fuerzas de activación resultantes ($\mathcal{F}_{res}$). Es necesario mencionar que esta intersección es entre la función de membresía de cada B_l y el valor escalar F_l. Esto genera un nuevo grupo de conjuntos resultantes ($\mathcal{B}_{res}$) que tiene asociado un conjunto de funciones de membresía $\boldsymbol{\mu}_{\mathcal{B}_{res}}(y_1)$, el cual se obtiene por medio de (4.48).

$$\boldsymbol{\mu}_{\mathcal{B}_{res}}(y_1) = \left\{(\mu_{B_1}(y_1) \wedge F_1), (\mu_{B_2}(y_2) \wedge F_2), \ldots, (\mu_{B_{NC_{red}}}(y_1) \wedge F_{NC_{red}})\right\} \tag{4.48}$$

Composición de consecuentes

Finalmente se hace una composición de los consecuentes que se observan en (4.48), con la finalidad de obtener una única función de pertenencia para la variable de salida y_1. Se deben combinar todos los conjuntos difusos que se activaron y ponderaron en las fases anteriores. Para lograr lo anterior se utiliza la unión de los conjuntos, de manera que el conjunto final B_{comp} tendrá asociada una función de pertenencia que se calcula en base a (4.49).

$$\begin{aligned} \mu_{B_{comp}}(y_1) &= U\left(\boldsymbol{\mu}_{\mathcal{B}_{res}}(y_1)\right) \\ &= (\mu_{B_1}(y_1) \wedge F_1) \vee (\mu_{B_2}(y_2) \wedge F_2) \vee \ \ldots \ \vee (\mu_{B_{NC_{red}}}(y_1) \wedge F_{NC_{red}}) \end{aligned} \tag{4.49}$$

En el proceso de inferencia todas las combinaciones se realizan por medio de los operadores intersección y unión.

4.4. Métodos de defusificación

La etapa de defusificación convierte el resultado que se obtiene de la inferencia en un valor nítido. Es decir, se cambia la representación de la variable de salida de un

contexto difuso a una representación numérica. La defusificación se puede ver como un proceso matemático que convierte un conjunto difuso en un número real. Existen técnicas que se conocen como métodos de defusificación. Generalmente se basan en un mapeo de una figura geométrica a un valor puntual (el conjunto de salida obtenido mediante la inferencia es básicamente una figura geométrica). A continuación se explican los métodos más conocidos que cumplen con esta tarea, destacando el método del centroide o centro de gravedad (en inglés Center of Gravity) (COG), por ser el que tiene un mayor uso.

4.4.1. Métodos basados en Máximos

La función de membresía resultante del proceso de inferencia ($\mu_{B_{comp}}(y_1)$) se usa para determinar el valor nítido de salida. Se asume esta función de membresía se encuentra en una parte o la totalidad del universo de discurso V_1, para la variable de salida y_1. Para determinar un valor puntual a partir de la forma que tiene la función de membresía, se puede usar la función máximo. En este sentido existen tres vertientes. El menor de los máximos (en inglés Smallest of Maximum) (SOM) ([Jacques u. a., 2002]), la media del máximo (en inglés Middle of Maximum) (MOM) ([Lee, 1990] y [Orchard, 2004]) y el mayor de los máximos (en inglés Largest of Maximum) (LOM) ([Toolbox, 1995–2015]).

Método del menor de los máximos

El método SOM toma como resultado el valor de la variable de salida y_1 más pequeño pero que sea un máximo de la forma que tiene la función de membresía del conjunto difuso de salida. En otras palabras, debe ser el máximo que regrese el menor valor de y_1. Esto se expresa según (4.50).

$$y^* = Min\left(\forall\, y_1 \in V_1 \,|\, \mu_{B_{comp}}(y_1) = Max\left(\mu_{B_{comp}}(y_1)\right)\right) \tag{4.50}$$

Método del mayor de los máximos

El método LOM toma como valor de salida el máximo que regrese el mayor valor posible para y_1 (se encuentre más a la derecha del plano cartesiano donde se representa

la función de membresía). Es decir, se toma el valor más grande de y_1 para el cual la pertenencia es máxima. Este se puede calcular según(4.51).

$$y^* = Max\left(\forall\, y_1 \in V_1 \,|\, \mu_{B_{comp}}(y_1) = Max\left(\mu_{B_{comp}}(y_1)\right)\right) \tag{4.51}$$

Método de la media de los máximos

El método MOM obtiene un promedio de los valores y_1 para los cuales la función de membresía es máxima. O sea, se obtiene la media del conjunto de valores para y_1 que sean valores máximos de la función de pertenencia. Por lo que se puede obtener simplemente como el promedio de calcular el SOM y el MOM. Esto se expresa en (4.52).

$$y^* = \frac{y_{SOM} + y_{LOM}}{2} = Prom\left(\forall\, y_1 \in V_1 \,|\, \mu_{B_{comp}}(y_1) = Max\left(\mu_{B_{comp}}(y_1)\right)\right) \tag{4.52}$$

4.4.2. Métodos basados en Área

Los métodos basados en área intentan obtener un valor de y_1 tal que el área de la figura geométrica que representa la función de membresía se reparta de diferentes maneras. Es decir, se busca que el valor de salida exprese de alguna manera como está distribuida la función de membresía dentro del universo de discurso. En este contexto existen las siguientes vertientes: método bisector ([Toolbox, 1995–2015]), método del COG a veces también llamado centro de área (en inglés Center of Area) (COA) ([Ross, 2009] y [Lee, 1990]), método de promedios ponderados a veces llamado centro de los conjuntos (en inglés Center of Sets) (COS) ([Ross, 2009] y [Mendel, 2007]). A continuación se explican estos métodos.

Método bisector

El método bisector toma como valor de salida para la variable y, el punto en el cual la función de membresía queda dividida en dos figuras con áreas iguales. Es decir, parte la figura geométrica de la función de membresía en dos regiones con áreas equivalentes. Lo anterior se puede expresar matemáticamente como se presenta en (4.53).

$$y^* \mid \int_{y_{ini}}^{y^*} \mu_{B_{comp}}(y_1) = \int_{y^*}^{y_{fin}} \mu_{B_{comp}}(y_1) = \frac{1}{2} \int_{y_1 \in V_1} \mu_{B_{comp}}(y_1) \tag{4.53}$$

donde $\int_{y_{ini}}^{y^*} \mu_{B_{comp}}(y_1)$ representa el área comprendida entre el inicio de la función de membresía y el punto donde se tiene tiene justamente la mitad del área. Asimismo el término $\int_{y^*}^{y_{fin}} \mu_{B_{comp}}(y_1)$ es la otra mitad del area de la figura.

Método del centro de gravedad (COG)

El método del centroide o centro de gravedad (en inglés Center of Gravity) (COG) es el más conocido para realizar la defusificación dentro de un sistema de inferencia. Parte del procedimiento homónimo dentro de la física, que sirve para calcular el punto en el cual se tendría en equilibrio una masa con una forma dada. Esto es, en este punto se tiene el efecto resultante de la gravedad sobre todas las porciones del objeto en cuestión. Por otro lado, se relaciona este método de defusificación con el concepto de centroide o baricentro en geometría. El centroide se conoce como el punto en el que se intersectan todos los hiperplanos que dividen a una figura geométrica de dimensión n_d en regiones de igual área.

En el contexto de la lógica difusa su interpretación es el valor de la variable de salida, que resulta de sumar las ponderaciones de cada valor que asume y_1 con su valor de membresía correspondiente, divido entre la suma de todos los valores de membresía posibles. Estas sumas son de naturaleza continua, por lo que se interpretan como integrales. Así el método del centroide para el conjunto difuso de salida B_{comp}, con función de membresía asociada $\mu_{B_{comp}}(y_1)$, se calcula según (4.54).

$$y^* = \frac{\int_{y_{ini}}^{y_{fin}} y_1 \mu_{B_{comp}}(y_1)}{\int_{y_{ini}}^{y_{fin}} \mu_{B_{comp}}(y_1)} \tag{4.54}$$

donde y_{ini} y y_{fin} representan el primer y último valores de y_1 para los cuales existe una membresía no nula.

Método de centro de los conjuntos (COS)

El método del centroide es adecuado para calcular un valor nítido a partir de un conjunto difuso. Sin embargo, requiere una carga computacional bastante considerable. Si se

supone que las funciones de membresía de los conjuntos difusos dentro de los consecuentes mostradas en (4.48) tienen formas simétricas. Se puede emplear una simplificación al método del centroide.

Esta simplificación es conocida como centro de los conjuntos (en inglés Center of Sets) (COS) o también se le llama método de promedios ponderados. El valor de salida se calcula como la suma de los centros de cada conjunto ponderados por la fuerza de activación resultante de cada conjunto (estas fuerzas se calculan según (4.47)). Así el método del centro de los conjuntos (en inglés Center of Sets) obtiene el valor nítido como se observa en (4.55).

$$y^* = COS(\mathcal{C}, \mathcal{F}) = \frac{\sum_{l=1}^{NC_{red}} C_l F_l}{\sum_{l=1}^{NC_{red}} F_l} \tag{4.55}$$

donde $\mathcal{C} = \{C_1, C_2, \ldots, C_{NC_{red}}\}$ son los centros de los conjuntos de los consecuentes (no repetidos) y $\mathcal{F} = \{F_1, F_2, \ldots, F_{NC_{red}}\}$. Cuando se trabaja con funciones de membresía simétricas calcular estos centros resulta un proceso sencillo, en el sentido de que se pueden usar fórmulas preestablecidas.

La obtención de los centros depende de la forma que tengan las funciones de membresía. Por ejemplo si éstas son triángulos isósceles o funciones trapezoidales ya existen fórmulas geométricas predeterminadas. Por esto, en aplicaciones donde se requiere velocidad de procesamiento es más común implementar el COS en lugar del COG, y usar funciones simples como las triangular (Λ), trapezoidal (Π), monotónamente creciente (Γ) y monótonamente decreciente ($\mathbb{L}$) (Definiciones 28, 29, 30 y 31). En (4.56) y (4.57) se muestra como obtener el centro de un triángulo escaleno y un trapecio, respectivamente. Las funciones Γ y $\mathbb{L}$ pueden considerarse un tipo de trapecio así que se usa la misma formula. En el caso cuando un trapecio es simétrico, el centro se obtiene como se observa en (4.58).

$$C_\Lambda = a_\Lambda + \frac{2}{3}(m_\Lambda - a_\Lambda) + \frac{1}{3}(b_\Lambda - m_\Lambda) \tag{4.56}$$

$$C_\Gamma = C_\mathbb{L} = C_\Pi = \frac{(d_\Pi - a_\Pi)}{3}\frac{2(d_\Pi - a_\Pi) + (c_\Pi - b_\Pi)}{(d_\Pi - a_\Pi) + (c_\Pi - b_\Pi)} \tag{4.57}$$

$$C_\Pi = \frac{(d_\Pi - a_\Pi)}{2} \tag{4.58}$$

Ejemplo de aplicación de la lógica difusa

Para ilustrar el proceso de fusificación, la inferencia y la defusificación, se plantea el siguiente ejemplo. Se desea controlar el nivel de riego de un cultivo y se cuenta con mediciones de la humedad del suelo. Entonces la variable de entrada es el porcentaje de humedad y la variable de salida es el porcentaje de apertura de la valvula de agua. Los universos de discurso $U_1 = V_1 = [0, 80]$ de estas variables se dividen en tres conjuntos difusos, 'bajo', 'medio' y 'alto', aquí se consideran los porcentajes hasta 80 pensando en que en la practica jamas se abrirá la valvula al máximo y tampoco se tendrá un porcentaje de humedad total. los tres conjuntos se representan por medio de funciones triangulares (Λ) con parámetros $a_{\Lambda_1} = 0, m_{\Lambda_1} = 20, b_{\Lambda_1} = 40$, $a_{\Lambda_2} = 20, m_{\Lambda_2} = 40, b_{\Lambda_2} = 60$ y $a_{\Lambda_3} = 40, m_{\Lambda_3} = 60, b_{\Lambda_3} = 80$. Entonces los cojuntos difusos (tanto para la variable de salida como de entrada) tienen la forma de la Figura 4.11(a) en donde la forma azul corresponde al conjunto 'bajo', la naranja al conjunto 'medio' y la roja al conjunto 'alto'. Se supone además que un experto determinó que existen 3 reglas a seguir:

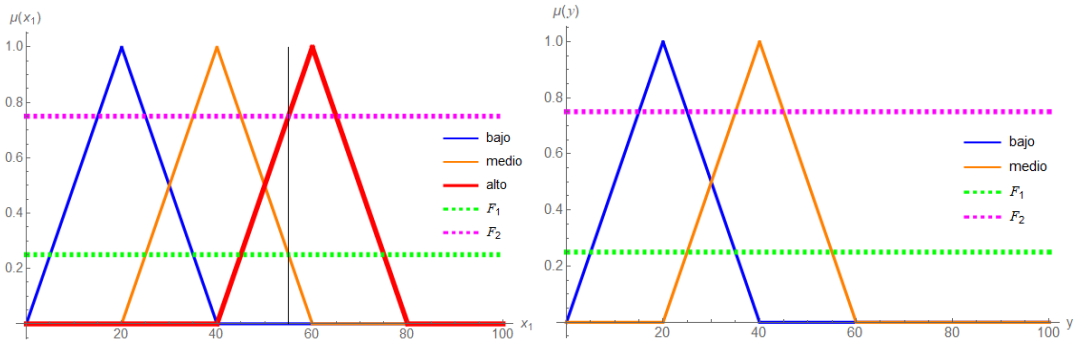
1 **Si** " el porcentaje de «humedad» es 'bajo' " **Entonces** " «el porcentaje de apertura de la válvula» debe ser 'alto' " .

2 **Si** " el porcentaje de «humedad» es 'medio' " **Entonces** " «el porcentaje de apertura de la válvula» debe ser 'medio' " .

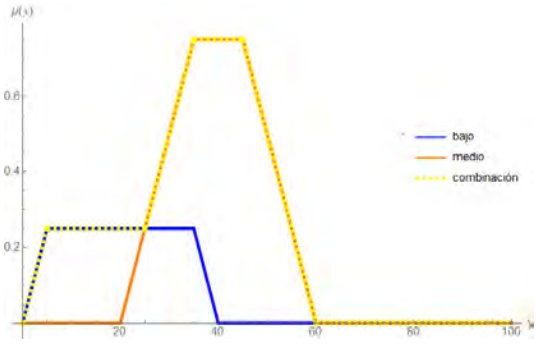
3 **Si** " el porcentaje de «humedad» es 'alto' " **Entonces** " «el porcentaje de apertura de la válvula» debe ser 'bajo' " .

Si la variable de entrada (porcentaje de humedad) toma un valor de $x_1 = 55$, su fusificación genera una pertenencia al conjunto 'medio' de $\mu_{medio}(x_1 = 55) = 0.25$ (línea verde de la Figura 4.11) y otra al conjunto 'alto' de $\mu_{alto}(x_1 = 55) = 0.75$ (línea morada de la Figura 4.11). Estas pertenencias activan las reglas 2 y 3 con una fuerzas de activación de $F_1 = 0.25$ y $F_2 = 0.75$, respectivamente (ver Figura 4.11(b)). Estas funciones se combinan, tomando en cuenta las fuerzas de activación dando como resultado la función de membresía que se observa en la Figura 4.11(c) (usando las funciones máximo y mínimo como la unión e intersección). En estas figuras las funciones de membresía (para la variable

de salida) activadas son la azul y naranja y las fuerzas de activación son la verde y morada y la forma amarilla representa la función de membresía resultante que posteriormente se defusifica.



(a) ***Conjuntos difusos de entrada y salida*** **(b)** ***Conjuntos activados y fuerzas de activación***



(c) ***Función de membresía resultante***

Figura 4.11: Ejemplo de control de riego

En la Tabla 4.1 se presentan los valores numéricos que regresa cada método de defusificación para este ejemplo, se puede apreciar que la diferencia entre COG y COS es pequeña ($< 5\,\%$) al igual que con el bisector, en tanto que COS y SOM para este ejemplo obtienen el mismo valor. Se aprecia que no varian muchos los resultados entre todos los métodos presentados, sin embargo se puede considerar como el mejor método al COS ya que es el método que está menos sesgado y también es uno de los más eficientes en cuanto al tiempo.De forma complementaria en la Figura 4.12(a) se muestra la respuesta para SOM, en la Figura 4.12(b) para MOM, en la Figura 4.12(c) para LOM, en la Figura 4.12(d) para

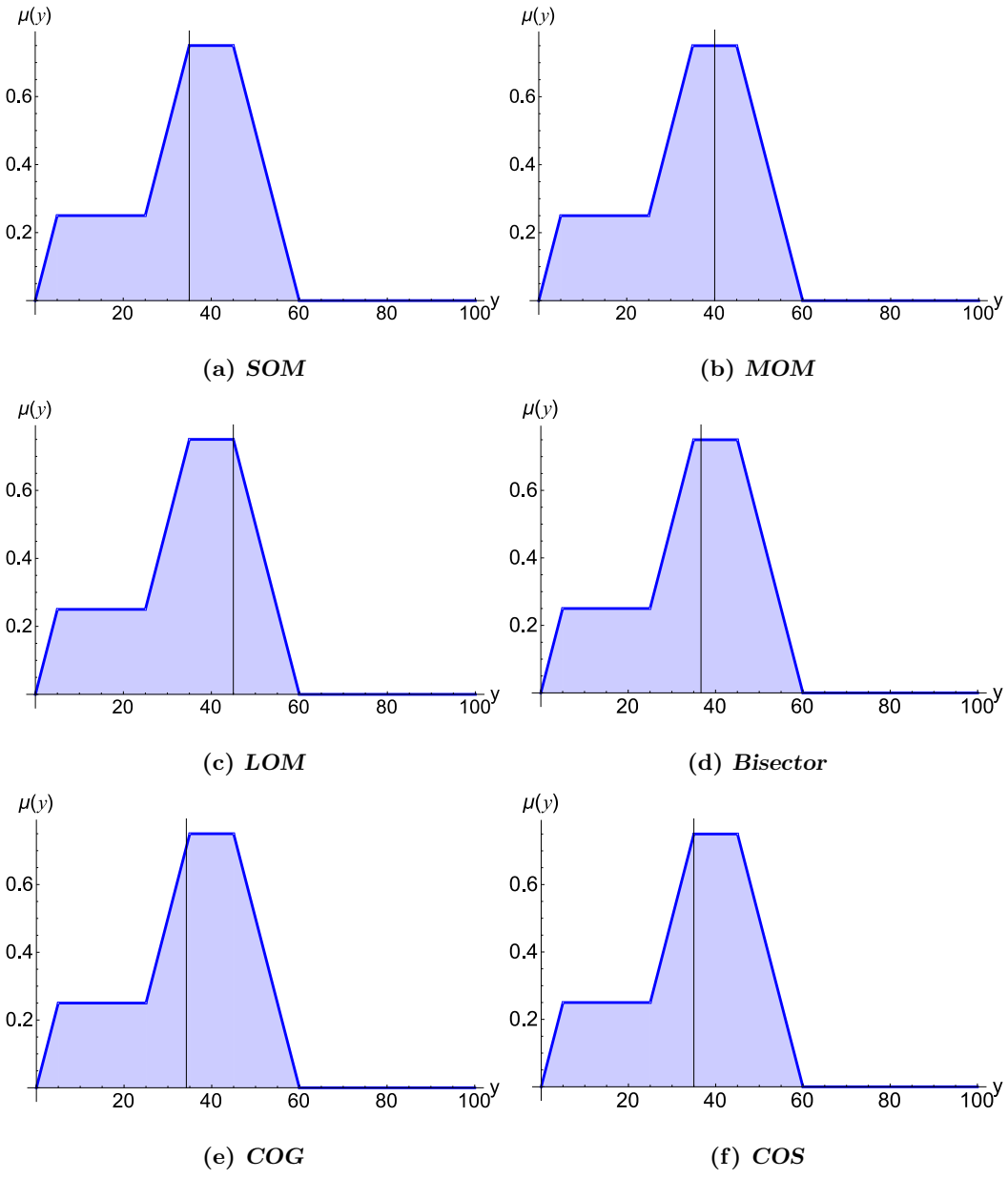
bisector, en la Figura 4.12(e) para COG y en la Figura 4.12(e) para COS.

Método de Defusificación	Valor nítido y^*
SOM	35.0000
MOM	39.9950
LOM	44.9900
Bisector	34.2105
COG	36.6667
COS	35.0000

Tabla 4.1: Respuesta de los diferentes métodos de defusificación

Conclusiones del capítulo

En este capítulo se mencionó la teoría más relevante de lógica difusa, destacando las partes de un sistema difuso: la fusificación, la inferencia y la defusificación. En la inferencia se abordaron los pasos que se siguen tales como activación de las reglas, calculo de las fuerzas de activación, combinación y composición de los consecuentes. Después se explicaron los métodos más importantes que se usan para llevar a cabo la etapa de defusificación en un sistema difuso, se resaltan los métodos COG y COS. El primero porque es el que pondera más justamente la contribución de cada punto contenido en la función de membresía resultante. El segundo al ser una simplificación del COG resulta aún más atractivo ya que requiere una menor cantidad de tiempo para regresar un resultado.



(a) *SOM* **(b) *MOM***

(c) *LOM* **(d) *Bisector***

(e) *COG* **(f) *COS***

Figura 4.12: Interpretación gráfica de los diferentes métodos de defusificación

Capítulo 5

Diseño e Implementación

En este capítulo se explica como se diseñó el sistema de FF. En primera instancia se mencionan las fases que contiene este sistema y posteriormente se abordan los algoritmos que se siguen dentro del sistema FF.

5.1. Diseño del sistema de pronóstico difuso

El sistema de pronóstico de series de tiempo que se realiza en la presente tesis se basa en la idea de que situaciones actuales en la serie de tiempo pueden parecerse a situaciones del pasado, de modo que la predicción actual puede ser basada en los siguientes valores de esos casos similares. Partiendo de esta idea, el primer punto a resolver sería buscar una manera de expresar la información (del pasado), que permita abstraer el comportamiento de la serie. Esto tendría por consecuencia poder ponderar que tanto se parecen las situaciones del pasado a las actuales.

Una forma de ver el sistema de pronóstico es en dos fases principales, una etapa de aprendizaje (entrenamiento) y otra de validación. En la etapa de aprendizaje se extrae la información relevante de la serie de tiempo, esto se hace creando una base de conocimiento. En tanto que en la etapa de validación se hacen pronósticos utilizando los datos actuales y la base de conocimiento; esto se realiza por medio de la inferencia difusa. El sistema de pronóstico es un mecanismo que procesa la serie de tiempo, pero en base a la lógica difusa; así que también está compuesto por las partes esenciales de un sistema difuso, que se mostraban

en la Figura 4.4. Al diseñar el sistema se necesitan ambas maneras de entenderlo, desde el punto de vista de la lógica difusa, así como desde el enfoque de series de tiempo.

El procedimiento que se sigue en la fase de aprendizaje es: definir la forma que tendrá la colección de conjuntos difusos, que se guardan en la base de conocimiento. Después se crea una lista de vectores de retardo, que contienen la información relevante de la serie de tiempo; posteriormente se fusifican estos datos según la colección de conjuntos difusos que se definió. Finalmente con la versión difusa de los vectores de retardo se crea la base de reglas, la cual se agrega a la base de conocimiento.

En la etapa de validación, primeramente se transforman los datos actuales en vectores de retardo. Luego se fusifican esos datos y se comparan con las reglas que se crearon en la etapa de entrenamiento; a continuación se aplica el proceso de inferencia que se describió en el Capítulo 4. Como último paso, el resultado del proceso de inferencia se defusifica; de esta manera se obtiene el pronóstico. Para conceptualizar como está compuesto el sistema de pronóstico se presenta su estructura general en la Figura 5.1.

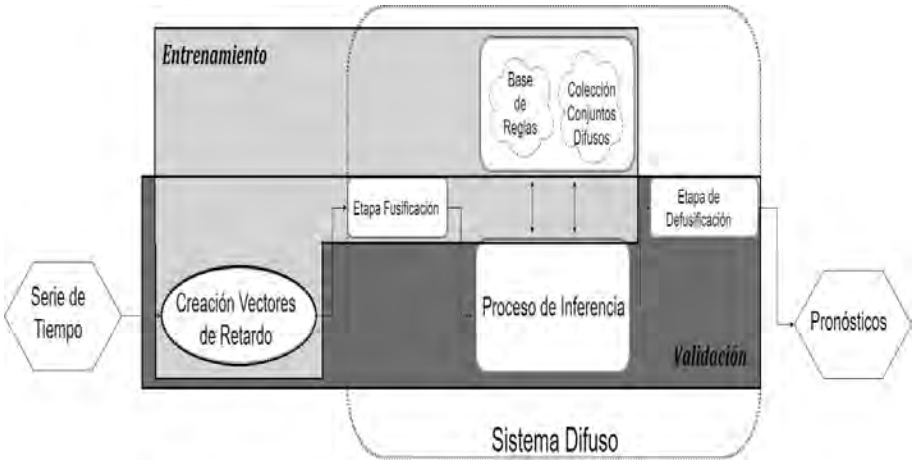


Figura 5.1: Estructura General del Sistema de Pronóstico

A continuación se explica como se llevan a cabo la creación de los vectores de retardo, y cual es su finalidad; también se explica como se crea la base de conocimiento y finalmente se detalla como se realizan los pronósticos partiendo de la inferencia difusa. En esta última parte se mencionan dos vertientes en el pronóstico; un paso a futuro (en inglés One Step Ahead) (OSA) y la otra como enfoque iterativo (de n pasos a futuro).

5.1.1. Creación de los vectores de retardo

Partiendo de una serie de tiempo con la forma de (1.1), la cual cuenta con N mediciones, se pretende extraer la información más relevante de ésta, usando el concepto de vectores de retardo. Estos se explicaron en la Sección 3.3.3 (donde se mencionaba el algoritmo de pronóstico no lineal basado en vecinos cercanos). Estos vectores de retardo sirven para crear una colección de las situaciones pasadas de la serie de tiempo y son usados porque permiten representar alguna tendencia específica (contenida en la dinámica del proceso) como un conjunto de puntos en el espacio de fase.

Al final de cada vector de retardo mostrados en (3.36) se agrega el punto X_{t+1}, de esta forma los vectores de retardo expresan que si se tienen puntos parecidos a $X_{t-(m-1)\tau}$, $X_{t-(m-2)\tau}$, ..., $X_{t-\tau}$, X_t posiblemente el punto que les sigue inmediatamente se parecerá a X_{t+1}. Así cada vector de retardo estará dado por (5.1).

$$S_t = \left\{ \left\{ X_{t-(m-1)\tau}, X_{t-(m-2)\tau}, \ldots, X_{t-\tau}, X_t \right\}, X_{t+1} \right\}. \tag{5.1}$$

Para poder ponderar qué tanto se parece una situación (representada por un vector de retardo) a otra, se usa la lógica difusa. Las situaciones anteriores son capturadas en reglas difusas y la comparación con las situaciones actuales se hace por medio de la inferencia.

5.1.2. Creación de la base de conocimiento

La base de conocimiento está compuesta por una colección de conjuntos difusos y una base de reglas (ver Figura 4.10). La colección de conjuntos difusos se divide en: conjuntos difusos asociados a las entradas mostrados en (4.29) y los asociados a las salidas que se observan en (4.30). Considerando que son los datos contenidos en los vectores de retardo los que se quieren fusificar, se puede notar que las variables de entrada son puntos de la serie de tiempo, al igual que las variables de salida. Por lo anterior los conjuntos que se definen para cada entrada del sistema difuso son los mismos entre sí y para la salida. Es decir, las reglas tendrían en los antecedentes los puntos $X_{t-(m-1)\tau}, X_{t-(m-2)\tau}, \ldots, X_{t-\tau}, X_t$ fusificados y en el consecuente la versión difusa de X_{t+1}.

Colección de conjuntos

Dado que la colección de conjuntos difusos será un único arreglo que sirve para todas las variables de entrada y la de salida, sólo se debe determinar que forma tendrán procurando que se distribuyan dentro del rango de la serie de tiempo. O sea, se definirán conjuntos difusos que estén situados entre los valores mínimo y máximo que se tienen registrados de la serie de tiempo. Por consiguiente la colección de conjuntos que se expresaba en (4.29) y (4.30) quedará resumida como se observa en (5.2).

$$\mathcal{A} = \{A_1, A_2, \ldots, A_{NC}\} \tag{5.2}$$

donde *NC* representa el número de conjuntos difusos que se usan dentro del rango de la serie de tiempo, así que el universo de discurso U es el rango $[Min(X), Max(X)]$. El término *NC* se convierte en un parámetro de entrada del algoritmo difuso.

Ahora bien, se debe definir la forma de los conjuntos A_i expresados en (5.2), se podría elegir cualquiera de las mencionadas en las Definiciones 28 a 35. Sin embargo, comúnmente se usan las funciones triangulares (Definición 28) para representar los conjuntos difusos intermedios, o sea, los que no se sitúan en los extremos del universo de discurso. Asímismo se usan las funciones Γ (Definición 30) para el extremo superior y la $\mathbb{L}$ para el extremo inferior. Esto se hace debido a que con estas funciones se puede operar más rápidamente tanto al evaluarlas como cuando se quiere defusificar.

Por lo anterior los conjuntos $A_2, A_3, \ldots, A_{NC-1}$ serán funciones triangulares mientras que A_1 será una función $\mathbb{L}$ y A_{NC} será una función Γ. Ahora resulta necesario definir que tanto se han de extender los conjuntos, es decir, en que medida se traslapan unos con otros. Se ha decidido que existan dos posibilidades, la primera que se distribuyan los conjuntos uniformemente sobre el rango y que cada conjunto se traslape solo con los dos adyacentes, en la mitad del tamaño del conjunto. La segunda opción es que los conjuntos se repartan asumiendo que los datos de la serie de tiempo están distribuidos normalmente con una media $\bar{X}$ (media muestral) y con una varianza σ_X (que toma también el valor de su equivalente muestral).

Distribución uniforme

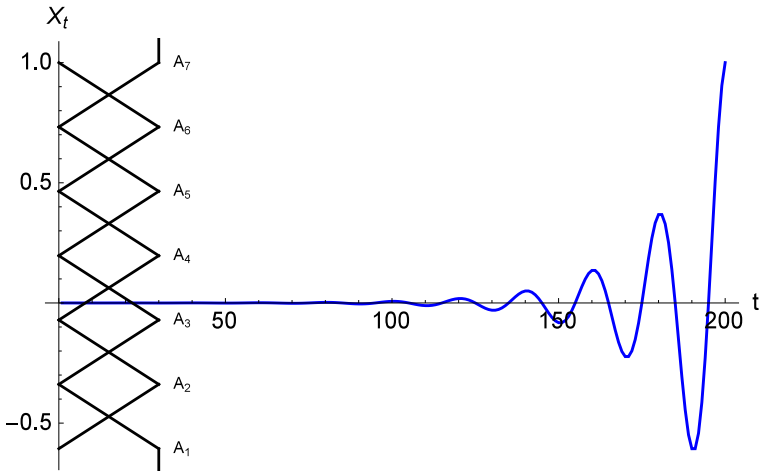
En este enfoque se distribuyen los conjuntos uniformemente, de modo que exista la misma posibilidad de que un dato cualquiera, pertenezca a cada conjunto. Se optó porque los conjuntos se traslapen solo con los más cercanos hacia un lado y otro ya que eso permite que cada punto se clasifique en bandas. Si se usaran conjuntos clásicos un punto solamente pertenecería a un conjunto específico; de esta manera un punto en la serie de tiempo pertenece solo a dos conjuntos y a cada uno de ellos con cierto grado. Por la misma razón los conjuntos difusos se traslaparán hasta la mitad uno de otro; esto sería como una manera de discretizar los valores que toma la serie de tiempo en cada instante, pero con la ventaja de que la discretización no se restringe a una sola partición.

Es de poco interés que cada punto pertenezca a todos los conjuntos difusos o a una gran parte de ellos. Esto ocasionaría que una situación genere demasiadas reglas y aún peor que estas reglas realmente no representen la situación que se está fusificando. En cuanto a los conjuntos de los extremos se decidió que la parte donde la función de pertenencia no es constante, quede traslapada con los conjuntos adyacentes, esto hasta la mitad de este vecino. Mientras que la parte constante se extiende fuera del rango, hasta un punto en el cual el centro de dichos conjuntos quede situado exactamente sobre el mínimo o el máximo de la serie de tiempo, según sea el caso. Lo anterior se hace buscando que en los extremos la fusificación sea igual que con los conjuntos intermedios y también que al momento de hacer una defusificación utilizando los conjuntos de los extremos, el valor resultante no salga del rango.

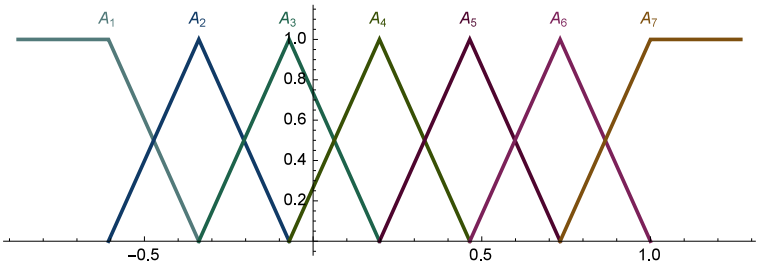
Se puede definir un vector que contendrá los límites de los conjuntos los cuales tienen la forma de (5.3), en donde la función de pertenencia $\mathbb{L}$ tendría los límites λ_1 y λ_2, cada función triangular tendría de forma genérica los limites $\lambda_{j-1}, \lambda_j, \lambda_{j+1}$ para $j = 2, 3, \ldots, NC-1$ y la función Γ los límites λ_{NC-1} y λ_{NC}.

$$\begin{aligned} \boldsymbol{\lambda} &= \{\lambda_1, \lambda_2, \ldots, \lambda_{NC}\} \\ \lambda_i &= \begin{cases} X_{min} \text{ si } i = 1 \\ \lambda_{i-1} + \frac{X_{max} - X_{min}}{NC-1} \text{ otro caso} \end{cases} \end{aligned} \tag{5.3}$$

La Figura 5.2 ilustra como se distribuyen los conjuntos difusos cuando $NC = 7$. Es decir, cinco conjuntos con funciones de membresía triangulares, uno con Γ y otro con $\mathbb{L}$, esto dentro de un universo de discurso $U = [-0.60653, 1]$. Lo anterior para una serie de tiempo de 200 muestras (las muestras se tomaron al revés) generada a partir de una función coseno que decrece exponencialmente con frecuencia angular igual a 2π. La Figura 5.2(a) muestra los conjuntos distribuidos en el eje vertical de la gráfica de la serie de tiempo y la Figura 5.2(b) muestra únicamente los conjuntos, en este caso el eje horizontal sería los valores de la serie de tiempo X y el eje vertical son los valores de pertenencia. En este ejemplo los límites $\boldsymbol{\lambda}$ serían $\lambda_1 = -0.60653, \lambda_2 = -0.33878, \lambda_3 = -0.07102, \lambda_4 = 0.19673, \lambda_5 = 0.46449, \lambda_6 = 0.75224, \lambda_7 = 1$.



(a) ***Conjuntos difusos en el rango de la serie de tiempo, en el plano*** $t - X_t$



(b) ***Conjuntos difusos en el plano*** $X_t - \mu(X_t)$

Figura 5.2: Distribución uniforme de los conjuntos difusos

Distribución normal

En este caso los conjuntos también tienen funciones de membresía L, Λ y Γ asociadas. El traslape que existe entre los conjuntos ahora se obtiene mediante un arreglo que contendrá los puntos en los cuales los conjuntos triangulares tienen cada uno de sus tres parámetros. Para calcular lo anterior, se usa la función de distribución acumulativa (en inglés Cumulative Distribution Function) (CDF) y su inversa función de distribución acumulativa inversa (en inglés Inverse Cumulative Distribution Funtion) (ICDF) de la siguiente manera. Se obtienen la media $\bar{X}$ y la desviación estándar σ_X muestrales, asimismo se deben calcular el mínimo y máximo de la serie de tiempo. Posteriormente se calcula el valor que toma la CDF para $X_{min} = Min(X)$, expresado como $\kappa_1 = CDF(N(\bar{X}, \sigma_X), X_{min})$; en tanto que κ_{NC} se fija a 1. Así que el rango de la función acumulativa es $[\kappa_1, \kappa_{NC}] = [\kappa_1, 1]$. Después se particiona este rango de la CDF en $NC - 1$ regiones, empezando en κ_1. Esto queda representado por un arreglo $\boldsymbol{\kappa} = \{\kappa_1, \kappa_2, \ldots, \kappa_{NC}\}$ mostrado en (5.4).

$$\begin{aligned} \boldsymbol{\kappa} &= \{\kappa_1, \kappa_2, \ldots, \kappa_{NC}\} \\ \kappa_i &= \begin{cases} CDF(N(\bar{X}, \sigma_X), X_{min}) \text{ si } i = 1 \\ \kappa_{i-1} + \left(\frac{1-\kappa_1}{NC-1}\right) \text{ otro caso} \end{cases} \end{aligned} \tag{5.4}$$

Finalmente, para calcular los límites de cada conjunto, se usa la ICDF aplicada al arreglo $\boldsymbol{\kappa}$, lo cual regresa una lista ($\boldsymbol{\lambda}$) en la que cada punto λ_i expresa el valor dentro del rango de la serie de tiempo para el cual ya se ha acumulado κ_i en la CDF. O sea, que los puntos λ_{i-1} y λ_i tienen comprendida la región en la cual se encuentran distribuidos $100 * \lambda_i\,\%$ de los datos. Sin embargo al aplicar la ICDF en muchas ocasiones los términos κ_1 y κ_{NC} quedan fuera del rango de la serie, o bien son incongruentes con el resto. Lo anterior se resuelve considerando que estos puntos deben coincidir con el máximo y mínimo de la serie de tiempo. De esta manera para la función $\mathbb{L}$ se considera que la región en la que decrementa empieza en X_{min} y termina en κ_2; en tanto que para la función Γ usada su región de incremento comienza en κ_{NC-1} y termina en X_{max}. Por lo anterior, el vector que contiene los límites de los conjuntos se calcula como se observa (5.5).

$$\begin{aligned} \boldsymbol{\lambda} &= \{\lambda_1, \lambda_2, \ldots, \lambda_{NC}\} \\ \lambda_i &= \begin{cases} X_{min} \text{ si } i = 1 \\ X_{max} \text{ si } i = NC \\ ICDF(N(\bar{X}, \sigma_X), \kappa_i) \text{ otro caso} \end{cases} \end{aligned} \tag{5.5}$$

Considerar que los datos están distribuidos normalmente se hace con la finalidad de que en las regiones donde existe una mayor concentración de puntos se tengan más conjuntos difusos. De esta manera se le da una mayor granularidad (una especie de resolución) a las regiones que más lo ocupan.

Para aclarar este procedimiento considerese la misma serie de tiempo generada por medio de una señal que es el producto de una exponencial y una función coseno, que se observaba en la Figura 5.2, asimismo se usan nuevamente 7 conjuntos difusos. Entonces la CDF tiene la forma que se aprecia en la Figura 5.3(a), en donde también se aprecian las divisiones que se calculan según (5.4) y son:

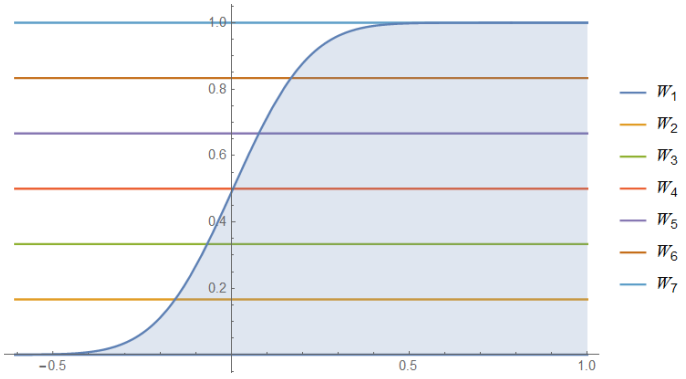
$$\{\kappa_1 = 0.00013, \kappa_2 = 0.16618, \kappa_3 = 0.33342, \kappa_4 = 0.50007, \\ \kappa_5 = 0.66671, \kappa_6 = 0.83336, \kappa_7 = 1.0\}.$$

Mientras que en la Figura 5.3(b) se muestra la forma que toman los conjuntos en el rango de la serie de tiempo donde sus límites son:

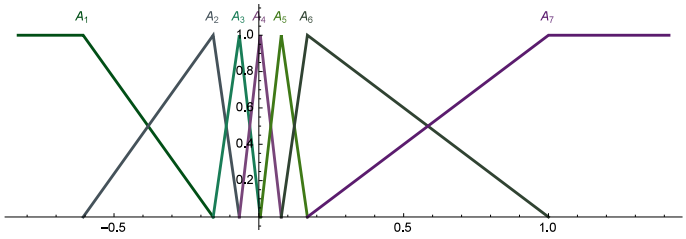
$$\{\lambda_1 = -0.60653, \lambda_2 = -0.15177, \lambda_3 = -0.06739, \lambda_4 = 0.00502, \\ \lambda_5 = 0.07844, \lambda_6 = 0.16768, \lambda_7 = 1.0\}.$$

Base de reglas

Una vez que se ha definido como crear los conjuntos difusos, que se usarán para procesar la serie de tiempo, la siguiente fase es crear la base de reglas difusas. Esta base de reglas se crea en el proceso de aprendizaje. Los pasos a seguir para crear las reglas son como se explica a continuación.



(a) *Limites en la CDF*



(b) *Conjuntos difusos en el rango de la serie de tiempo*

Figura 5.3: Distribución de los conjuntos difusos, considerando una distribución normal $N(\bar{X} = 0.00499, \sigma_X = 0.16816)$

Se toma cada vector de retardo S_t y se fusifica, esta fusificación se puede realizar por medio de (4.32) o (4.33). Además se tiene en cuenta que el número de variables (NV) es $m+1$, es decir, la longitud de cada vector de retardo. Considerando que cada punto en la serie de tiempo se fusifica según (4.33) se obtiene la representación difusa del vector de retardo S_t, como se aprecia en (5.6) en donde los conjuntos difusos tienen la forma de (5.7) y sus membresías asociadas son como se observa en (5.8).

$$(\mathcal{A}_t, \boldsymbol{\mu}_{\mathcal{A}_t}) = Fuzz(x = S_t) \tag{5.6}$$

$$\mathcal{A}_t = \{A_1, A_2, \ldots, A_m, A_{m+1}\} \tag{5.7}$$

$$\begin{aligned}\boldsymbol{\mu}_{\mathcal{A}_t} = \big\{&\mu_{A_1}(x_1 = X_{t-(m-1)\tau}), \mu_{A_2}(x_2 = X_{t-(m-2)\tau}),\\ &\ldots, \mu_{A_m}(x_m = X_t), \mu_{A_{m+1}}(x_{m+1} = X_{t+1})\big\}\end{aligned} \tag{5.8}$$

Cuando se fusifica usando los dos conjuntos difusos a los que puede pertenecer cada punto dentro de cada vector de retardo S_t, entonces habrá una colección que contiene los $m+1$ grupos de conjuntos difusos que se activaron y otra en la que se encuentran los grados de pertenencia a cada conjunto. En (5.9), (5.10) y (5.11) se muestra como se representa la fusificación de S_t, los conjuntos a los que pertenece cada punto en S_t y las membresías de esos conjuntos, respectivamente. Cabe mencionar que cuando se elige fusificar con los dos conjuntos a los que pertenece cada punto se generan 2^{m+1} reglas por ese vector de retardo. Esto ya que cada vector tiene $m+1$ puntos y se hacen las todas las posibles combinaciones. En el otro caso sólo se genera una regla por vector ya que se toma por cada punto solo el conjunto con mayor pertenencia.

$$(\mathcal{A}_t, \boldsymbol{\mu}_{\mathcal{A}_t}) = Fuzz(x = S_t) \tag{5.9}$$

$$\mathcal{A}_t = \{A_{j,1}, A_{j,2}, \ldots, A_{j,m}, A_{j,m+1}\} \tag{5.10}$$

$$\begin{aligned}\boldsymbol{\mu}_{\mathcal{A}_t} = \big\{&\mu_{A_{j,1}}(x_1 = X_{t-(m-1)\tau}), \mu_{A_{j,2}}(x_2 = X_{t-(m-2)\tau}),\\ &\ldots, \mu_{A_{j,m}}(x_m = X_t), \mu_{A_{j,m+1}}(x_{m+1} = X_{t+1})\big\}\end{aligned} \tag{5.11}$$

en donde cada término $A_{j,i}$ representa los j conjuntos asociados a cada punto i dentro de S_t para $i = 1, 2, \ldots, m+1$ y $j = \{1, 2\}$.

Cada vector de retardo ya fusificado se convertirá en una regla de la base de conocimiento. Sin embargo, la cantidad de reglas se puede reducir debido a que es posible que al ir creando las reglas dos o más situaciones (vectores de retardo) generen la misma regla. Por esta razón también se introduce el concepto de la "fortaleza de la regla", la cual es una ponderación que sirve para indicar en que grado influyeron varias situaciones para la creación de cada regla. De esta manera existirá un vector $\mathcal{FR}$ que contendrá las NR fortalezas, el cual tiene la forma de (5.13), en donde cada fortaleza FR_k se calculará como el promedio de la intersección de los antecedentes de los vectores de retardo que la generan. Es decir, por cada vector $S_{t'}$ que al ser fusificado pueda generar esa regla, se calculará la intersección de las membresías $\{\mu_{A_1}(x_1 = X_{t'-(m-1)\tau}) \wedge \mu_{A_2}(x_2 = X_{t'-(m-2)\tau}) \wedge \ldots \wedge \mu_{A_m}(x_m = X_{t'}) \wedge \mu_{A_{m+1}}(x_{m+1} = X_{t'+1})\}$ y se guardará ese valor en su correspondiente FR_k, al existir otro vector que active los mismos conjuntos, el valor de FR_k se actualiza según (5.12).

$$FR_k = \frac{t' FR_k + I(\mu_{A'_{t',i}})}{t'+1} \quad t' = 1, 2, \ldots, NR_k \tag{5.12}$$

$$\mathcal{FR} = \{FR_1, FR_2, \ldots, FR_{NR}\} \tag{5.13}$$

donde $k = 1, 2, \ldots, NR$ que es el número de reglas que quedarán finalmente en la base de datos, en tanto que NR_k representa el número de reglas que coinciden con la regla R_k. Esto considerando que los datos de entrada son los vectores de retardo de (5.1) y se fusifican los datos según (5.9) (ya que es el caso más genérico). Así que la base de reglas toma la forma de (5.14), en la que ya no aparece explícitamente a partir de qué vectores fue generada cada regla. En su lugar solamente se muestran los conjuntos que la conforman y cada regla R_k está dada por (5.15).

$$\mathcal{R} = \{R_1, R_2, \ldots, R_{NR}\} \tag{5.14}$$

$$R_k = (A_{k,1}) \wedge (A_{k,2}) \wedge \ldots \wedge (A_{k,m}) \wedge (A_{k,m+1}) \tag{5.15}$$

donde la implicación ya está considerada como la intersección (implicación de Mamdani) entre antecedentes $(X_{t-(m-1)\tau}, X_{t-(m-2)\tau}, \ldots, X_t)$ y consecuente (X_{t+1}).

5.1.3. Generación de pronósticos utilizando la inferencia difusa

La inferencia difusa es el proceso que se encarga de realizar los pronósticos. Una vez que se han definido la forma que tienen los conjuntos difusos de entrada y se tiene creada la base de reglas se procede a realizar inferencias para los puntos posteriores de la serie de tiempo. Es decir ahora se extraerán vectores de retardo para los puntos siguientes a X_N. Dependiendo de cuantos puntos se quieren pronosticar se usan dos enfoques: un paso a futuro (en inglés One Step Ahead) (OSA) y varios pasos a futuro (n), el cual se conoce como enfoque iterativo.

Predicción de un paso a futuro (OSA)

Dada una serie de tiempo $X = X_1, X_2, \ldots, X_N$, en el enfoque OSA se busca hacer una estimación del valor que tendrá X_{N+1}. También se pueden hacer estimaciones de puntos posteriores de la serie de tiempo $X_{N+1}, X_{N+2}, X_{N+3}, \ldots, X_{N+n}$, pero siempre teniendo en cuenta que para calcular X_{N+k} se debe contar con los valores de la serie de tiempo hasta X_{N+k-1}, esto considerando que $k = 1, 2, \ldots, n$. Por esta razón se conoce como de un paso a futuro ya que solo se estima el valor inmediato al último punto que se conoce. $\hat{X}_{N+1}$ denota el pronóstico del punto X_{N+1}, enseguida se explica como se obtiene este valor.

Para calcular la estimación $\hat{X}_{N+1}$ se parte del vector de retardo asociado a X_N (o sea S_N), que tiene la forma de (5.1). Sin embargo, carece de sentido incluir el último término X_{N+1} del vector S_N, ya que es el valor que se busca; así que cuando se realiza pronóstico la última posición de los vectores de retardo se ignora. Entonces el vector se fusifica de acuerdo a (5.9), (5.10) y (5.11) si se usan los dos conjuntos a los que pertenece cada punto; o bien a (5.6), (5.7) y (5.8) considerando sólo el conjunto de mayor pertenencia. Los datos fusificados para S_{N+k} tendrán de manera genérica la forma de (5.16), donde los conjuntos a los que pertenece son como se observa en (5.17), y sus funciones de membresía asociadas como se aprecia en (5.18).

$$
\begin{aligned}
Fuzz(x = S_{N+k}) &= \left(\mathcal{A}_{N+k}, \boldsymbol{\mu}_{\mathcal{A}_{N+k}}\right) && (5.16)\\
\mathcal{A}_{N+k} &= \{A_{j,1}, A_{j,2}, \ldots, A_{j,m}\} && (5.17)\\
\boldsymbol{\mu}_{\mathcal{A}_{N+k}} &= \big\{\mu_{A_{j,1}}(x_1 = X_{N+k-(m-1)\tau}), \mu_{A_{j,2}}(x_2 = X_{N+k-(m-2)\tau}),\\
\ldots, \mu_{A_{j,m}}(x_m = X_{N+k})\big\} && (5.18)
\end{aligned}
$$

donde $i = 1, 2, \ldots, m$, $j = \{1, 2\}$, $k = 1, 2, \ldots, n$ y n es el número de puntos que se quieren pronosticar usando OSA. Además se debe tomar en cuenta que cuando $k > 1$ los valores de X_{N+1} hasta X_{N+k-1} se agregaron a la serie de tiempo.

Entonces para cada regla activada su fuerza de activación nueva se obtendrá de hacer la intersección de la fortaleza de la regla con su fuerza de activación anterior. Tomando en cuenta esto y que se está trabajando con los vectores de retardo, las fuerzas de activación $\mathcal{F} = \{F_1, F_2, \ldots, F_{NR_{activ}}\}$ se calculan según (5.19) y la base de reglas tiene la forma de (5.14).

$$F_k = \mu_{A_{k,1}}(x_1 = c_1) \wedge \mu_{A_{k,2}}(x_2 = c_2) \wedge \ldots \wedge \mu_{A_{k,NV}}(x_{NV} = c_{NV}) \wedge FR_k \qquad (5.19)$$

donde los términos FR_k se toman de (5.13) y se considera que cada término $\mu_{A_{k,i}}(x_i = c_i)$ es la pertenencia de la variable x_i (que tomó el valor c_i) al conjunto $A_{k,i}$ (asociado al i-ésimo antecedente de la k-ésima regla activada).

Se prescindirá de combinar los consecuentes con sus fuerzas de activación y de la composición de consecuentes, ya que se usará como método de defusificación el COS que se presentó en (4.55). Considerando los conjuntos $\mathcal{A}_{red} = \{A_1, A_2, \ldots, A_{NC_{red}}\}$ mostrados en (4.45) y las fuerzas de activación $\mathcal{F}_{red} = \{F_1, F_2, \ldots, F_{NC_{red}}\}$ que se observan en (4.47) se puede calcular la salida nítida si primero se obtienen los centros $C_1, C2, \ldots, C_{NC_{red}}$ de los conjuntos. Como se usarán funciones triangulares, $\mathbb{L}$ y Γ los centros se pueden obtener según (4.56), (4.57) y (4.58) presentadas en el Capítulo 4, Sección 4.4.

Entonces el pronóstico $\hat{X}_{N+k}$ es la respuesta nítida del sistema difuso. Se debe tomar en cuenta que posiblemente después de fusificar los datos, estos no coincidan con

ninguna regla existente en la base de conocimiento. Debido a lo anterior se plantea como posible solución que en estos casos se obtenga como pronóstico sencillamente el valor X_{N+k-1} (con el que obviamente se cuenta). Esta forma de hacer pronóstico se conoce como NAÏVE porque es la forma más simple que se puede pensar para pronosticar. Finalmente, la predicción se calcula como se observa en (5.20).

$$\hat{X}_{N+k} = \begin{cases} y^* = \frac{\sum_{l=1}^{NC_{red}} C_l F_l}{\sum_{l=1}^{NC_{red}} F_l} & \text{Si } NR > 0 \\ X_{N+k-1} & \text{Otro caso} \end{cases} \tag{5.20}$$

Pronóstico iterativo

En este caso se pretende generar múltiples predicciones a futuro. Partiendo de que se tienen N valores en la serie de tiempo, se quieren pronosticar n valores a futuro. Es decir, se quieren calcular las predicciones $\{\hat{X}_{N+1}, \hat{X}_{N+2}, \ldots, \hat{X}_{N+n}\}$ para los valores $\{X_{N+1}, X_{N+2}, \ldots, X_{N+n}\}$. Para estimar el punto X_{N+k} se genera el vector de retardo S_{N+k-1} que está dado por la predicción anterior $\hat{X}_{N+k-1}$ y los puntos del pasado $\{X_{N+k-(m-1)\tau+1}, X_{N+k-(m-2)\tau+1}, \ldots, X_{N+k-\tau+1}\}$ tomando en cuenta que $k = 1, 2, \ldots, n$. Observe que en general las predicciones subsecuentes estarán basadas en las hechas previamente. Se debe considerar que si $n = 1$ este método se convierte en el enfoque OSA. En este enfoque se debe tener cuidado con el valor que asume n, ya que si es superior al valor de τ se usan dos pronósticos previos en el vector de retardo. Cuando se supera el valor de 2τ ahora se requieren tres pronósticos y así sucesivamente. En el caso límite, cuando n es superior a $(m-1)\tau$, el vector de retardo estará lleno de pronósticos previos.

El procedimiento a seguir es similar al explicado para el enfoque OSA. La diferencia sustancial con el de un paso a futuro es que aquí se toman pronósticos previos para generar las predicciones. Esta situación provoca que entre más lejano sea el horizonte de predicción, los resultados sean más deficientes, porque se está acumulando el error de todas las predicciones previas al calcular la actual. En este enfoque los vectores de retardo para $N + k$ se van generado conforme se tienen nuevas predicciones. Finalmente en (5.21) se muestra como se calculan los pronósticos $\{\hat{X}_{N+1}, \hat{X}_{N+2}, \ldots, \hat{X}_{N+n}\}$, en donde el primer pronóstico $\hat{X}_{N+1}$ siempre se obtiene a partir de (5.20).

$$\hat{X}_{t+k} = \begin{cases} y_k^* = \frac{\sum_{l=1}^{NC_{red}} C_l F_l}{\sum_{l=1}^{NC_{red}} F_l} & \text{Si } NR > 0 \\ \hat{X}_{t+k-1} & \text{Otro caso} \end{cases} \tag{5.21}$$

donde $k = 2, 3, \ldots, n$, por lo que y_k^* representa la salida nítida que regresa el sistema de inferencia ante el vector de entrada S_{t+k-1}, asimismo $\hat{X}_{t+k-1}$ representa la última predicción calculada.

5.2. Algoritmo de pronóstico difuso

En esta sección se explica cuales son los pasos a seguir para realizar estimaciones a futuro de series de tiempo a partir del sistema de pronóstico difuso, que se detalló en la sección anterior. Es decir, se explica en que consiste el algoritmo de pronóstico difuso (en inglés Fuzzy Forecast) (FF) y el aprendizaje difuso (en inglés Fuzzy Learning) (FL). Estos algoritmos resumen las etapas de cada parte del sistema de pronóstico difuso.

5.2.1. Descripción del algoritmo

En la Figura 5.1 se distinguían la etapa de aprendizaje (entrenamiento) y la de pronóstico (validación); a partir de estas se puede llegar a que los pasos a seguir en el sistema FF son los siguientes.

1 Transformar la serie de tiempo a una agrupación de vectores de retardo. En este punto se debe tener en cuenta que si para un vector de retardo no existen todos sus componentes (datos faltantes) este se desprecia.

2 Generar una colección de conjuntos difusos, los cuales servirán para transformar los vectores de retardo a un contexto difuso.

3 Obtener una versión difusa de los vectores de retardo calculados.

4 Crear una base de reglas a partir de la forma difusa de los vectores de retardo, en este punto se deben considerar que distintos vectores de retardo pueden generar la misma regla.

5 Calcular las fortalezas de las reglas, siendo una ponderación de cuantos vectores de retardo contribuyeron a generar una regla y en que grado.

6 Transformar los datos actuales de la serie de tiempo en vectores de retardo. En este paso se fusifican los datos que van llegando al sistema ya sea que se trabaje en tiempo real o que se cuente con los datos y solo se haga como validación.

7 Revisar si existen datos faltantes en el conjunto de validación de ser así se ignora esta predicción.

8 Verificar las reglas que se activen ante los datos de entrada. En caso de que no existan, se usa el método NAÏVE para pronosticar y termina el proceso.

9 Calcular las fuerzas de activación de las reglas que coinciden con situaciones parecidas del pasado. Se debe considerar que si dos reglas tienen el mismo consecuente su fuerza de activación sera una combinación de las fuerzas individuales de cada regla.

10 Obtener los conjuntos de salida de las reglas activadas. Aquí se deben seleccionar los conjuntos eliminando los repetidos.

11 Calcular la defusificación usando el método COS, aunque se pueden utilizar otras técnicas, el algoritmo actualmente solo contempla esta.

Para realizar lo anterior se necesitan una serie de parámetros iniciales los cuales son: la serie de tiempo (X) con su respectivo número de elementos N y el número de puntos a pronosticar n. Además se calculan el tiempo de retardo τ (usando la técnica de información mutua) y la dimensión de embebido (m) (usando la técnica de FNN). En base a los pasos anteriormente explicados y estas variables de entrada se pueden escribir formalmente dos algoritmos, el de aprendizaje difuso (Algoritmo 2) y el de pronóstico difuso (Algoritmo 3).

Los valores de salida de este algoritmo de aprendizaje son la base de reglas y sus fortalezas. En general, este algoritmo hace lo siguiente. En la Línea 2 por cada punto de la serie de tiempo se calcula su vector de retardo, entre las Líneas 3 y 5 se selecciona como se fusificarán los datos y se obtienen pares de conjuntos y funciones de membresía por cada punto contenido en cada vector de retardo. En las Líneas 6 y 7 se generan las reglas por

Algoritmo 2 $FL\,(X, m, \tau)$

1: $N \leftarrow longitud(X)$

2: $S \leftarrow S_t = \left\{ \left\{ X_{t-(m-1)\tau}, X_{t-(m-2)\tau}, \ldots, X_{t-\tau}, X_t \right\}, X_{t+1} \right\}\ \forall t \in [(m-1)\tau, N-1]$

3: **para** $S_t \in S$, $(t \in [(m-1)\tau, N-1])$ **hacer**

4: $\quad (\mathcal{A}_t, \boldsymbol{\mu}_{\mathcal{A}_t}) \leftarrow Fuzz(x = S_t)$

5: **fin para**

6: Calcular el número de reglas a generar (NR)

7: Generar reglas difusas $(\mathcal{R})$ usando los conjuntos $\mathcal{A}_t$

8: Calcular fortalezas de las reglas $(\mathcal{FR})$ usando las membresías $\boldsymbol{\mu}_{\mathcal{A}_t}$

9: **devolver** $\mathcal{R}$ y $\mathcal{FR}$

cada ventana, si existen ventanas que generan la misma regla se reducen enseguida, dando una agrupación de reglas no repetidas. Finalmente en la Línea 8 se calculan las fortalezas de las reglas las cuales se obtienen como intersecciones de las membresias de los antecedentes de las reglas por vector de retardo que empatan con cada regla.

FF obtiene el vector de retardo del punto anterior al que se quiere pronosticar, esto se hace en la Línea 3. En la Línea 4 se fusifica cada vector de retardo, así se obtiene una lista de los conjuntos a los que pertenece este vector y las membresías a cada conjunto y se evalua cuales reglas coinciden. En las Líneas 5 a 7 se verifica si ninguna regla se activa, de ser así el pronóstico será el valor inmediato anterior de la serie de tiempo para el enfoque OSA y la predicción anterior para el enfoque iterativo. En la Línea 8 se obtienen los conjuntos consecuentes de cada regla así como las fuerzas de activación de estas (considerando que se activaron reglas). Después en la Línea 9 se calcula la fuerza de activación de cada regla, como la intersección de su fortaleza de activación y las membresías de los antecedentes. En la Línea 10 se calculan los centros de los conjuntos de los consecuentes. En la Línea 11, se obtiene el valor nítido para la variable de salida usando para hacer la defusificación el método COS, las fuerzas de activación y los centros de los consecuentes.

Algoritmo 3 $FF\,(X, m, \tau, n)$

1: Obtener la longitud N de la serie de tiempo
2: **para** $1 \leq k \leq n$ **hacer**
3: Calcular S_{N+k-1}
4: $\mathcal{R}_{activ} \leftarrow (Fuzz(S_{N+k-1}), R)$
5: **si** $NR_{activ} = 0$ **entonces**
6: $\hat{X}_{N+k} \leftarrow X_{N+k-1}$
7: **fin si**
8: Extraer $A_{l,m+1}$ de $\mathcal{R}_{activ}$
9: $\mathcal{F} \leftarrow \boldsymbol{\mu}_{\mathcal{A}_k} \wedge \mathcal{FR}_k$
10: Obtener $\mathcal{C}$ de los $A_{l,m+1}$
11: $\hat{X}_{N+k} \leftarrow y_k^* = COS(\mathcal{C}, \mathcal{F})$
12: **fin para**
13: **devolver** $\hat{X}_{N+k}$

5.2.2. Implementación del Algoritmo

El sistema de pronóstico difuso se implementó en el software comercial Mathematica ®. En la práctica hubo consideranciones adicionales que se tomaron. Se adicionaron determinadas variables internas en cada algoritmo, con la finalidad de tener un mayor control sobre el sistema de pronóstico difuso. Internamente el algoritmo trabaja con parámetros como el número de conjuntos difusos (NC), este valor se debe definir antes de iniciar el aprendizaje. Se agregaron ciertos selectores lógicos que sirven para hacer más robusto el sistema, ya que permiten controlar cómo se generan las reglas, cómo se combinan, la forma de los conjuntos difusos, etc. Estos selectores son: el selector de todas las reglas (S_{TR}), selector de todos los conjuntos (S_{TC}), selector de enfoque de pronóstico (S_{EP}), selector de conjuntos variables (S_{CV}), selector de intersección (S_I) y el selector de datos faltantes (S_{DF}).

El selector S_{TR} controla si en la etapa de aprendizaje, se usarán todas las combinaciones posibles de antecedentes y consecuentes en las reglas, entonces controla si se generan muchas o pocas reglas por cada vector de retardo. El selector S_{TC} controla las combinaciones de los conjuntos en la etapa de validación, es decir, tiene la misma finalidad que el S_{TR}

pero para los datos en el pronóstico. El selector S_{EP} determina si se usará el enfoque OSA o el iterativo, pero no cambia cuantos pronósticos se desean obtener, o sea, en ambos casos se obtienen n puntos a futuro. El selector S_{CV} controla si los conjuntos se reparten en el rango de la serie de tiempo usando una distribución uniforme o una normal, de esta manera se pueden adaptar más los conjuntos difusos a los datos de la serie de tiempo. El selector S_I controla si la intersección se hace usando la función mínimo o producto. Finalmente el selector S_{DF} sirve para indicar si en la serie de tiempo faltan datos, entoces estos datos se etiquetan con un marcador de datos faltantes (M_{DF}). De esta manera cuando se encuentra un dato faltante en el aprendizaje, no se generan los vectores de retardo donde vendría ese dato. En la validación si se encuentra un dato faltante no se puede generar pronósticos y en el mejor de los casos (cuando existe el dato que precede al actual) se usa el enfoque de pronóstico NAÏVE.

Los selectores descritos en el parrafo anterior se implementaron para facilitar las pruebas del algoritmo bajo diferentes modos de operación. Por ejemplo, para el caso del S_{TR}, inicialmente solo se tomaba el conjunto al que más pertenece cada punto de los vectores de retardo, sin embargo esto hacia que hubiera pocas reglas en la base de conocimiento. Por otro lado si se escogía usar las 2^{m+1} reglas generadas por vector de retardo en situaciones donde se ocupaba hacer pruebas sencillas resultaba innecesario. Así se decidió agregar este selector y poder controlar, antes de iniciar el aprendizaje, si se usarán todas las reglas generadas por vector de retardo. El selector de todos los conjuntos se agregó pensando en que si en el aprendizaje los datos se pueden fusificar en una o varias reglas (por vector de retardo) se puede hacer lo mismo para la etapa de pronóstico con los datos actuales.

Los selector S_{DF} y el marcador M_{DF} se agregaron debido a que se observó que muchas series de tiempo no tienen completas todas las muestras. Cuando se ejecutaba el algoritmo sin tomar en cuenta los datos faltantes simplemente no funcionaba o generaba reglas que no tienen sentido. Ya que había la posibilidad de que las reglas se formaran con puntos que realmente no están cercanos en el tiempo. Por ejemplo si faltaran meses de datos en una serie de tiempo de temperatura, al no considerar que faltan las reglas se formarian con mediciones de hace meses y otras de hace dias y horas.

El S_{EP} se agregó pensando en que no se tuviera que modificar el codigo para elegir

como llevar a cabo las estimaciones futuras. En varios casos solo se ocupaba conocer el dato siguiente al actual, pero para experimentos más completos casi siempre es más atractivo hacer varias estimaciones a futuro. El S_{CV} es el que aportó un mayor cambio al sistema. Se introdujo al observar que algunas series de tiempo tenian datos complemetamente diferentes a los que se habian visto. Incialmente solo se distribuían los conjuntos de manera uniforme, pero con esos datos atípicos resultaba que una región enorme en la que realmente no había mediciones era abarcada por conjuntos difusos. Esto repercutía mucho en las reglas, las cuales representaban situaciones inexistentes. Además cuando se hacian las defusificaciones los centros de los conjuntos quedaban fuera del rango real de la serie de tiempo. Entonces con que existiera un dato atípico en el conjunto de entrenamiento era suficiente para que los conjuntos difusos se distribuyeran en un rango inadecuado. Al final se llego a la idea de repartirlos con una distribución normal para que donde se encuentran muchos datos se coloquen muchos conjuntos y donde hay datos atípicos un solo conjunto para que tenga el menor efecto posible.

En el algoritmo FL se usó un arreglo asociativo para generar las reglas y hacer su reducción. De esta manera cuando se crea una regla, fácilmente se puede verificar por medio de la llave si ya está contenida en la base de reglas; en caso de que no, se crea una entrada para esa combinación de conjuntos difusos y se almacena la fortaleza de la regla. Cuando la regla ya existe en la base de datos se accede con la llave y la nueva fortaleza de la regla se calcula como se mencionó en (5.12).

Esto más la actualización de las fortalezas de las reglas permite que el algoritmo de aprendizaje sea incremental. Es decir, cuando se tengan nuevos datos de la serie de tiempo (X_t para $t > N$) disponibles, estos pueden generar nuevas reglas, que se agregan a la base $\mathcal{R}$, o hacer una aportación a las ya existentes, siguiendo la actualización propuesta en (5.12), agregando un registro de cuantos vectores de retardo han aportado información para cada regla.

Ejemplo del uso de FF

Para ilustrar como se obtienen pronósticos usando el sistema FF y también en donde es más conveniente aplicarlo se plantea el siguiente ejemplo. Considérese una serie

de tiempo del tipo de cambio de la moneda virtual Bitcoin con el peso mexicano (MXN). En la página plus500 se pueden hacer inversiones en cambios de divisas usando los Bitcoin y muchas otras más, la compra-venta se puede realizar incluso cada minuto, así que en este tipo de situaciones se requiere un modelo rápido en el tiempo de aprendizaje y más importante en el de pronóstico. A partir de la información encontrada en este sitio se toman 60 muestras del tipo de cambio entre el Bitcoin y la divisa MXN. Estas muestras corresponden a los valores que tomó este tipo de cambio entre las 10:30 y las 11:29 am del dia 27 de septiembre del 2017. Para este ejemplo se tomarán 50 minutos o muestras como entrenamiento y 10 para validación. En la Figura 5.4(a) se muestra la gráfica del cambio entre estas divisas remarcando el conjunto de entrenamiento de color azul y en rojo el de validación. En tanto que la Figura 5.4(b) se muestra como se divide el rango de la serie de tiempo usando 5 conjuntos difusos con una distribución uniforme.

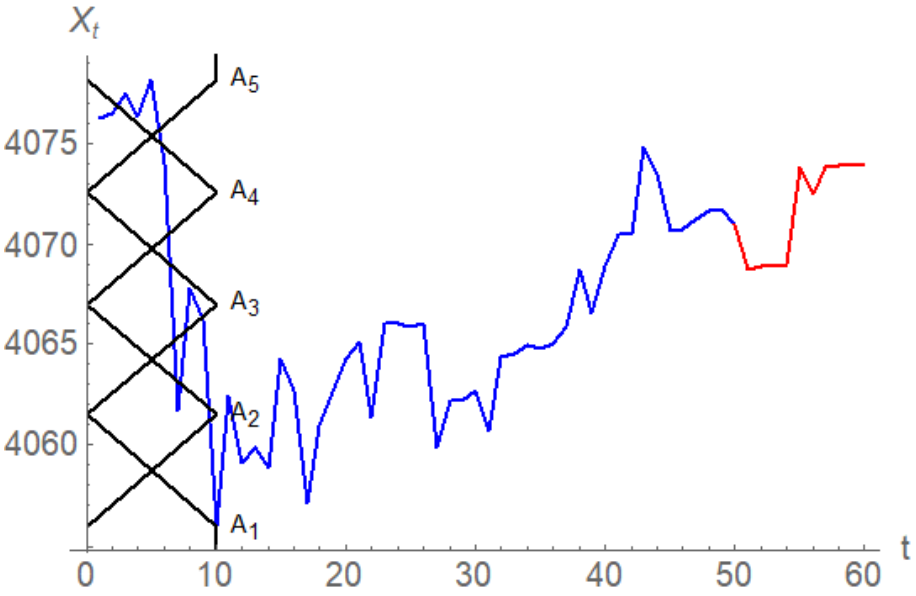
Supongase que se determinó que los valores para m y τ son 5 y 2, entonces en la etapa de entrenamiento se generarán $N-1-((m-1)\tau) = 50-8 = 41$ vectores de retardo. Tomando el primer vector de retardo (que contiene las muestras $\{\{1,3,5,7,9\},10\}$) y tiene la forma $S_1 = \{\{4076.24, 4077.54, 4078.19, 4061.69, 4066.29\}, 4055.94\}$ se puede obtener su versión difusa según (5.6) como:

$$(\{\{A_5, A_5, A_5, A_2, A_3\}, A_1\}, \{\{\mu_{A_5} = 0.649438, \mu_{A_5} = 0.883146, \mu_{A_5} = 1, \mu_{A_2} = 0.966292, \mu_{A_3} = 0.860674\}, \mu_{A_1} = 1\}) = Fuzz(x = S_1)$$

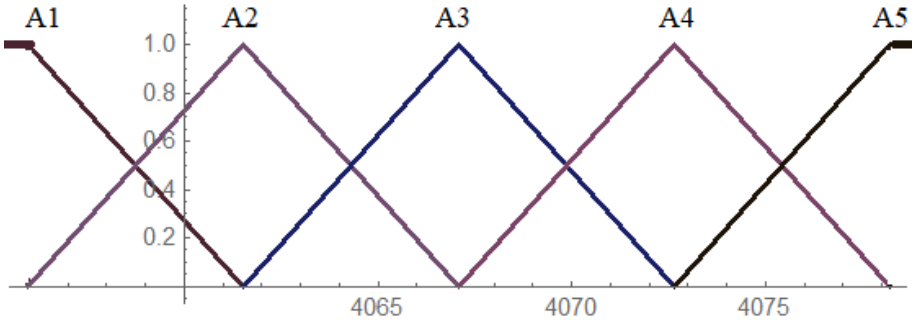
O bien si se consideran los dos conjuntos a los que pertenence cada punto según lo propuesto en 5.9 su versión difusa estaría dada por $2^{m+1} = 64$ posibilidades, ya que se hacen las combinaciones de los dos conjuntos a los que pertenece cada punto. Para este vector S_1 los dos conjuntos a los que pertenece cada punto y sus pertenencias son:

$$\{\{(A_4, A_5), (A_4, A_5), (A_5), (A_2, A_3), (A_2, A_3)\}, (A_1)\}$$
$$\{\{(0.35056, 0.64944), (0.11685, 0.88315), (1), (0.96629, 0.033708), (0.13933, 0.86067)\}, (1)\}.$$

Con la finalidad de entender como se extrae y fusifica este vector, en la Figura 5.5 se muestra la serie de tiempo desde la muestra 1 hasta la 10 (señal azul), el vector de retardo (puntos en rojo) y los conjuntos difusos (formas en negro).



(a) ***Serie de tiempo***



(b) ***Conjuntos difusos***

Figura 5.4: Cambio de divisas MXN y Bitcoin

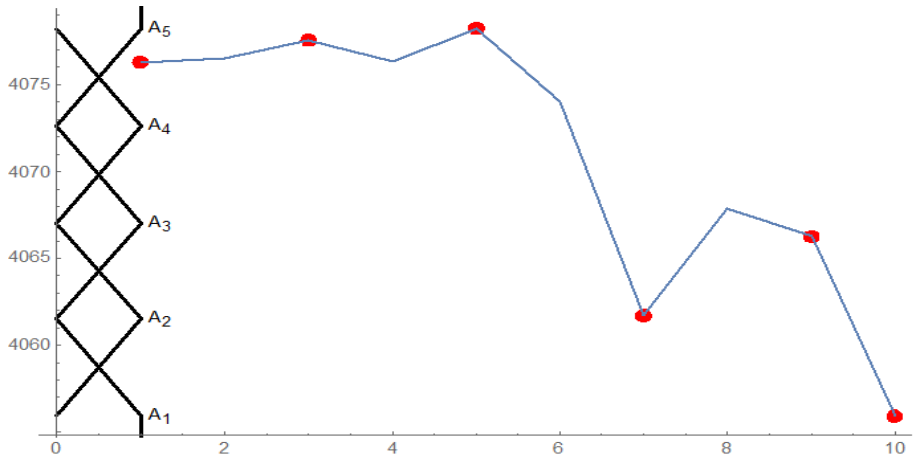


Figura 5.5: Extracción y fusificación del vector S_1

Para calcular los pronósticos finales se usaron todas las reglas posibles por vector de retardo. Ahora se busca conocer el número de reglas que se generarón, se podría pensar que son $41 * 64 = 2,624$ (64 por cada vector) pero como se dijo antes dos o más vectores pueden contribuir a una misma regla. Realmente se generaron $1,054$ reglas (34 cuando se fusifica usando solo los conjuntos de mayor pertenencia), asimismo cada una de estas reglas tiene una fortaleza asociada. Por ejemplo la primera regla que se encuentra en la base de reglas tiene la forma: $R_1 = (A_5) \wedge (A_5) \wedge (A_5) \wedge (A_2) \wedge (A_3) \wedge (A_1)$ y resulta que solo el primer vector de retardo aportó información para generar esta regla, así que la fortaleza de la regla se calculó como $Min\,(0.649438, 0.883146, 1, 0.966292, 0.860674, 1) = 0.649438$ ya que se usó la función mínimo como operador de intersección. De forma análoga se crearon las demás reglas contenidas en la base de conocimiento y con eso quedaría terminada la fase de aprendizaje.

Para la etapa de pronóstico (usando el enfoque OSA) se parte del punto $N = 50$ y se quiere predecir el 51, una vez que se tiene la muestra 51 se predice la 52 y así sucesivamente. Primero se calcula $S_N = \{4,070.54, 4,073.49, 4,070.79, 4,071.69, 4,070.99\}$ (considerando $N = 50$) que está asociado a las muestras $\{42, 44, 46, 48, 50\}$. Aquí con el S_{TC} se podría elegir fusificar considerando los dos conjuntos a los que pertenece cada punto a solo al que más, de forma similar a como se hizó con los vectores de retardo en el entrenamiento. Por simplicidad se elegirá al de mayor pertenencia únicamente, donde la

versión difusa de S_N es:

$$(\{A_4, A_4, A_4, A_4, A_4\}, \{\mu_{A_4} = 0.624719, \mu_{A_4} = 0.844944, \\ \mu_{A_4} = 0.669663, \mu_{A_4} = 0.831461, \mu_{A_4} = 0.705618\}) = Fuzz(x = S_N).$$

Entonces al evaluar cuales reglas se activaron se llega a que las reglas activadas tienen la forma:

$$R_{activ,1} = (A_4) \wedge (A_4) \wedge (A_4) \wedge (A_4) \wedge (A_4) \wedge (A_3),$$
$$R_{activ,2} = (A_4) \wedge (A_4) \wedge (A_4) \wedge (A_4) \wedge (A_4) \wedge (A_4),$$

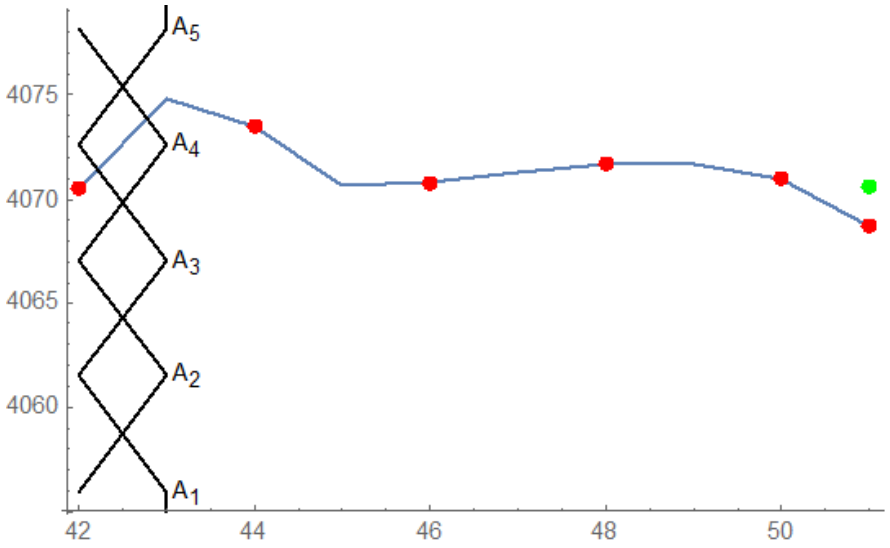
de las cuales se extraen los consecuentes A_3 y A_4, después se calculan sus centros según (4.56) y (4.57) obteniendo los valores $C_{A_3} = 4,067.065$ y $C_{A_4} = 4,072.6275$. También se obtienen las fuerzas de activación que se calculan en base a (5.19) y tienen la forma:

$$F_1 = Min(0.624719, 0.844944, 0.669663, 0.831461, 0.705618, \mathbf{0.234457}) = 0.234457,$$
$$F_2 = Min(0.624719, 0.844944, 0.669663, 0.831461, 0.705618, \mathbf{0.413483}) = 0.413483,$$

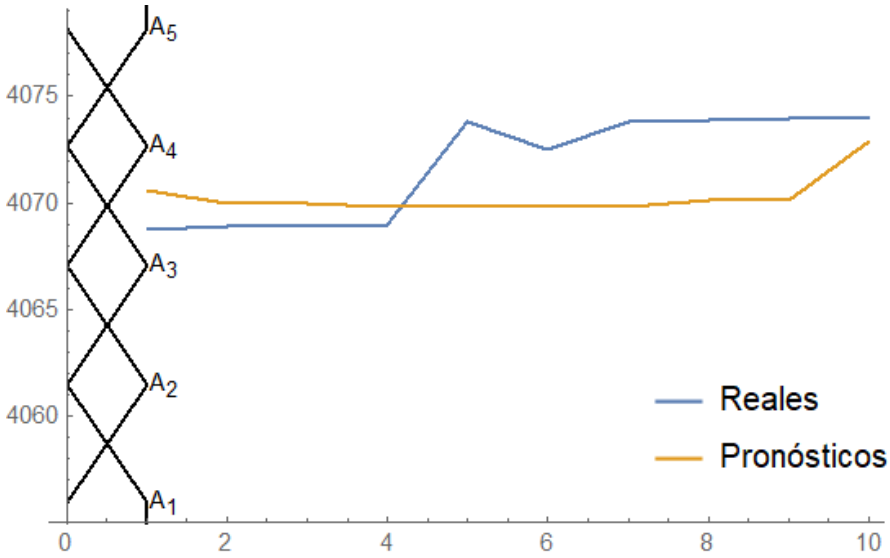
donde los números en negrita representan la fortaleza de cada regla. Finalmente el pronóstico se obtiene de acuerdo a (4.55) como sigue:

$$\frac{(4,072.6275 * 0.413483) + (4,067.065 * 0.234457)}{(0.413483 + 0.234457)} = 4,070.614711,$$

al compararlo con el valor real para la muestra 51 $(4,068.74)$ se obtiene un error MSE de 3.514538 y un MAPE de 0.046076 %. Para ilustrar el pronóstico se muestra en la Figura 5.6(a) la forma de la serie de tiempo para las muestras 42 a 51 (señal azul), el vector de retardo S_N (puntos en rojo), los conjuntos difusos (formas en negro) y el pronóstico es el punto en verde. En tanto que en la Figura 5.6(b) se observan los pronósticos hechos (señal naranja) para los 10 valores a futuro usando el enfoque OSA comparandose con los valores reales (señal azul), obteniendo un MSE de 4.37853 y un MAPE de 0.0319246 %.



(a) ***Vector de Retardo*** S_N ***y pronóstico***



(b) ***Pronósticos para las muestras*** 51 **a** 60

Figura 5.6: Pronósticos para la serie de tiempo cambio Bitcoin-MXN usando OSA

Conclusiones del capítulo

En este capítulo se abordó que el sistema de pronóstico difuso se compone de una fase de aprendizaje y otra de validación. Se exlicaron los pasos que se siguen en el aprendizaje (Algoritmo 2) y en el pronóstico (Algoritmo 3). Se mencionaron las consideraciones más importantes que se tuvieron al implementar de manera real los algoritmos. Se mencionó de manera particular que el modelo implementado tiene la caráceristica de ser incremental. Lo cual representa una ventaja sustancial con los modelos clásicos, especialmente comparándose con las redes neuronales, ya que el proceso de aprendizaje sólo se hace una vez y el sistema puede seguir incorporando información nueva sin la necesidad de volverse a contruir el modelo.

Capítulo 6

Pruebas y resultados

En este capítulo se explican las pruebas que se realizaron para evaluar el desempeño del pronóstico difuso (en inglés Fuzzy Forecast) y se presentan los resultados que se obtuvieron a partir de estas pruebas.

6.1. Condiciones de las pruebas

En esta sección se explican los casos de estudio que se plantearon y posteriormente se mencionan las medidas de desempeño que se usaron. Las cuales evaluan la precisión de los métodos y finalmente se describe las pruebas que se hicieron.

Casos de estudio

Desde un inicio se planteó que el algoritmo de pronóstico difuso se enfocaría principalmente a las series de tiempo caóticas, por esta razón como casos de estudio se usaron 24 series de tiempo generadas a partir de sistemas caóticos. Las primeras cuatro son obtenidas sintéticamente y consisten en los sistemas de Lorenz, Henon, Rossler y Mackey-Glass explicados en la Sección 3.3.

El sistema de Lorenz usado tiene la estructura de (3.31). El cual se simula (para producir una serie de tiempo, tomando la variable x) bajo las condiciones iniciales $x_0 = 0.0, y_0 = 1.0, z_0 = 0.0$ y considerando un paso de integración de 0.01. Se usaron los algoritmos de información mutua y FNN para calcular su tiempo de retardo τ

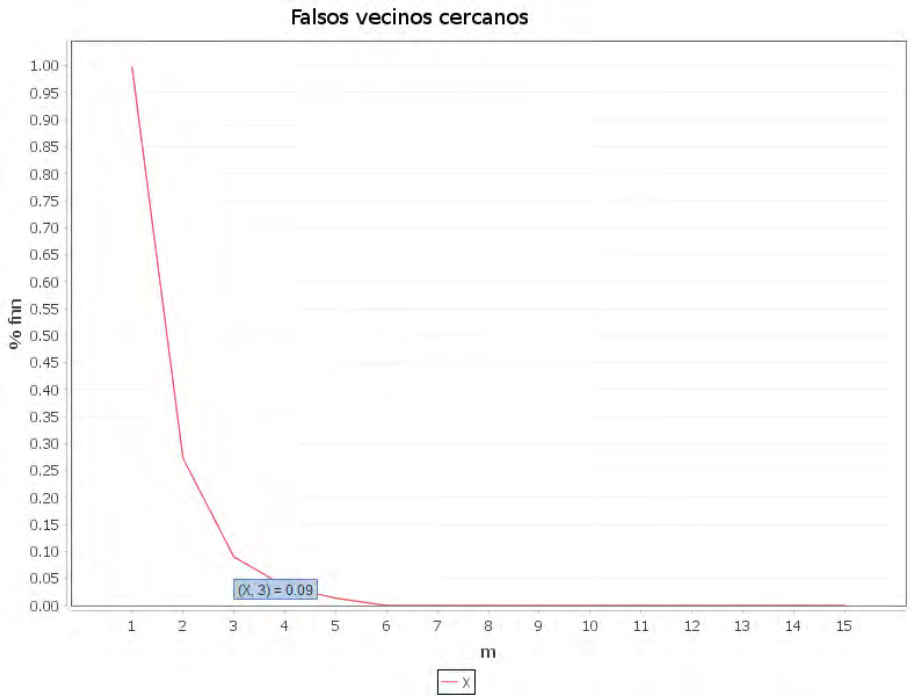
y su dimensión de embebido m, respectivamente. En la Figura 6.1(a) se muestra la gráfica con la cual se obtuvo el valor de m y en la Figura 6.1(b) se muestra como se obtuvo τ según (3.38) y (3.37)), respectivamente. Se tomaron en cuenta los criterios explicados en la Sección 3.3; o sea, se toma el primer mínimo de la función o donde haya un mayor cambio entre una dimensión y otra.

De manera similar, se toma la componente x del sistema de Henon mostrado en (3.32) partiendo de los valores iniciales $x_0 = 0.0, y_0 = 0.0$ y un paso de integración de 1. Nuevamente se calcularon los parámetros m y τ para está serie de tiempo que tomaron los valores 2 y 1, respectivamente. Esto coincide con el hecho de que el sistema de Henon es discreto, así que tiene por defecto un periodo de muestreo uno. En cuanto a la dimensión de embebido, es bastante razonable que se calcule como 2 ya que el sistema de Henon tiene esa dimensión real.

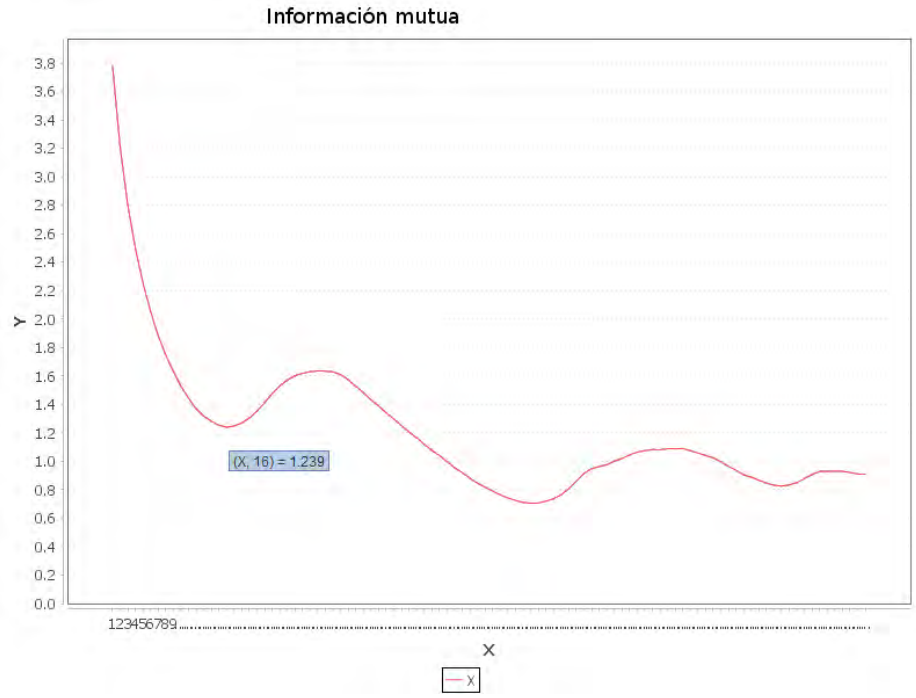
Como tercer caso de prueba se tiene una serie de tiempo generada a partir del sistema de Rossler. Se toman las condiciones iniciales $x_0 = 1.0, y_0 = 1.0, z_0 = 0.0$ para el sistema de (3.33) y su componente x, con un paso de integración de 0.02. En este caso se obtuvó una $m = 3$ (lo cual es congruente con el orden real del sistema) y una $\tau = 45$.

Por último, se tomó el sistema de Mackey-Glass presentado en (3.34) con la condición inicial $x_0 = 1.2$ y un paso de integración de 0.1. Los valores para la dimensión de embebido y el tiempo de retardo son $m = 4$ y $\tau = 124$.

Las siguientes diez series de tiempo son adquiridas de una base de datos de series de tiempo de velocidad de viento en Rusia. La medidas fueron tomadas cada tres horas y durante poco más de treinta años, en un periodo comprendido entre los años $1,966$ y $2,000$; cada serie contiene alrededor de $100,000$ muestras. Las otras diez series de tiempo también son de velocidad de viento y fueron obtenidas en diferentes municipios del estado de Michoacán. Estas mediciones se tomaron a intervalos de una hora y contienen aproximadamente $40,000$ datos. Sin embargo, representan un reto especial, ya que contienen bastantes datos atípicos, faltantes y ruido en las mediciones. En las series de viento se siguieron los mismos criterios anteriormente mencionados para calcular gráficamente los parámetros m y τ. La Tabla 6.1 muestra los nombres de las series de tiempo, su longitud y los valores estimados para la dimensión de embebido y el tiempo de retardo de cada serie. Asimismo



(a) ***Cálculo de la dimensión de embebido para Lorenz***



(b) ***Cálculo del tiempo de retardo para Lorenz***

Figura 6.1: Parámetros $m = 3$ y $\tau = 16$ para Lorenz

muestra el valor del máximo exponente de Lyapunov (λ_m)[1] mencionado en la Subsección 3.3.1, el cual es una medida para determinar que tan caótica es una serie de tiempo. El algoritmo utilizado para calcular estos exponentes es el planteado en [Wolf u. a., 1985].

	Serie de Tiempo	m	τ	Longitud	λ_m
Sintéticas	Lorenz	3	16	50,000	2.1590
	Henon	2	1	50,000	**N/A**
	Rossler	3	45	50,000	0.0714
	Mackey-Glass	4	124	50,000	**N/A**
Rusia	20891	8	1	102,272	0.0974
	22641	7	6	102,272	0.0998
	22887	8	13	102,272	0.0931
	23711	7	5	102,272	0.0870
	24908	7	1	102,272	**N/A**
	27947	6	1	102,272	0.0880
	28722	6	1	102,272	0.0977
	29231	6	5	102,272	**N/A**
	30230	6	1	102,272	**N/A**
	37099	6	1	102,272	0.1045
Michoacán	Aristeo Mercado	8	8	41,965	**N/A**
	Cointzio	6	6	42,364	0.0655
	Corrales	7	6	43,378	0.0523
	El Fresno	6	9	31,656	0.072
	La Palma	5	5	26,332	0.0759
	La Piedad	5	10	32,381	0.0496
	Malpais	9	1	22,617	0.0290
	Markazuza	5	1	43,374	0.0599
	Melchor Ocampo	5	1	32,381	0.0676
	Patzcuaro	11	1	43,651	**N/A**

Tabla 6.1: Dimensión de embebido, tiempo de retardo y longitud de las diferentes series de tiempo usadas como casos de estudio

Medidas de desempeño

Se debe contar con ciertas métricas que ponderen el desempeño en precisión de los algoritmos en el pronóstico. En la comparación de desempeño de algoritmos de pronóstico se usaron tres medidas de error: MSE, MAPE y SMAPE. Para medir el desempeño en cuanto

[1] Los términos **N/A** representan que el algoritmo no pudo calcular el exponente o se obtuvo una indeterminación debido a un alto nivel de ruido en las series de tiempo.

al tiempo se introducen dos variables: tiempo de aprendizaje (t_a) y tiempo de pronóstico (t_p).

El MSE es la medida de error más común, sirve para analizar la diferencia existente entre dos arreglos de datos (mediante un solo valor). En el caso de las series de tiempo sirve para medir la diferencia entre los datos reales X_{N+k} y sus pronósticos $\hat{X}_{N+k}$ y se puede calcular mediante (6.1).

$$\epsilon_{mse} = MSE(\{X_{N+k}\}, \{\hat{X}_{N+k}\}) = \frac{\sum_{k=1}^{n} (X_{N+k} - \hat{X}_{N+k})^2}{n} \tag{6.1}$$

El MAPE es especialmente utilizado cuando se desea comparar resultados en la serie de tiempo de diferentes ámbitos de aplicación. La magnitud del error se calcula relativa al rango en el que se encuentran los datos, así que es independiente de los datos. En las series de tiempo, esta medida de error tiene su principal aplicación al comparar los resultados de pronóstico sobre dos o más series de tiempo. La forma que toma es como se aprecia en (6.2).

$$\epsilon_{mape} = MAPE(\{X_{N+k}\}, \{\hat{X}_{N+k}\}) = \frac{100 \sum_{k=1}^{n} \mid \frac{X_{N+k} - \hat{X}_{N+k}}{X_{N+k}} \mid}{n} \tag{6.2}$$

El SMAPE se diferencia del MAPE en que se puede indicar si se hizo una subestimación o una sobreestimación. Por ejemplo, si el pronóstico es $\hat{X}_{N+k} = 90$ o $\hat{X}_{N+k} = 110$ y la medición real es $X_{N+k} = 100$, el MAPE genera el mismo resultado en ambos casos (10 %) en tanto que el SMAPE genera 10.5263 % para el primer caso y 9.5238 % para el segundo. Otra diferencia sustancial entre el MAPE y el SMAPE es que el primero toma valores entre 0 % y 100 % mientras que el segundo entre 0 % y 200 %. El SMAPE aplicado en series de tiempo se puede definir como se aprecia en (6.3).

$$\epsilon_{smape} = SMAPE\left(\{X_{N+k}\}, \{\hat{X}_{N+k}\}\right) = \frac{100 \sum_{k=1}^{n} \frac{|\hat{X}_{N+k} - X_{N+k}|}{(|\hat{X}_{N+k}| + |X_{N+k}|)/2}}{n} \tag{6.3}$$

Para poder comparar simultáneamente la precisión y el tiempo de aprendizaje de los modelos se puede recurrir a hacer un análisis costo-beneficio. Es decir, se evalua que tanto conviene aumentar el tiempo de ejecución para que mejore la precisión. De la misma

manera sirve como medida para determinar que tanto conviene reducir la precisión para que el tiempo de ejecución no sea excesivo. Esta medida debe expresar la relación que existe entre eficiencia en tiempo y precisión. Para enunciar matemáticamente esta relación se puede usar cualquiera de las medidas de error explicadas previamente y los tiempos t_a o t_p. Así para dos modelos dados, se puede medir el porcentaje de incremento del error que se denota como $\mathbf{\Delta}_\epsilon$ y se calcula según (6.4), donde ϵ_1 representa el error del modelo uno y ϵ_2 el error del modelo dos. También se mide el porcentaje de ganancia en tiempo ($\mathbf{\Delta}_t$) que se calcula como se aprecia en (6.5), donde t_1 y t_2 representan el tiempo de ejecución del modelo uno y dos. Entonces la relación costo-beneficio que se expresa como $\mathbf{\Delta}_\epsilon/\mathbf{\Delta}_t$.

$$\mathbf{\Delta}_\epsilon = 100 * \frac{\epsilon_1}{\epsilon_2} \tag{6.4}$$

$$\mathbf{\Delta}_t = 100 * \frac{t_1}{t_2} \tag{6.5}$$

Esta relación también se puede escribir en términos del número de datos en lugar del tiempo de ejecución, esto se aprecia en (6.6), donde N_1 es el número de datos para el caso uno y N_2 es el número de datos para el segundo caso.

$$\mathbf{\Delta}_\epsilon/\mathbf{\Delta}_N = \frac{100 * \frac{\epsilon_1}{\epsilon_2}}{100 * \frac{N_1}{N_2}} \tag{6.6}$$

Descripción de las Pruebas a Realizar

Se realizaron diferentes pruebas para evaluar el desempeño del algoritmo. Las primeras fueron sobre los mismos parámetros del algoritmo, para ver cuales ofrecen mejores resultados. Las segundas pruebas son comparativas con otros métodos de pronóstico y finalmente se hicieron pruebas para ver el desempeño cuando se trabaja con datos masivos.

Las pruebas con las variantes se hicieron utilizando las cuatro series de tiempo sintéticas descritas anteriormente con los parámetros de la Tabla 6.1. Se probaron los siguientes parámetros: S_{TR} (selector de todas las reglas), S_{TC} (selector de todos los conjuntos), S_{CV} (selector de conjuntos variables), NC (número de conjuntos difusos) y S_I (selector de intersección) tomando el enfoque iterativo, en donde se pronosticaron 250 puntos por

serie de tiempo. En estas pruebas solo el parámetro que se quiere evaluar es el que varía y los demás quedan fijos.

Como segundo punto dentro de las pruebas fue comparar el método contra modelos comunes en el pronóstico. En esta parte se comparó el pronóstico difuso (en inglés Fuzzy Forecast) (FF) con una red neuronal artificial (en inglés Artificial Neural Network) (ANN), el modelo autoregresivo integrado de media móvil, (en inglés Autoregressive Integrated Moving Average) (ARIMA) y con el algoritmo de pronóstico no lineal basado en vecinos cercanos (NN), adicionalmente se agregó una modificación a NN llamada NNDE. Para comparar estos métodos se usan las series de tiempo de velocidad de viento, las cuales representan casos de prueba reales. Se usó el enfoque OSA y el iterativo, donde este último tomó la forma del enfoque un día a futuro (en inglés One Day Ahead) (ODA), en las series rusas los pronósticos ODA implican 8 muestras por día y se hace para 10 días. Para ODA en Michoacán se tienen 24 muestras por día y 10 días. En esta prueba cada modelo se ajustó para que tuviera el mejor desempeño posible. Por ejemplo, para ARIMA se tenía un modelo diferente para cada serie de tiempo, NNDE calcula parámetros óptimos por cada serie, etc. En este experimento no es tan relevante como trabaja internamente cada modelo, lo que se quiere es ver cual es el desempeño de cada modelo, comparándose con los demás.

El último experimento evalúa, utilizando nuevamente las series de tiempo sintéticas, el desempeño de FF cuando se trabaja con datos masivos. Se probó la respuesta en cuanto al tiempo de ejecución y precisión para las siguientes cantidades de datos $10,000$, $20,000$, $50,000$, $100,000$, $500,000$ y $1,000,000$, en cada serie de tiempo. De esta manera, se hicieron pronósticos OSA e iterativos tomando conjuntos de validación del $1\,\%$ de los datos, donde en el enfoque iterativo se hacían predicciones por secciones. Se definió el mismo número de secciones para todos los casos (10), así que el número de puntos dentro de cada sección era el que variaba realmente. Por ejemplo, para $1,000,000$ de datos se calcularon $10,000$ pronósticos en 10 secciones de $1,000$ predicciones cada uno. Esto implica que se están haciendo pronósticos (con el enfoque iterativo) del $0.1\,\%$ de los datos. Esto para que conforme crezca el número de datos también lo haga el número de puntos dentro de cada sección (partiendo de la idea de que entre más datos se tienen más se puede pronósticar). Las predicciones para datos masivos (usando el enfoque iterativo) se hacen de acuerdo a la

Tabla 6.2.

Tamaño	Datos a pronosticar	# secciones	# Puntos por sección
10,000	100	10	10
20,000	200	10	20
50,000	500	10	50
100,000	1000	10	100
500,000	5,000	10	500
1,000,000	10,000	10	1,000

Tabla 6.2: Condiciones de pronóstico para datos masivos usando el enfoque iterativo

6.2. Resultados obtenidos

A continuación se muestran los resultados obtenidos para los tres tipos de pruebas que se aplicaron al sistema de pronóstico difuso.

6.2.1. Pruebas en los parámetros

En esta prueba se considera (a menos que se indique lo contrario) que $N = 47,750$, $NC = 20, S_{TR} = Falso, S_{TC} = Falso, S_{EP} = iterativo, S_{CV} = uniforme, n = 250$, $S_I = Min()$, $S_{DF} = Falso$ y $M_{DF} = N/A$. Asimismo m y τ se obtienen de la Tabla 6.1, además se miden el t_a y el t_p. También se mide la precisión con el MSE, MAPE y SMAPE.

El primer parámetro a probar fue el S_{TR}, para evaluar si es más conveniente generar todas las reglas posibles o sólo las de los conjuntos con mayor pertenencia. Los resultados obtenidos son como se aprecia en la Tabla 6.3. En donde las cantidades en negrita representan el mejor resultado (menor error y menor tiempo de ejecución).

Serie	S_{TR}	t_a(seg)	t_p(seg)	MSE	MAPE	SMAPE
Lorenz	Cierto	81.9617	0.3268	**60.5393**	**111.1090**	177.9630
	Falso	**55.4247**	**0.2822**	86.6514	215.5110	**167.1040**
Henon	Cierto	51.2954	0.2779	**0.6531**	**722.7600**	**92.2269**
	Falso	**42.2506**	**0.2330**	0.6558	726.3410	92.2341
Rossler	Cierto	81.2207	0.3205	**38.4907**	**132.1780**	**83.0449**
	Falso	**54.8466**	**0.2692**	404.8970	415.9340	138.5520
Mackey-Glass	Cierto	130.1740	0.3924	**0.0002**	**0.8559**	**0.8616**
	Falso	**67.2810**	**0.3300**	0.0031	4.8489	4.8342

Tabla 6.3: Prueba de desempeño para S_{TR}

En la Tabla 6.3 se observa que utilizar todas las reglas mejora la precisión considerablemente. Por ejemplo, para la serie de Rossler el MAPE es casi cuatro veces más chico cuando se generan más reglas por vector de retardo. Esto obviamente aumenta el tiempo de aprendizaje, pero es bastante aceptable ya que en el peor de los casos aumentó al doble (en la serie de Mackey-Glass). En base a esto se puede decir que vale la pena realizar el aprendizaje utilizando todas las reglas posibles. En este caso el tiempo de ejecución del pronóstico no influye en los resultados.

El segundo parámetro a evaluar fue el S_{TC}, esto nos puede decir si es mejor fusificar los datos en la validación usando todos los conjuntos a los que pertenece cada vector o solo a los que tiene una mayor pertenencia. Los resultados obtenidos se muestran en la Tabla 6.4.

Serie	S_{TC}	t_a(seg)	t_p(seg)	MSE	MAPE	SMAPE
Lorenz	Cierto	54.7305	0.5300	88.6460	**182.5640**	**152.3690**
	Falso	55.4247	**0.2658**	**86.6514**	215.5110	167.1040
Henon	Cierto	41.6634	0.3808	0.9637	770.5100	121.9410
	Falso	41.8471	**0.2335**	**0.6558**	**726.3410**	**92.2341**
Rossler	Cierto	54.4021	0.5173	**36.4997**	**67.3571**	**68.2950**
	Falso	54.2441	**0.2662**	404.8970	415.9340	138.5520
Mackey-Glass	Cierto	130.1740	**0.3924**	0.0033	**4.1937**	**4.0444**
	Falso	66.5860	0.8453	**0.0031**	4.8489	4.8342

Tabla 6.4: Prueba de desempeño para S_{TC}

A partir de la Tabla 6.4 se observa que el t_p en general incrementa al doble cuando

se usan todos los conjuntos. Sin embargo, considerar todos los conjuntos ayuda muy poco a la precisión, ya que a excepción de la serie de Rossler en las demás el desempeño en precisión es casi el mismo independientemente del valor del selector S_{TC}. Por lo anterior, se puede decir que (al menos en los casos de prueba) este parámetro no influye considerablemente en la precisión y no es necesario utilizar todos los conjuntos en el pronóstico.

Como siguiente medida a verificar se tiene el S_{CV}, para determinar si es más adecuado, al menos en los casos de prueba, usar una distribución normal o uniforme. Los resultados se aprecian en la Tabla 6.5.

Serie	S_{CV}	t_a(seg)	t_p(seg)	MSE	MAPE	SMAPE
Lorenz	normal	55.1687	0.2675	372.1437	621.9165	**163.9509**
	uniforme	54.4665	0.2621	**86.6514**	**215.5110**	167.1040
Henon	normal	42.1990	0.2447	**0.6301**	**683.0470**	93.1959
	uniforme	42.0313	0.2309	0.6558	726.3410	**92.2341**
Rossler	normal	55.0310	0.2712	**331.4864**	**361.0691**	**134.5531**
	uniforme	54.7770	0.2685	404.8970	415.9340	138.5520
Mackey-Glass	normal	67.1120	0.3289	0.0542	20.6599	23.3653
	uniforme	67.2091	0.3415	**0.0031**	**4.8489**	**4.8342**

Tabla 6.5: Prueba de desempeño para S_{CV}

En la Tabla 6.5 se puede apreciar que los tiempos de ejecución son prácticamente los mismos cuando los conjuntos difusos se distribuyen uniforme y normalmente. Las series de Lorenz y Mackey tiene mejor desempeño usando una distribución uniforme, las de Henon y Rossler cuando se usa una distribución normal. Conforme a esta información es difícil determinar cual distribución tiene un mejor desempeño. Aunque se debe tener presente que la distribución normal está pensada en series donde existen datos atípicos y como estas series de prueba no los tienen no se ve una mejora significativa. Además para una distribución uniforme, 20 conjuntos son relativamente pocos, aún se puede obtener una mejora al aumentar el número de conjuntos, en cambio para la distribución normal ya es una cantidad razonable. Ambas opciones tienen utilidad por lo que no se pueden dejar de lado.

A continuación se probó el S_I para ver si es mejor hacer la intersección por medio del mínimo o el producto. Los resultados obtenidos son como se aprecian en la Tabla 6.6.

Serie	S_I	t_a(seg)	t_p(seg)	MSE	MAPE	SMAPE
Lorenz	$Min()$	**54.3934**	0.3103	**61.1618**	**116.8779**	178.1912
	$Prod()$	54.9145	**0.2692**	86.6514	215.5110	**167.1040**
Henon	$Min()$	42.0245	**0.2335**	**0.6547**	**724.8866**	**92.2140**
	$Prod()$	**41.9031**	0.2362	0.6558	726.3410	92.2341
Rossler	$Min()$	55.0020	0.2740	**223.6504**	**262.2890**	**122.9670**
	$Prod()$	**54.7900**	**0.2697**	404.8970	415.9340	138.5520
Mackey-Glass	$Min()$	**66.6727**	**0.3357**	0.0248	12.6776	13.8191
	$Prod()$	66.9324	0.3470	**0.0031**	**4.8489**	**4.8342**

Tabla 6.6: Prueba de desempeño para S_I

La Tabla 6.6 muestra que los tiempos de ejecución son ligeramente diferentes, pero esta diferencia no es significativa, además en la mitad de los casos gana en ambos tiempos de ejecución hacer la intersección por medio del producto y en la otra mitad con el mínimo. En cuanto a la precisión en 3 de los 4 casos se tuvo mejor desempeño con la función mínimo. Por lo anterior, se puede decir que es más eficiente usar la función mínimo como el operador intersección ya que es más preciso y la diferencia en tiempo es muy pequeña.

Finalmente para probar que tanto contribuye el número de conjuntos al pronóstico se prueba para $NC = \{10, 20, 30, 40, 50, 100\}$. Los resultados se pueden apreciar en la Tabla 6.7, asímismo en la Figura 6.2 se muestra el comportamiento del tiempo de aprendizaje (t_a), en la Figura 6.3 se aprecia el tiempo de pronóstico (t_p) y en la Figura 6.4 se observa la precisión conforme aumenta el número de conjuntos basándose en el MAPE.

En la Tabla 6.7 se puede apreciar que tanto t_a como t_p aumentan conforme se usan más conjuntos difusos. En cuanto a la precisión se observa que, contrario a lo que podría esperarse, no siempre se obtiene el mejor resultado con más conjuntos difusos. Para la serie de Lorenz el mejor resultado se obtiene para $NC = 50$, para Henon y Mackey-Glass es en 20 y para Rossler es en $NC = 100$. Por lo anterior, se puede decir que usar más conjuntos no necesariamente garantiza mayor precisión.

A partir de las Figuras 6.2 y 6.3 se puede ver que la relación entre los tiempos de ejecución y el número de conjuntos difusos es lineal. En la Figura 6.4 se hace más evidente que el error MAPE no decrece monótonamente conforme aumenta el número de conjuntos difusos (NC). Esto es debido a que si se usa una cantidad muy grande de conjuntos hace

Serie	NC	t_a	t_p	MSE	MAPE	SMAPE
Lorenz	10	**38.3725**	**0.1805**	261.6790	523.1260	167.4730
	20	55.9771	0.2653	86.6514	215.5110	167.1040
	30	73.7415	0.3490	72.1239	161.1330	168.3710
	40	91.7056	0.4332	282.6810	521.4590	**159.6340**
	50	108.8490	0.5722	**63.5166**	**123.1110**	172.2140
	100	193.0200	0.9627	69.2827	144.1600	166.6980
Henon	10	**29.6582**	**0.1699**	0.6997	870.4050	99.8388
	20	42.8733	0.2301	**0.65584**	**726.3410**	**92.2341**
	30	55.5472	0.3146	1.0259	746.5490	129.9070
	40	68.2189	0.3772	1.0714	1056.7400	126.6070
	50	81.4402	0.4410	1.0907	800.3010	130.5290
	100	143.7120	0.7975	0.9424	791.2020	116.9950
Rossler	10	**38.2962**	**0.1876**	438.8870	439.6280	140.3720
	20	55.7492	0.2691	404.8970	415.9340	138.5520
	30	73.0514	0.3489	349.2070	374.5110	135.6610
	40	89.9019	0.4282	290.2480	325.599	129.7840
	50	108.664	0.5210	199.6710	237.5270	**120.6480**
	100	190.6730	0.9573	**193.8770**	**233.6880**	123.7830
Mackey-Glass	10	**46.2156**	**0.2233**	0.0230	12.0882	13.1406
	20	68.5340	0.3370	**0.0031**	**4.8489**	**4.8342**
	30	90.8440	0.4410	0.0162	9.4264	10.1472
	40	112.7890	0.5380	0.0554	20.9345	23.7067
	50	134.5010	0.6549	0.0087	6.2036	6.5723
	100	238.2910	1.2226	0.0134	8.0450	8.6328

Tabla 6.7: Prueba de desempeño para NC

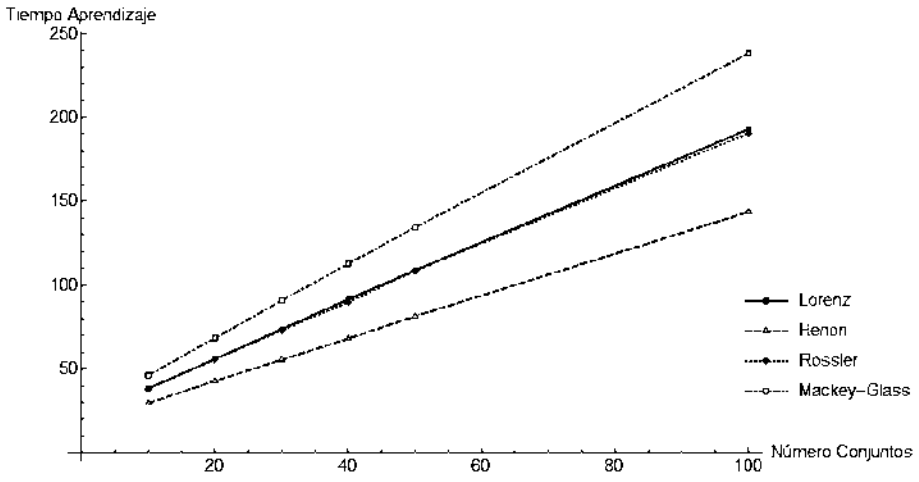


Figura 6.2: Tiempo de aprendizaje vs. número de conjuntos

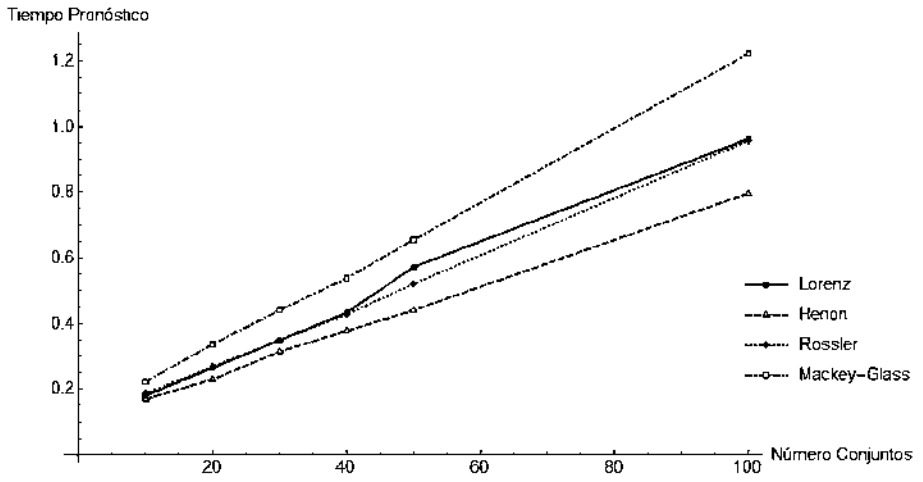


Figura 6.3: Tiempo de pronóstico vs. número de conjuntos

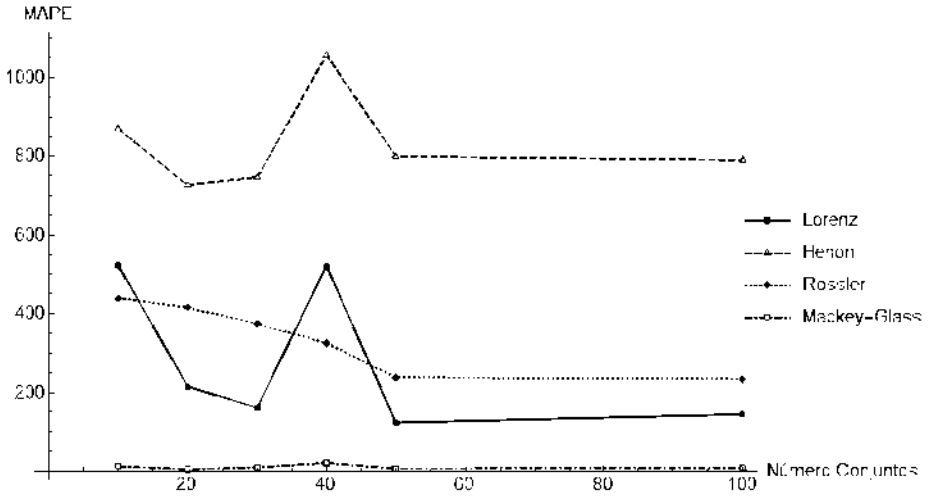


Figura 6.4: MAPE en el pronóstico vs. número de conjuntos

que dos situaciones bastante similares generen dos reglas diferentes, porque coinciden con diferentes conjuntos (que tienen un tamaño muy pequeño). Entonces este efecto hace parecer que no existen situaciones similares en el pasado y finalmente la predicción no se realiza usando lógica difusa sino el método NAÏVE. Al ser este último una técnica muy básica efecta la precisión del sistema. En el caso opuesto, muy pocos conjuntos generan muy pocas reglas que se empatan casi con cualquier situación de la serie de tiempo, pero realmente no se parecen y al defusificar y calcular los centros estos quedan muy distanciados del valor real (sería como hacer una especie de promedio).

6.2.2. Comparación entre métodos de pronóstico

Para determinar el desempeño del método propuesto en esta tesis (FF), contra los métodos ANN, ARIMA, NN y NNDE, siguiendo los experimentos descritos de la Sección 6.1. Los resultados obtenidos se muestran en las Tablas 6.8, 6.9, 6.10 y 6.11, usando los parámetros m y τ de la Tabla 6.1, tanto para FF como para NN. NNDE calcula sus propios valores de m y τ. En las Tablas 6.8 y 6.9 se muestran los resultados usando MSE como medida de error para OSA y ODA, respectivamente. En tanto que en las Tablas 6.10 y 6.11 se aprecian los resultados usando como medida de error SMAPE también para OSA y ODA.

En las Tablas 6.8 y 6.9, se aprecia que el método que obtiene un mejor desempeño es NNDE. Sin embargo, FF también obtiene un desempeño aceptable ya que es primer lugar en 5 de los 20 casos para el enfoque ODA. En la práctica, tiene mayor relevancia obtener pronósticos aceptables para el enfoque ODA que para el OSA, ya que se busca poder realizar la mayor cantidad de pronósticos con la información disponible. En las tablas 6.10 y 6.11 se aprecia nuevamente que NNDE tiene un error inferior a los otros métodos en la mayoria de los casos.

Serie	**NN**	**NNDE**	**ARIMA**	**ANN**	**FF**
20891	1.3324	**1.1303**	1.1454	4.5582	1.6882
22641	1.4736	**1.2249**	1.3915	2.3453	2.0232
22887	1.3017	0.7952	**0.7662**	1.9404	0.9231
23711	0.3572	0.2887	**0.2681**	5.4700	0.7066
24908	0.8993	**0.2920**	0.3951	0.3114	3.8117
27947	2.0727	**1.7919**	2.0578	2.5374	2.3074
28722	3.3189	**1.8658**	1.9485	5.0489	3.1781
29231	0.7296	**0.6496**	0.7226	1.1951	1.1799
30230	0.5521	**0.2412**	0.3174	0.7289	4.7282
37099	0.5980	**0.5315**	0.5708	1.0972	0.9590
aristeomercado	8.6396	9.5286	30.5793	**7.9018**	9.7058
cointzio	15.0498	**6.3672**	14.3769	7.0904	7.6163
corrales	5.8738	3.1352	5.4077	6.5962	**2.9667**
elfresno	26.4713	**9.4749**	24.5548	17.0143	15.5030
lapalma	5.3685	2.8200	4.5706	**2.8034**	3.3172
lapiedad	8.2668	**4.8590**	9.3776	8.0730	6.3834
malpais	265.3791	**206.7756**	255.3817	1134.2485	447.3400
markazuza	3.5002	**2.4588**	4.6290	13.1508	4.8123
melchorocampo	7.1123	**4.5191**	7.1231	40.2660	7.7880
patzcuaro	6.5915	**5.7043**	24.0700	8.3362	6.4960

Tabla 6.8: Resultados para OSA de los diferentes métodos usando MSE

Serie	**NN**	**NNDE**	**ARIMA**	**ANN**	**FF**
20891	2.9579	**1.4146**	3.1139	10.9384	2.5303
22641	1.8851	**1.5431**	2.2760	11.1189	2.9277
22887	3.0457	**0.8606**	1.8112	4.5098	2.2294
23711	0.9595	**0.3331**	0.4980	17.3722	1.2474
24908	0.4890	**0.2906**	0.7991	0.3924	3.3124
27947	3.4069	**2.6616**	3.6071	12.6127	4.5089
28722	3.3989	**2.7219**	3.6214	6.5057	4.9290
29231	1.0235	**0.7178**	1.2033	1.2701	1.0800
30230	0.4419	**0.2348**	0.3021	3.5384	3.8638
37099	0.5974	**0.5785**	0.6391	2.0549	1.0083
aristeomercado	24.1204	16.7335	34.2877	32.0470	**5.0148**
cointzio	45.8938	14.2265	50.3599	24.1834	**6.4294**
corrales	12.6666	3.6947	27.0927	35.4674	**2.1247**
elfresno	57.7648	13.5116	65.3169	49.3597	**2.0943**
lapalma	6.1634	**2.9760**	19.4168	4.2553	5.7179
lapiedad	14.4020	11.6457	43.8031	26.1892	**7.1864**
malpais	297.1535	**249.0032**	298.9521	302.1138	406.5020
markazuza	8.1281	**4.7288**	14.3091	74.5250	22.2188
melchorocampo	9.7091	**6.6848**	39.6040	50.2435	24.1220
patzcuaro	28.6182	**10.3706**	33.8383	34.1921	114.2330

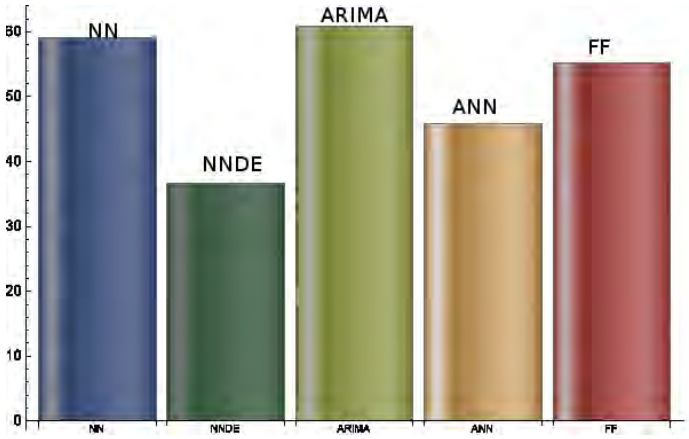
Tabla 6.9: Resultados para ODA de los diferentes métodos usando MSE

Serie	**NN**	**NNDE**	**ARIMA**	**ANN**	**FF**
20891	34.53	**29.06**	38.09	48.47	38.36
22641	63.35	**43.05**	61.59	58.86	58.84
22887	90.81	**56.40**	93.49	58.49	67.44
23711	105.03	49.17	**23.12**	84.75	111.60
24908	145.64	42.47	146.15	**26.72**	145.59
27947	77.49	44.85	52.53	**26.10**	55.77
28722	79.30	**46.08**	78.11	58.83	83.43
29231	52.34	45.13	54.11	**45.09**	65.03
30230	136.31	72.92	148.25	**33.09**	149.57
37099	40.51	**36.36**	40.09	49.57	48.48
aristeomercado	**28.82**	**28.82**	49.37	28.98	30.33
cointzio	30.23	21.19	51.55	20.11	**20.03**
corrales	32.33	24.87	49.92	36.77	**22.19**
elfresno	47.54	**28.48**	70.64	33.32	30.89
lapalma	36.44	28.16	51.05	**26.84**	27.81
lapiedad	37.04	33.17	37.48	41.07	**31.00**
malpais	48.56	**26.22**	31.39	50.09	28.96
markazuza	31.43	**28.07**	45.61	57.94	33.92
melchorocampo	28.98	**24.06**	31.99	77.42	27.07
patzcuaro	33.78	**26.17**	61.85	53.29	28.61

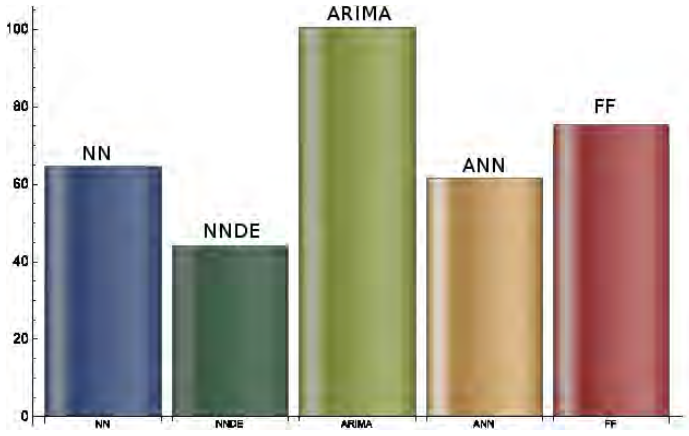
Tabla 6.10: Resultados para OSA de los diferentes métodos usando SMAPE

La información presentada en estas tablas es demasiado densa para apreciar claramente el desempeño de cada método. Con la finalidad de mostrar más claramente la respuesta de cada técnica se puede hacer un promedio de los errores obtenidos para cada serie. Esto sólo es válido para la medida de error SMAPE, ya que por su estructura permite comparar la diferencia independientemente del rango de la serie. En la Figura 6.5(a) se aprecia el valor promedio de los errores SMAPE para los diferentes modelos, considerando el enfoque de pronóstico OSA (se origina de la Tabla 6.10) y en la Figura 6.5(b) se observa algo similar pero tomando en cuenta el enfoque ODA.

A partir de estas gráficas se puede decir que el método más preciso es NNDE (con cerca de 40 % de error en promedio), en tanto que el modelo FF está situado en tercer lugar (con cerca del 65 % de error en promedio). Se observa que todos los métodos tienen menor precisión en el enfoque ODA, sin embargo a los que menos les afecta realizar más pronósticos son a NN (aumenta el error en 5 %) y NNDE (aumenta el error en 8 %) en contraste con



(a) ***Desempeño OSA***



(b) ***Desempeño ODA***

Figura 6.5: Desempeño en precisión de los modelos de pronóstico usando SMAPE para OSA y ODA

Serie	**NN**	**NNDE**	**ARIMA**	**ANN**	**FF**
20891	40.49	**33.55**	52.48	73.93	47.35
22641	71.60	**67.82**	74.54	84.12	76.62
22887	100.26	91.54	94.12	**67.35**	114.55
23711	111.49	**62.92**	91.52	104.81	120.27
24908	146.98	38.33	138.63	**29.32**	144.76
27947	57.27	53.11	56.96	**46.44**	64.74
28722	84.69	64.17	82.56	**63.37**	88.77
29231	56.00	51.06	62.33	**46.45**	64.37
30230	134.55	88.17	170.72	**59.53**	147.33
37099	41.07	**40.64**	42.97	56.02	49.84
aristeomercado	**37.39**	38.26	48.96	43.45	62.50
cointzio	43.83	**21.51**	110.46	32.49	61.98
corrales	40.79	**25.80**	153.68	80.92	54.24
elfresno	75.08	**22.07**	110.04	47.12	36.65
lapalma	37.65	**26.21**	149.31	32.29	39.92
lapiedad	45.48	**41.34**	178.53	64.37	75.10
malpais	47.66	**28.62**	42.07	37.95	62.48
markazuza	45.88	**30.76**	113.56	97.08	62.08
melchorocampo	30.95	**23.24**	133.76	79.46	40.90
patzcuaro	44.97	**32.94**	103.23	87.42	92.00

Tabla 6.11: Resultados para ODA de los diferentes métodos usando SMAPE

ARIMA que aumenta mucho el error (cerca del 40 %). En la Figura 6.5(a) se observa que FF tiene un desempeño cercano a NN (4 % de diferencia) y a la ANN (10 % de diferencia), supera a ARIMA en cerca del 5 % y es superado significativamente por NNDE con un 20 %. Se puede concluir que usando la medida de error SMAPE el desempeño de FF en las series de tiempo de velocidad de viento es aproximadamente la media de las respuestas de los demás modelos, teniendo el mejor desempeño NNDE y el peor ARIMA.

Para el MSE se cuentan los casos en los que cada modelo fue el mejor, el peor, etcétera. A partir de la Tabla 6.8 se obtuvo la Tabla 6.12 que muestra el número de veces que cada modelo quedó en cada posible lugar. Es decir, se cuenta el número de series de tiempo para las cuales cada modelo asumió los posibles lugares (son cinco modelos, así que son cinco lugares). De la misma manera se obtuvo la Tabla 6.13 a partir de los datos contenidos en la Tabla 6.9.

OSA/MSE	1° lugares	2° lugares	3° lugares	4° lugares	5° lugares
NN	0	3	**9**	5	3
NNDE	**15**	4	1	0	0
ARIMA	2	**8**	4	3	3
ANN	2	2	2	2	**12**
FF	1	3	4	**10**	2

Tabla 6.12: Posiciones por modelo para OSA y MSE

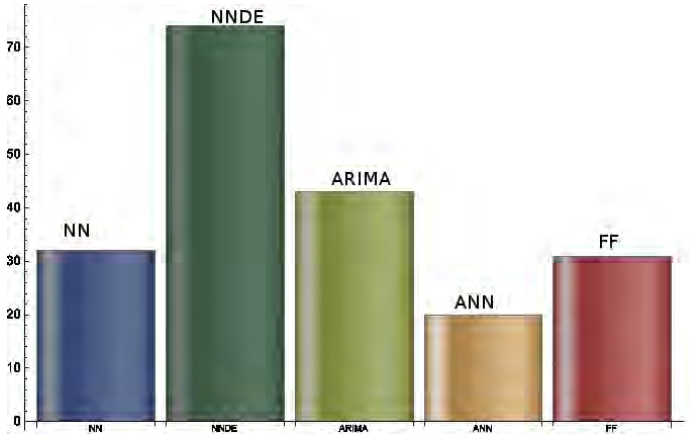
ODA/MSE	1° lugares	2° lugares	3° lugares	4° lugares	5° lugares
NN	0	**9**	**7**	4	0
NNDE	**15**	5	0	0	0
ARIMA	0	3	**7**	5	5
ANN	0	2	2	5	**11**
FF	5	1	4	**6**	4

Tabla 6.13: Posiciones por modelo para ODA y MSE

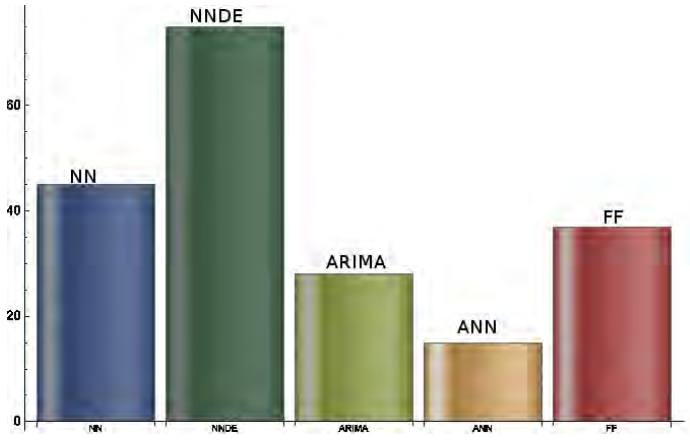
De la Tabla 6.12 se puede ver que el modelo que tiene más primero lugares es NNDE con 15, el que tiene más segundos lugares es ARIMA con 8, el que cuenta con más terceros lugares es NN con 9. Por su parte FF es el que cuenta con más cuartos lugares con 10 y ANN es quien tiene mayor número de quintos lugares con 12. En la Tabla 6.13 se aprecia que los primeros lugares se reparten entre NNDE y FF con 15 y 5, respectivamente. Los segundos lugares en su mayoria los tienen NN (con 9) y NNDE (con 5), los terceros lugares se distribuyen principalmente entre ARIMA (7) y NN (7). Los cuartos lugares están repartidos casi igual entre NN, ARIMA, ANN y FF con 4, 5, 5, y 6 ocasiones, respectivamente. Por último los quintos lugares en su mayoria los tiene ANN con 11 seguido de ARIMA con 5.

Para determinar (en general) cual es el orden que tienen los modelos para MSE se decide ponderar cada lugar obtenido por cada método. Las ponderaciones se hacen asignándole un valor de 4 a cada primer lugar obtenido, 3 a cada segundo lugar, 2 a cada tercer lugar y 1 para cada cuarto lugar, sin considerar aporte por los quintos lugares que se hayan obtenido. Al final se suman los puntos acumulados por cada lugar y este valor es la medida de que tan bueno fue el desempeño del modelo en cuestión (en el intervalo $[0, 80]$). Siguiendo este procedimiento se generó la gráfica de la Figura 6.6(a) que muestra

la ponderación de los lugares obtenidos cuando se usa el enfoque OSA. De manera similar en la Figura 6.6(b) se aprecia el desempeño con estas ponderaciones para ODA.



(a) *Desempeño OSA*



(b) *Desempeño ODA*

Figura 6.6: Desempeño en precisión de los modelos de pronóstico usando MSE para OSA y ODA

Hasta ahora, sólo se ha mencionado el rendimiento con respecto a la precisión, sin mencionar el desempeño en los tiempos de ejecución. En este sentido, el tiempo más relevante es t_a, ya que es el que implica una carga computacional mayor. En la Tabla 6.14 se muestran los tiempos aproximados de aprendizaje para cada modelo y cada serie de tiempo, donde los tiempos están redondeados a minutos. En este punto es necesario mencionar que el método FF se corrió en una computadora con 8 Gb de memoria de acceso aleatorio (en

inglés Random Access Memory) (RAM) y un procesador de Intel core $i5$ a 3.2 GHz. Los métodos NN, ARIMA y ANN, en una computadora con 16 Gb de RAM y un procesador Intel core $i7$ a 2.2 GHz. El método NNDE por sus carácteristicas no puede ejecutarse en una computadora convencional así que se corrió en un Clúster (propiedad de la División de Estudios de Posgrado de la Facultad de Ingeniería Eléctrica, UMSNH) que tiene 96 Gb de RAM y dos procesadores Intel Xeon-E5-2670 ® de 8 núcleos trabajando a 2.6 GHz . Debido a que los métodos se corrieron en diferentes equipos, la comparación en tiempo puede no ser del todo justa, especialmente para los métodos que se ejecutaron en computadoras con menos recursos. La comparación no se hace considerando complejidad computacional ya que para ANN y NNDE no es posible derivar sus complejidades.

Serie	NN-t_a (min)	NNDE-t_a (min)	ARIMA-t_a (min)	ANN -t_a (min)	FF-t_a (min)
20891	12	30	10	120	**4**
22641	12	30	10	120	**3**
22887	12	30	10	120	**4**
23711	12	30	10	120	**4**
27947	12	30	10	120	**4**
28722	12	30	10	120	**3**
29231	12	30	10	120	**3**
30230	12	30	10	120	**3**
37099	12	30	10	120	**3**
aristeomercado	5	12	4	48	**1**
cointzio	5	12	4	49	**1**
corrales	5	13	4	51	**1**
elfresno	4	9	3	36	**1**
lapalma	3	8	3	31	**1**
lapiedad	4	9	3	38	**1**
malpais	3	6	2	26	1
markazuza	5	13	4	51	**1**
melchorocampo	4	9	3	38	**1**
patzcuaro	5	13	4	51	**2**
promedio	8.15	20.22	6.74	80.89	**2.05**

Tabla 6.14: tiempos de ejecución del aprendizaje para los diferentes métodos

En la Tabla 6.14 se aprecia claramente que el método FF es el que requiere un menor tiempo de ejecución del aprendizaje; ARIMA queda en segundo lugar. El tiempo que tarda FF (máximo 4 minutos) es muy pequeño considerando que se está trabajando con decenas de miles de datos. Esto deja entrever que puede procesar grandes cantidades de datos (esto se aborda en la Sección 6.2.3) en un tiempo aceptable. Por otro lado, NNDE es

el que requiere una mayor cantidad de recursos computacionales para ejecutarse y aún así requiere media hora para el aprendizaje en las series de tiempo de velocidad de viento rusas que cuentan con cerca de $100,000$ datos. El modelo que necesita una mayor cantidad de tiempo en el aprendizaje son las redes neuronales, que para $100,000$ datos tardaban cerca de dos horas.

Considerando el análisis de costo-beneficio explicado en la Sección 6.1, se pueden comparar el desempeño de NNDE (mayor precisión) con FF (menor tiempo de aprendizaje). En realidad se podría calcular una medida para cada método con respecto a otro, pero la relación de mayor interés es comparar los dos métodos que tienen un mejor rendimiento en tiempo y precisión. Considerando los datos de la Figura 6.5(a) (para SMAPE y OSA) y los t_a de la Tabla 6.14 se puede obtener para NNDE y FF la medida expresada en (6.7), esto conforme a (6.4) y (6.5). De forma similar tomando los datos de la Figura 6.5(b) para SMAPE y ODA se obtiene la medida de (6.8).

$$\boldsymbol{\Delta}_\epsilon/\boldsymbol{\Delta}_t = \frac{100 * \frac{36.735}{55.246}}{100 * \frac{20.22}{2.05}} = \frac{66.494}{986.341} = 6.741 \tag{6.7}$$

$$\boldsymbol{\Delta}_\epsilon/\boldsymbol{\Delta}_t = \frac{100 * \frac{44.103}{75.3225}}{100 * \frac{20.22}{2.05}} = \frac{58.552}{986.341} = 5.936 \tag{6.8}$$

En ambos casos la relación costo-beneficio es pequeña, esto indica que debido a la precisión que se pierde se reduce significativamente el t_a, o también que no hay una perdida tan grande en precisión considerando la ganancia en tiempo. Para dejar más en claro esta afirmación se debe mencionar que el costo en el tiempo de aprendizaje entre NNDE y FF es casi de $10:1$. Es decir, NNDE tarda cerca de diez veces más tiempo en el aprendizaje que FF, en tanto que su relación en la precisión nos indica que disminuye el error en cerca del $33\,\%$ para OSA y $42\,\%$ para ODA. De esta manera, queda de manifiesto que FF logra obtener resultados aceptables (ya que está en el tercer lugar de los cinco) en un tiempo bastante reducido. La relación costo-beneficio enfatiza este hecho y ayuda a comprender que FF tiene un mayor equilibrio entre precisión y costo en tiempo de ejecución. Si se comparan los demás métodos contra NNDE se obtienen los valores mostrados en la Tabla 6.15, donde se incluyen los resultados tanto para OSA como para ODA.

	Δ_ϵ/Δ_t	
Modelo	OSA	ODA
NN	25.0862	27.4735
NNDE	100.0000	100.0000
ARIMA	20.1335	14.62
ANN	320.9394	285.9798
FF	**6.7414**	**5.9363**

Tabla 6.15: Relación costo-beneficio entre los diferentes modelos y NNDE para OSA y ODA

En la Tabla 6.15 se aprecia que la relación costo-beneficio más pequeña es obtenida por FF, por su parte ARIMA y NN tienen una relación cerca de 3 y 5 veces más grande que FF. La relación del 100 % obtenida por NNDE indica que se está comparando contra sí mismo, pero en general entre más grande es la relación costo beneficio significa que la ganancia en precisión tiene un costo significativo en el tiempo o que el aumento en el tiempo genera poca mejora en la precisión. Es decir, una relación costo-beneficio grande indica que el costo es grande en comparación del beneficio obtenido, por el contrario si es pequeña indica que el costo es bajo en comparación del beneficio obtenido. Así que sería una pésima elección cambiar NNDE por ANN ya que no tiene mejor precisión pero si tarda más en el aprendizaje. En este sentido el modelo que tiene mejor desempeño es FF ya que tiene la relación costo-beneficio más baja.

6.2.3. Pruebas con datos masivos

Estas pruebas se realizaron en base a las condiciones expuestas en la Sección 6.1. Se usaron conjuntos difusos siguiendo una distribución normal ($S_{CV} = normal$), los selectores S_{TR} y S_{TC} mantuvieron un valor igual a $Falso$ (cada vector en el aprendizaje genera solo una regla y los vectores en la validación también se fusifican solo tomando en cuenta los conjuntos a los que más pertenece cada punto). Se usó como función de intersección el mínimo. Además se definió que $S_{DF} = Falso$ y $M_{DF} = N/A$. En esta prueba se busca medir la precisión mediante las tres métricas de errores MSE, MAPE y SMAPE, para OSA y el enfoque iterativo. También se mide el tiempo de ejecución para aprendizaje y pronóstico (t_a y t_p). Finalmente se mide el número de reglas generadas en cada caso. La Tabla 6.16 muestra las mediciones anteriormente mencionadas para $10,000$, $20,000$, $50,000$, $100,000$,

500, 000 y 1, 000, 000 datos. En esta tabla cada celda de las medidas de error y del tiempo de pronóstico cuenta con dos elementos, el primero corresponde a los valores para OSA y el segundo es para el enfoque iterativo.

	Serie	# Reglas	t_a	t_p	MSE	MAPE	SMAPE
				OSA/Iterativo			
10, 000	Lorenz	1111	11.2226	0.1179/0.1112	0.7632/2.2438	12.0626/54.2345	13.5192/28.1838
	Henon	204	8.5836	0.0947/0.0964	0.0096/0.6421	27.2233/146.0063	17.4845/92.5136
	Rossler	1076	11.1392	0.1126/0.1085	0.5594/1.0933	12.2863/55.3211	9.3365/15.5229
	Mackey	1325	13.2583	0.14/0.1398	0.0001/0.0003	0.9644/1.2774	0.9657/1.2730
	Serie	# Reglas	t_a	t_p	MSE	MAPE	SMAPE
				OSA/Iterativo			
20, 000	Lorenz	1314	22.2298	0.2260/0.2196	1.0000/14.9740	19.1785/118.8837	15.3398,52.2615
	Henon	211	17.2367	0.1851/0.1900	0.0109/0.6291	33.1580/184.6019	20.5218,96.0257
	Rossler	1127	22.3729	0.2298/0.2242	0.3862/0.3.2437	34.8720/120.0131	12.9965,120.0131
	Mackey	1939	26.9026	0.2758/0.2736	0.0015/0.0029	4.4945/6.3946	4.7249/6.7706
	Serie	# Reglas	t_a	t_p	MSE	MAPE	SMAPE
				OSA/Iterativo			
50, 000	Lorenz	1392	56.1754	0.5644/0.5292	0.9851/58.3794	26.9552/869.0749	24.9183,92.9203
	Henon	212	42.8769	0.4610/0.4714	0.0125/0.6497	74.1387/462.0897	21.1248,95.8145
	Rossler	1154	55.7676	0.5726/0.5449	0.6533/37.0205	12.9290/251.7196	13.7103,60.6451
	Mackey	2357	68.3976	0.6968/0.6881	0.0007/0.0054	2.6219/6.1530	2.6717/6.1530
	Serie	# Reglas	t_a	t_p	MSE	MAPE	SMAPE
				OSA/Iterativo			
100, 000	Lorenz	1454	111.6724	1.1389/1.0510	0.8171/129.0025	50.4462/715.0211	16.9321/99.1257
	Henon	213	86.5190	0.9088/0.9547	0.0119/0.6551	109.3961/636.2774	21.1874/99.2574
	Rossler	1167	111.5379	1.1495/1.0775	0.8735/106.0574	15.5741/367.2754	12.2256/73.9501
	Mackey	2423	136.8869	1.3904/1.3653	0.0006/0.0116	2.2386/7.8339	2.2616/8.0600
	Serie	# Reglas	t_a	t_p	MSE	MAPE	SMAPE
				OSA/Iterativo			
500, 000	Lorenz	1541	558.3207	5.6875/5.5465	0.8059/204.4049	25.5859/1,108.5251	18.0382/134.7283
	Henon	214	430.6913	4.5548/5.0855	0.01226/0.7887	82.1521/439.7596	21.3376/111.2892
	Rossler	1188	555.9661	5.6171/5.4420	0.8008/276.1266	39.5852/530.6398	12.2556/134.2936
	Mackey	2612	682.2405	6.8795/6.6547	0.0006/0.1506	2.2320/34.0295	2.2851/41.4181
	Serie	# Reglas	t_a	t_p	MSE	MAPE	SMAPE
				OSA/Iterativo			
1, 000, 000	Lorenz	1555	1123.2207	11.2746/11.3245	0.8171/235.0602	29.0112/946.0469	18.6419/144.8181
	Henon	213	860.2369	9.0448/10.1470	0.0123/0.9207	123.3832/1,040.6479	20.9777/108.8465
	Rossler	1197	1115.3405	11.20766/11.3002	0.8073/239.1069	170.2405/2,167.9971	12.3727/129.5892
	Mackey	2653	1373.2272	13.8141/13.9773	0.0007/0.0726	2.2854/29.2106	2.3445/22.7562

Tabla 6.16: Medidas de desempeño para datos masivos

A partir de la Tabla 6.16 se puede observar que, en general, entre más datos se tienen aumentan más los valores de las medidas de error, especialmente el MAPE. Esto se enfatiza más en los pronósticos que siguen el enfoque iterativo. Por ejemplo, para la serie de Lorenz el MAPE para 10, 000 datos (en el enfoque iterativo) es de 54.23 %, cuando se tienen 50, 000 ya aumentó a 869 % y en 500, 000 es de 1108 %. Esta situación se presenta debido a que entre más datos se está tomando un horizonte de predicción más grande, de esta manera el error se va acumulando en el enfoque iterativo. Tomando como base la

información de la Tabla 6.2 presentada en la Sección 6.1 se observa que para $10,000$ datos se harán 10 secciones de 10 predicciones, o sea, que se están prediciendo 10 datos por cada $10,000$ lo que equivale a un $0.1\,\%$. Esta relación se mantiene para las demás cantidades de datos, así para $1,000,000$ de datos se hacen predicciones de secciones de $1,000$ datos ($0.1\,\%$). Sin embargo, entre más datos se tienen se observa que ya no crecen mucho las reglas, es decir, las situaciones que se presentan en su mayoria coinciden con alguna regla ya existente. Se aprecia que para la serie de Lorenz, el número de reglas tan sólo aumenta en 149, considerando $50,000$ y $500,000$ datos. Entonces, la cantidad de datos aumentó en una relación $10:1$ ($900\,\%$) mientras que la información que se aportó (número de reglas) se incrementó en una relación $1.107:1$ ($10\,\%$). De esta manera es evidente que el error debe ir incrementando ya que se está pronósticando más a futuro contando con casi el mismo número de reglas.

Para comprender que comportamiento tiene el error usando el enfoque OSA, se pueden extraer de las tablas anteriores los valores de MAPE para cada serie de tiempo. De esta manera se obtiene la Tabla 6.17. En donde los valores en negrita indican que son más pequeños que el error para un menor número de datos.

# Datos	Lorenz	Henon	Rossler	Mackey
10,000	12.0626	27.2233	12.2863	0.9644
20,000	19.1785	33.1580	34.8720	4.4945
50,000	26.9552	74.1387	**12.9290**	**2.6219**
100,000	50.4462	109.3961	**15.5741**	**2.2386**
500,000	**25.5859**	**82.1521**	39.5852	**2.2320**
1,000,000	**29.0112**	123.3832	170.2405	**2.2854**

Tabla 6.17: MAPE para las series de tiempo caóticas (sintéticas) variando el número de datos

En la Tabla 6.17 se puede observar que algunos valores del MAPE para una mayor cantidad de datos son inferiores que con menos datos. En 14 de los 24 casos, aún cuando se tiene un límite más grande para los pronósticos, disminuye el error lo cual puede indicar que entre más datos se tienen el error debería disminuir si se mantiene fijo el número de pronósticos a futuro. Esta afirmación se hace pensando en que si se tienen más datos se generan nuevas reglas y es más probable que los datos actuales empaten con una base de

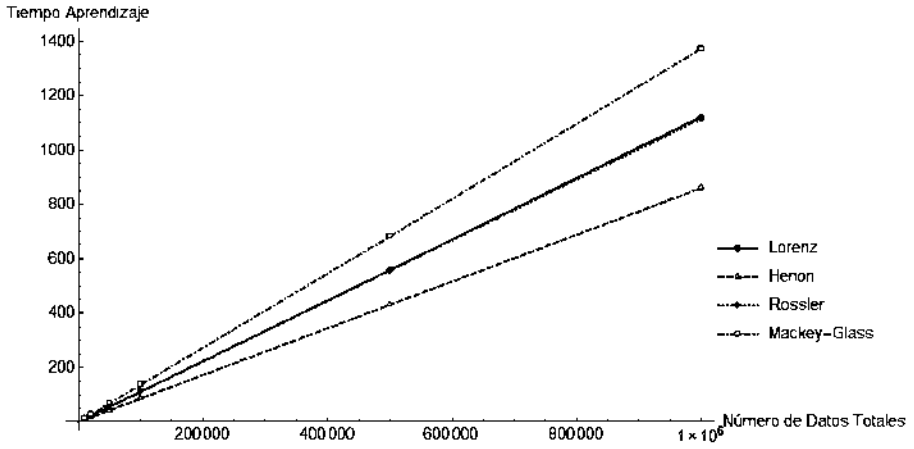
reglas más grande.

Con la finalidad de apreciar (de mejor manera) como cambian, el error, el tiempo de ejecución y el número de reglas con respecto al número de datos, se decidió graficar estas variables. Por lo que el tiempo de aprendizaje con respecto al número de datos se muestra en la Figura 6.7(a), asimismo en las Figuras 6.7(b), 6.7(c), se muestran las gráficas para el error MAPE y el número de reglas, respectivamente. El MAPE que se considera para estas gráficas es el relacionado con el enfoque iterativo, ya que esta forma de pronóstico es de mayor interés. Se decidió graficar solo esta medida de error porque está diseñada para poder hacer comparaciones entre diferentes series de tiempo (es independiente de las escalas de las series de tiempo).

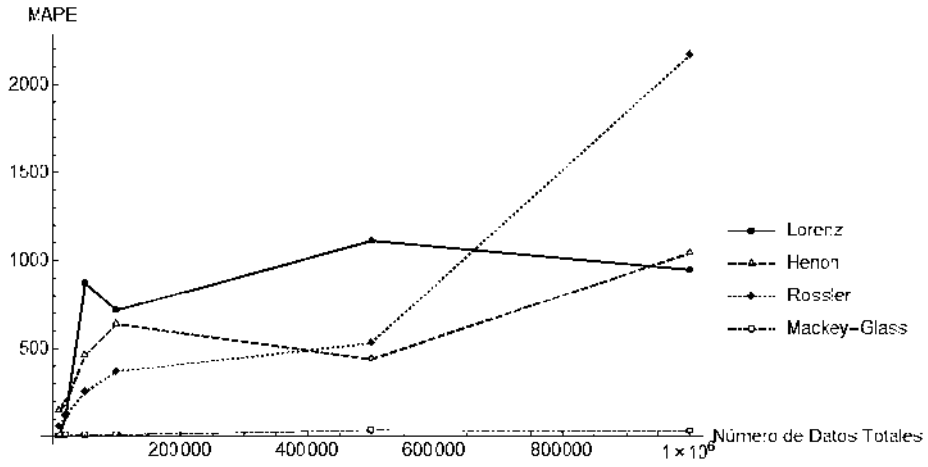
En la Figura 6.7(a) se muestra que el tiempo de aprendizaje (t_a), para las cuatro series de tiempo caóticas aumenta de manera lineal. En la Figura 6.7(b) se observa que en general el MAPE aumenta, pero tiene un comportamiento que no parece lineal. La Figura 6.7(c) muestra que el número de reglas aumenta considerablemente al principio y después de algún punto se mantiene casi constante. Lo anterior indica que después de un número de datos la información que va llegando ya se tenía contemplada (al menos hasta el horizonte con el que se cuenta). Esto va acorde a lo descrito en [Flores u. a., 2016a], donde se abordó la complejidad computacional de su método (que es la misma para FF).

Conclusiones del capítulo

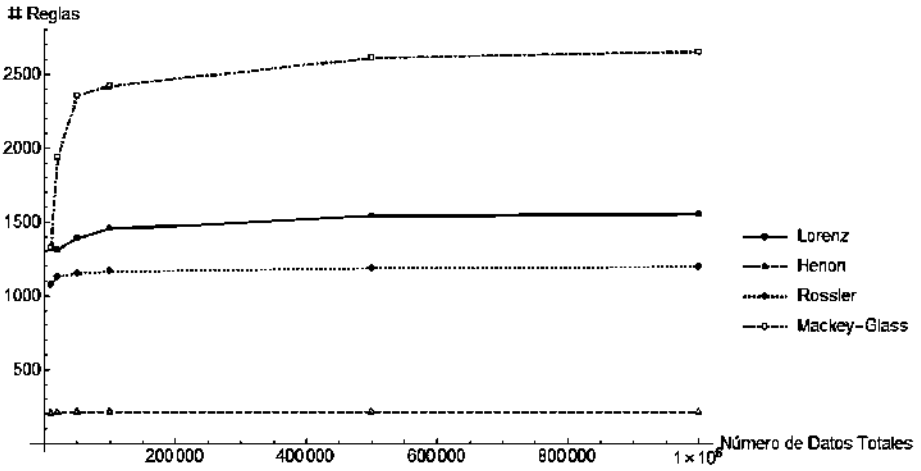
En este capítulo se presentaron diversas prubas sobre el FF, si bien aún hay muchos experimentos que se pueden plantear, los presentados aquí abordan las preguntas más interesantes e inmediatas que pueden surgir cuando se presenta un nuevo modelo. Las pruebas sobre los parámetros del algoritmo resultaron útiles para hacer una especie de calibración y también apreciar las partes que vuelven más robusto al método. Posteriormente en las pruebas comparativas se pudo ver que el modelo tiene un gran desempeño en cuanto al tiempo, no tanto así en la precisión y es una área que se debe trabajar, aún así FF mantiene un compromiso entre precisión y tiempo de aprendizaje. Por último, los experimentos sobre datos masivos sirvieron para ver el potencial de FF ya que podría trabajar en aplicaciones en tiempo real, con grandes cantidades de datos y sin necesidad de reentrenarse.



(a) ***Tiempo de aprendizaje vs. tamaño de la serie de tiempo***



(b) ***MAPE vs. tamaño de la serie de tiempo***



(c) ***Número de reglas generadas vs. tamaño de la serie de tiempo***

Figura 6.7: Medidas de desempeño considerando diferentes números de datos

Capítulo 7

Conclusiones

En este capítulo se mencionan las conclusiones a las que se llegó después del desarrollo del trabajo. Asímismo se mencionan de forma general las posibilidades que deja abiertas esta línea de investigación en particular. Dentro de las conclusiones se mencionan también una serie de observaciones que enriquecen la investigación realizada.

7.1. Conclusiones generales

Desde el inicio de la investigación se planteaba que los modelos existentes para pronóstico siempre tienen sus limitantes. Se observó que aunque existen una cantidad exorbitante de modelos para el pronóstico algunos solo sirven para sistemas lineales, otros no pueden procesar gran cantidad de datos, etcétera. Es decir, cada modelo siempre vendrá acompañado de algunas ventajas y de igual manera desventajas. Lo anterior puede parecer malo, pero solo depende de si existen problemas reales donde se puedan aplicar los modelos. Un modelo es aceptado en base la utilidad que tenga ("Todos los modelos son malos, algunos son útiles". George E. P. Box).

Las primeras conclusiones que se mencionan son a partir de los objetivos específicos que se tenían estipulados desde el inicio de la investigación. Al respecto del objetivo general se puede decir que se cumplió plenamente ya que se presentó un modelo basado en lógica difusa que convierte la información relevante de la serie de tiempo (por medio de los vectores de retardo) en una base de reglas. Del primer objetivo específico se puede decir

que se observaron trabajos de gran interés (especialmente los mapas cognitivos difusos), sin embargo también se corroboró que en ningún trabajo previó se había planteado un modelo puramente difuso y que además se auto-entrena, es decir, construye su base de conocimiento a partir de los datos, esto de manera autonoma. Esto es interesante ya que los sistemas difusos generalmente no contruyen su base de conocimiento más bien se las proporciona un experto.

Tomando en cuenta los primeros objetivos específicos, estos son la base del sistema que se desarrolló e incluso quedaron capturados de alguna manera en los algoritmos de aprendizaje y pronóstico difusos. Con respecto a los objetivos específicos que mencionaban realizar comparaciones entre modelos y aplicados a series de tiempo caóticas, es evidente que se llevaron a cabo estas acciones. Sin embargo, aquí se debe aclarar el porqué de elegir específicamente los modelos ARIMA, NN, NNDE y ANN. El modelo ARIMA es el más completo y utilizado dentro del enfoque clásico, así que resulta buena idea medir el desempeño de cualquier modelo con respecto a éste. La razón de escoger la red neuronal artificial (en inglés Artificial Neural Network) (ANN) fue porque estas trabajan adecuadamente en problemas altamente no lineales, además de que normalmente ofrecen resultados satisfactorios en cuanto a la precisión. Finalmente se eligió NN y su versión modificada, NNDE, ya que son métodos que comúnmente sirven para tratar series de tiempo caóticas.

Como conclusión general se puede decir que el modelo desarrollado cumple con varias de las características deseables en un método de predicción. Entre ellas destaca que el modelo obtenido tiene una precisión aceptable, es rápido en cuanto al tiempo de ejecución del aprendizaje, es un modelo incremental (no requiere reentrenarse), puede trabajar con datos masivos y finalmente también es versátil, ya que se aplicó indistintamente en series de tiempo sintéticas y de velocidad de viento. Por su estructura interna puede trabajar en series de tiempo muy diversas.

7.2. Conclusiones específicas

En base a lo anteriormente dicho y a partir de lo presentado en la sección previa de resultados se pueden obtener varias conclusiones útiles. A manera de resumen éstas se

presentan como conclusiones específicas.

Pruebas sobre los parámetros

A partir de las pruebas hechas sobre los parámetros de entrada de los algoritmos de pronóstico y aprendizaje se observó que al usar todos los conjuntos a los que pertenece cada punto ayuda a crear más reglas, esto para la etapa de aprendizaje. Lo anterior aumenta la posibilidad de que los pronósticos se calculen usando las reglas y no el método NAÏVE y conlleva a mejorar la precisión de las predicciones. Esto también tiene un efecto contraproducente, ya que entre más reglas mayor es el tiempo de aprendizaje. Así que el selector S_{TR} se puede activar (que tome el valor cierto) en casos que se requiera mayor precisión en el pronóstico y no importe tanto el tiempo que tarde en obtenerse.

No se notó mejora alguna al usar todos los conjuntos para fusificar los datos en la etapa de validación. O sea, no hay mucha diferencia en precisión entre $S_{TC} = Cierto$ ó $S_{TC} = Falso$. Al no observar un cambio significativo se puede concluir que este parámetro es innecesario, además hace que el t_p se aumente al doble cuando se elige fusificar con todos los conjuntos. Para mayor seguridad se deben realizar más pruebas en otras series de tiempo.

Con respecto al selector de conjuntos variables S_{CV} se observó que en algunos casos es mejor considerar que los datos se distribuyen normalmente. Esto puede ayudar cuando se quiere trabajar con pocos conjuntos. Al probar con 20 conjuntos difusos para ambos casos no se observa diferencia significativa, sin embargo cuando se usa la distribución uniforme normalmente es necesario agregar una mayor cantidad de conjuntos para mejorar los resultados. Por otro lado, cuando se usa la distribución normal no es necesario usar tantos conjuntos, pronto se tiene una resolución aceptable. Se puede afirmar que cualquiera de los dos opciones puede ser útil, esto depende de la serie de tiempo. Se puede concluir que la distribución normal se usa cuando se piensa que existen datos atípicos y la uniforme en el caso contrario, si se usa la normal cuando no existen datos atípicos los resultados no deben ser muy diferentes con respecto a usar la uniforme.

En cuanto al número de conjuntos difusos, se observó que no necesariamente se obtienen mejores resultados cuando se tienen más conjuntos, pero sí existe un límite mínimo para el cual la precisión decae significativamente. Usar más conjuntos difusos puede mejorar

la predicción, pero forzosamente aumenta el tiempo de aprendizaje y el de pronóstico.

Comparaciones con otros modelos

A partir de las comparaciones del método FF con ARIMA, ANN, NN y NNDE, se puede observar que el método está entre el tercer y cuarto lugar de los cinco (considerando sólo la precisión). El método NNDE en cuanto a precisión es el que tuvo mejores resultados, sin embargo el tiempo que requiere para entrenarse es bastante grande (30 minutos para $100,000$ datos), de igual manera las redes neuronales ocupan para entrenarse un tiempo extremadamente grande (2 horas para $100,000$ datos), lo cual es evidente considerando que se aplica el algoritmo de evolución diferencial para optimizar los parámetros m, τ, ϵ_r (aún cuando se hiciera en un espacio de búsqueda pequeño, los algoritmos evolutivos requieren una gran cantidad de tiempo y recursos). Esto constrasta con el tiempo que requiere FF que en todos los casos es bastante reducido (cerca de 4 minutos para $100,000$ datos). Se puede decir que, el orden de los modelos sin considerar el tiempo de aprendizaje, sólo tomando en cuenta la precisión, para OSA usando SMAPE es: NNDE, ANN, FF,NN y ARIMA. De manera análoga para ODA usando SMAPE es: NNDE, ANN, NN, FF y ARIMA.

En general, se observa que los métodos más precisos son los que requieren una mayor cantidad de tiempo para el aprendizaje (NNDE y ANN). Por el contrario ARIMA es, después de FF el método más rápido, pero también el que tiene un menor rendimiento en la precisión. Los resultados considerando la medida de error MSE son bastante similares, NNDE siempre se coloca en el primer puesto y ARIMA en el último, y FF en tercer lugar.

Cuando se hizó el análisis costo-beneficio se observó que el modelo FF es el que tiene una relación más pequeña de variación del error, respecto a la ganancia en tiempo. Es decir, si se toma en cuenta el tiempo que tarda en obtener sus resultados NNDE (es el método más preciso) en referencia al tiempo que tardan los demás métodos, FF es el que que obtiene un resultado más satisfactorio. Es decir, obtiene el error menor considerando el tiempo que le lleva obtener este resultado. A través de este análisis también se observó que el modelo ANN es el que tiene una relación entre precisión y tiempo de aprendizaje más deficiente. La red neuronal no compensa el tiempo que tarda en aprender con la eficiencia en los resultados que regresa. Se puede concluir que FF es el método con mejor rendimiento

tomando en cuenta (a la par) el tiempo de aprendizaje y la precisión.

Pruebas de datos masivos

En base a los experimentos de datos masivos se pueden concluir dos cosas. FF es bastante adecuado para tratar grandes cantidades de datos. Se procesaron hasta un millón de datos, sin embargo esto no fue un problema para el método; ya que tomando en cuanta la serie que tardó más, se requirieron alrededor de $1,370$ segundos (cerca de 23 minutos) en la tarea de aprendizaje. Lo cual es bastante rápido, ese mismo tiempo le tomaba a NNDE procesar $100,000$ datos (en un equipo con mucha mayor capacidad de procesamiento). Las redes neuronales generalmente no procesan tantos datos y aún cuando pudieran hacerlo se necesitaría una cantidad de tiempo considerable (para $100,000$ tardó 2 horas). Se observó que el tiempo de aprendizaje crece linealmente conforme aumentan la cantidad de datos, esto es congruente con lo presentado en [Flores u. a., 2016a] (donde se determina la complejidad del algoritmo).

Estas pruebas también permitieron observar que el error aumenta cuando se calculan más pronósticos. Por otro lado también se pudo vislumbrar que el error debe ir bajando cuando se aumentan la cantidad de datos para el aprendizaje y se mantiene constante el número de pronósticos. Un resultado interesante de las pruebas con datos masivos es que después de determinada cantidad de datos el número de reglas crece muy poco. Las gráficas muestran que crecen más rapido al inicio y luego se mantienen casi constantes. Esto nos lleva a concluir que después de determinada cantidad de datos los nuevos valores no aportan mucha información adicional.

Estos últimos experimentos muestran el potencial del método FF para procesar grandes cantidades de datos y también se puede concluir que su uso está más orientado en aplicaciones donde el tiempo de aprendizaje (t_a) debe ser relativamente pequeño. Una propiedad adicional con la que cuenta es que puede ser incremental. Lo anterior se afirma considerando la forma en que se guardan las reglas en la base de datos (esto se explicó en el Capítulo 5, Sección 5.2). Se puede obtener la versión difusa de cualquier vector de retardo cuando se necesite y esto a su vez convertirse en una regla. Por lo anterior, mientras se generan pronósticos se podría seguir entrenando el modelo, además al agregar las fortalezas

de las reglas y que éstas se calculen con respecto al valor anterior de ese mismo parámetro. Se puede notar que FF es aún más adecuado cuando se presenten más situaciones que se parecen a determinada regla, está tendrá más posibilidades de ser activada. En cuanto al tiempo de pronóstico (t_p) se puede decir que siempre es bastante bajo, para $1,000,000$ datos requirió máximo 13 segundos (serie de Mackey-Glass).

Un último hecho que se desea comentar es que en algunos casos el número de reglas para más datos fue menor. Esto puede ocurrir porque situaciones nuevas no generen nuevas reglas. La dimensión de embebido (m) influye directamente para la máxima cantidad de reglas que se pueden generar. Como se habia mencionado en la Sección 5.1, el número de reglas máximo está dado por 2^{m+1}.

7.3. Trabajos futuros

A continuación se mencionan los trabajos que se pueden realizar a futuro sobre el método de pronóstico difuso. Las mejoras irían en el sentido de cubrir las características deseables en un sistema de pronóstico. Las cuales se explicaron en el Capítulo 1.

La primera mejora que se propone es en el sentido de obtener mejores resultados en cuanto a la precisión. Se observó que tiene resultados inferiores a NNDE y en algunos casos que ANN o NN, casi siempre sobre ARIMA y posicionandose como tercer lugar. Para aumentar la precisión del método se puede pensar agregar otros métodos de defusificación, cambiar las funciones de intersección y unión, etc., aunque realmente esto ayudaría poco. Se observó que FF tiene problemas en la precisión, especialmente en series con datos atípicos (las series de velocidad de viento para Michoacán), una mejora sustancial que se puede hacer al algoritmo es utilizar conjuntos difusos tipo II.

Lo anterior se sustenta en el hecho de que la lógica difusa usando conjuntos difusos tipo I sirve como un punto de referencia para clasificar términos lingüísticos, sin embargo no puede aportar la dispersión que presenta cada término lingüístico. Los conjuntos difusos tipo II sirven para modelar más fiablemente la incertidumbre lingüística. Por ejemplo, al usar lógica difusa tipo I para clasificar con diferentes etiquetas lingüísticas a una variable, supongase la «temperatura », se podría dividir en ‘baja’ , ‘media’ , ‘alta’ (incluso se pueden

usar más conjuntos) de cualquier forma queda una cuestión sin resolver, como definir los límites de cada conjunto. Normalmente estos se definen a partir del conocimiento de un experto, pero si se considera que dos o más expertos podrían tener opiniones diferentes de como definir los límites, se puede pensar en modelar nuevamente esa incertidumbre entre opiniones mediante lógica difusa; lo anterior es el origen del estudio de la lógica difusa tipo II.

Esto tiene una aplicación directa en las series de tiempo ya que uno de los problemas principales en el algoritmo FF es como definir la distribución de los conjuntos en el rango de la serie de tiempo. Además de resolver este problema también se estima que puede ayudar a aumentar la precisión ya que la base de reglas podría contener de una forma más robusta las situaciones anteriores. Utilizando conjuntos difusos tipo II es más posible que un punto dado coincida con alguna regla de la base de datos.

Como segunda mejora sustancial se puede pensar en un nuevo enfoque de pronóstico. Hasta ahora las reglas tienen múltiples antecedentes y un único consecuente, podrían generarse reglas (y vectores de retardo) con múltiples puntos como consecuentes. En esta forma de pronosticar se evitaría hasta cierto punto hacer predicciones utilizando estimaciones previas (lo cual se hace en el enfoque iterativo).

Teniendo en cuenta que el algoritmo ya es incremental se puede agregar una parte que se encargue de eliminar definitivamente las reglas que puedan quedar obsoletas. Es decir, conforme lleguen nuevas situaciones las que ya no figuran pueden ser suprimidas. Adicionalmente se puede pensar en hacer que los parámetros m y τ se recalculen automáticamente con respecto a los nuevos datos que lleguen. Esto ya que algunos sistemas caóticos pueden tener una representación en el espacio de fase que varía constantemente.

Considerando que NNDE obtiene mejores resultados que NN se puede plantear hacer una versión de FF que optimize sus parámetros por medio de DE ($FFDE$). NNDE es el ganador, pero tiene una ventaja considerable sobre los demás métodos, sus parámetros son optimizados usando optimización por evolución diferencial (en inglés Diferencial Evolution). Para tener una comparación justa, se plantea desarrollar una versión de FF que también optimize sus parámetros por medio de DE.

Un experimento que podría realizarse a corto plazo es probar el método FF en

datos masivos dejando constante el número de pronósticos a obtener. Lo cual daría más certeza al hecho de que con más datos el error debe disminuir. También se puede trabajar en más casos de prueba, ya que el algoritmo hasta ahora solo se ha probado en un conjunto limitado de series caóticas. Fue creado especialmente pensando en este tipo de series pero esto no significa que no se pueda aplicar en otros casos de estudio. En relación con lo anterior y considerando las ventajas que presenta el método (es incremental, trabaja con datos masivos y es rápido) se puede aplicar en series de tiempo económicas y financieras. Principalmente porque en estas aplicaciones se requiere obtener predicciones en un tiempo reducido. En este sentido el tiempo de pronóstico es prácticamente instantáneo, si el tiempo de aprendizaje es bajo, el tiempo de pronóstico es ínfimo. Incluso no afecta tanto la precisión, ya que en algunos casos, como series de tiempo de tipo de cambio y divisas no se requiere un valor preciso sino solo saber su comportamiento general. Es decir, saber si conviene comprar o vender, si va a bajar o subir determinada moneda con respecto a otra, etc.

Cuando se realizaron predicciones en las series de viento de Michoacán se tuvieron tres problemas que afectaron en gran medida los resultados. El primero es que la información almacenada contenía un alto nivel de ruido. Como segundo punto, que existían demasiados datos faltantes y el tercer factor es que había presencia de datos atípicos. Una parte importante que se debe desarrollar es darle un tratamiento adecuado a estos tipos de datos. El sistema FF es hasta cierto punto robusto a datos faltantes, ya que si un vector de retardo no está completo se desprecia, sin embargo esto provoca contar con poca información para crear la base de reglas. El tratamiento de estos tres aspectos sería, quizás, el trabajo futuro más urgente.

Como último punto se aborda la posibilidad de crear los vectores de retardo usando técnicas de reconocimiento de patrones y clasificación a la par con la ya existente que se basa en la reconstrucción del espacio de fase. En este sentido lo más conveniente sería realizar una investigación exhaustiva de técnicas que permitan generar una especie de vectores de retardo a partir de vectores de características.

Referencias

[Acosta 2006] Acosta, Héctor N.: Diseño de controladores dedicados a la lógica difusa. (2006)

[Alligood u. a. 2006] Alligood, Kathleen T. ; Sauer, Tim D. ; Yorke, James A.: *Chaos: an introduction to dynamical systems.* Springer Science Business Media, 2006

[Amjad u. a. 2012] Amjad, Usman ; Jilani, Tahseen A. ; Yasmeen, Farah: A two phase algorithm for fuzzy time series forecasting using genetic algorithm and particle swarm optimization techniques. In: *International Journal of Computer Applications* 55 (2012), Nr. 16

[Arango und Velasquez 2014] Arango, Adriana ; Velasquez, Juan D.: Forecasting the Colombian Exchange Market Index (IGBC) using Neural Networks. In: *IEEE Latin America Transactions* 12 (2014), Nr. 4, S. 718–724

[Axelrod 2015] Axelrod, Robert: *Structure of decision: The cognitive maps of political elites.* Princeton university press, 2015

[Bishop 1995] Bishop, C.M.: *Neural Networks for Pattern Recognition.* Clarendon Press, 1995 (Advanced Texts in Econometrics). – ISBN 9780198538646

[Brockwell und Davis 2013] Brockwell, Peter J. ; Davis, Richard A.: *Time series: theory and methods.* Springer Science and Business Media, 2013

[Castillo 1999] Castillo, Carlos I.: *Lógica y teoría de conjuntos.* 1999

[Chafield 1975] Chafield, C: *The analysis of time series: theory and practice.* 1975

[Chen und Pham 2000] Chen, Guanrong ; Pham, Trung T.: *Introduction to fuzzy sets, fuzzy logic, and fuzzy control systems.* CRC press, 2000

[Chen 1996] Chen, Shyi-Ming: Forecasting enrollments based on fuzzy time series. In: *Fuzzy Sets and Systems* 81 (1996), Nr. 3, S. 311 – 319. – URL http://www.sciencedirect.com/science/article/pii/0165011495002200. – ISSN 0165-0114

[Cheng u. a. 2016] Cheng, Shou-Hsiung ; Chen, Shyi-Ming ; Jian, Wen-Shan: Fuzzy time series forecasting based on fuzzy logical relationships and similarity measures. In: *Information Sciences* 327 (2016), S. 272–287

[Cover und Hart 1967] Cover, Thomas ; Hart, Peter: Nearest neighbor pattern classification. In: *IEEE transactions on information theory* 13 (1967), Nr. 1, S. 21–27

[De La Vega u. a. 2014] De La Vega, Erick ; Flores, Juan J. ; Graff, Mario: K-Nearest-Neighbor by Differential Evolution for Time Series Forecasting. In: *Mexican International Conference on Artificial Intelligence* Springer (Veranst.), 2014, S. 50–60

[Douglas C. Montgomery 2008] Douglas C. Montgomery, Murat K.: *Introduction to Time Series Analysis and Forecasting.* John Wiley and Sons, Inc.,Publication, 2008

[Duda u. a. 2012] Duda, Richard O. ; Hart, Peter E. ; Stork, David G.: *Pattern classification.* John Wiley and Sons, 2012

[Efendi u. a. 2016] Efendi, Riswan ; Deris, Mustafa M. ; Ismail, Zuhaimy: Implementation of fuzzy time series in forecasting of the non-stationary data. In: *International Journal of Computational Intelligence and Applications* 15 (2016), Nr. 02, S. 1650009

[Efendi u. a. 2015] Efendi, Riswan ; Ismail, Zuhaimy ; Sarmin, Nor H. ; Mat Deris, Mustafa: A reversal model of fuzzy time series in regional load forecasting. In: *International Journal of Energy and Statistics* 3 (2015), Nr. 01, S. 1550003

[Efendigil u. a. 2009] Efendigil, Tuğba ; Önüt, Semih ; Kahraman, Cengiz: A decision support system for demand forecasting with artificial neural networks and neuro-fuzzy

models: A comparative analysis. In: *Expert Systems with Applications* 36 (2009), Nr. 3, S. 6697–6707

[Egrioglu u. a. 2011] Egrioglu, Erol ; Aladag, Cagdas H. ; Basaran, Murat A. ; Yolcu, Ufuk ; Uslu, Vedide R.: A new approach based on the optimization of the length of intervals in fuzzy time series. In: *Journal of Intelligent and Fuzzy Systems* 22 (2011), Nr. 1, S. 15–19

[Egrioglu u. a. 2010] Egrioglu, Erol ; Aladag, Cagdas H. ; Yolcu, Ufuk ; Uslu, Vedide R. ; Basaran, Murat A.: Finding an optimal interval length in high order fuzzy time series. In: *Expert Systems with Applications* 37 (2010), Nr. 7, S. 5052–5055

[Firat und Güngör 2008] Firat, Mahmut ; Güngör, Mahmud: Hydrological time-series modelling using an adaptive neuro-fuzzy inference system. In: *Hydrological Processes* 22 (2008), Nr. 13, S. 2122–2132

[Flores u. a. 2016a] Flores, Juan J. ; Calderon, Felix ; Espinosa, Elisa ; Cedeno, Rafael ; Garnica, Adan ; Flores, Georgina: Fuzzy Nearest Neighbor Time Series Forecasting-Computational Complexity. In: *Computational Science and Computational Intelligence (CSCI), 2016 International Conference on* IEEE (Veranst.), 2016, S. 490–495

[Flores u. a. 2016b] Flores, Juan J. ; Calderon, Felix ; Gonzalez, Jose R C. ; Ortiz, Jose ; Farias, Rodrigo L.: Comparison of time series forecasting techniques with respect to tolerance to noise. In: *Power, Electronics and Computing (ROPEC), 2016 IEEE International Autumn Meeting on* IEEE (Veranst.), 2016, S. 1–6

[Flores u. a. 2015a] Flores, Juan J. ; Calderón, Félix ; Lara, José Ortiz C.: Aprendizaje de Modelos Difusos para Predicción de Series de Tiempo. In: *10. Aprendizaje de Modelos Difusos para Predicción de Series de Tiempo* (2015), S. 10

[Flores u. a. 2015b] Flores, Juan J. ; Ortiz, Jose ; González, José R C. ; Lara, Carlos ; Farías, Rodrigo L.: FNN a fuzzy version of the nearest neighbor time series forecasting technique. In: *Power, Electronics and Computing (ROPEC), 2015 IEEE International Autumn Meeting on* IEEE (Veranst.), 2015, S. 1–6

[Fraser und Swinney 1986] Fraser, Andrew M. ; Swinney, Harry L.: Independent coordinates for strange attractors from mutual information. In: *Physical review A* 33 (1986), Nr. 2, S. 1134

[Glass und Mackey 2010] Glass, Leon ; Mackey, Michael: Mackey-glass equation. In: *Scholarpedia* 5 (2010), Nr. 3, S. 6908

[Hénon 1976] Hénon, Michel: A two-dimensional mapping with a strange attractor. In: *The Theory of Chaotic Attractors*. Springer, 1976, S. 94–102

[Homenda u. a. 2014] Homenda, Wladyslaw ; Jastrzebska, Agnieszka ; Pedrycz, Witold: Modeling time series with fuzzy cognitive maps. In: *Fuzzy Systems (FUZZ-IEEE), 2014 IEEE International Conference on* IEEE (Veranst.), 2014, S. 2055–2062

[Huang u. a. 2011] Huang, Yao-Lin ; Horng, Shi-Jinn ; He, Mingxing ; Fan, Pingzhi ; Kao, Tzong-Wann ; Khan, Muhammad K. ; Lai, Jui-Lin ; Kuo, I-Hong: A hybrid forecasting model for enrollments based on aggregated fuzzy time series and particle swarm optimization. In: *Expert Systems with Applications* 38 (2011), Nr. 7, S. 8014 – 8023. – URL `http://www.sciencedirect.com/science/article/pii/S0957417410014909`. – ISSN 0957-4174

[Huarng und Yu 2006] Huarng, Kunhuang ; Yu, Tiffany Hui-Kuang: The application of neural networks to forecast fuzzy time series. In: *Physica A: Statistical Mechanics and its Applications* 363 (2006), Nr. 2, S. 481 – 491. – URL `http://www.sciencedirect.com/science/article/pii/S0378437105008460`. – ISSN 0378-4371

[Ismail u. a. 2015] Ismail, Zuhaimy ; Efendi, Riswan ; Deris, Mustafa M.: Application of fuzzy time series approach in electric load forecasting. In: *New Mathematics and Natural Computation* 11 (2015), Nr. 03, S. 229–248

[Jacques u. a. 2002] Jacques, Maria Alice P. ; Pursula, Matti ; Niittymäki, Jarkko ; Kosonen, Iisakki: The impact of different approximate reasoning methods on fuzzy signal controllers. In: *Proceedings of the 13th Mini-EURO Conference*, 2002, S. 184–192

[Jang 1993] Jang, J-SR: ANFIS: adaptive-network-based fuzzy inference system. In: *IEEE transactions on systems, man, and cybernetics* 23 (1993), Nr. 3, S. 665–685

[Kantz und Schreiber 2004] Kantz, Holger ; Schreiber, Thomas: *Nonlinear time series analysis*. Bd. 7. Cambridge university press, 2004

[Kennel u. a. 1992] Kennel, Matthew B. ; Brown, Reggie ; Abarbanel, Henry D.: Determining embedding dimension for phase-space reconstruction using a geometrical construction. In: *Physical review A* 45 (1992), Nr. 6, S. 3403

[Kim und Kasabov 1999] Kim, J. ; Kasabov, N.: HyFIS: adaptive neuro-fuzzy inference systems and their application to nonlinear dynamical systems. In: *Neural Networks* 12 (1999), Nr. 9, S. 1301 – 1319. – URL `http://www.sciencedirect.com/science/article/pii/S0893608099000672`. – ISSN 0893-6080

[Klir und Yuan 1995] Klir, George ; Yuan, Bo: *Fuzzy sets and fuzzy logic*. Bd. 4. Prentice hall New Jersey, 1995

[Kosko 1986] Kosko, Bart: Fuzzy cognitive maps. In: *International journal of man-machine studies* 24 (1986), Nr. 1, S. 65–75

[Kosko 1992] Kosko, Bart: Neural networks and fuzzy systems: a dynamic systems approach to machine intelligence. In: *Englewood Cliffs, NY Prentice Hall* (1992)

[Kurian u. a. 2006] Kurian, Ciji P. ; George, VI ; Bhat, Jayadev ; Aithal, Radhakrishna S.: ANFIS model for the time series prediction of interior daylight illuminance. In: *International Journal on Artificial Intelligence and Machine Learning* 6 (2006), Nr. 3, S. 35–40

[Lee 1990] Lee, Chuen-Chien: Fuzzy logic in control systems: fuzzy logic controller. I. In: *IEEE Transactions on systems, man, and cybernetics* 20 (1990), Nr. 2, S. 404–418

[Lorenz 1963] Lorenz, Edward N.: Deterministic nonperiodic flow. In: *Journal of the atmospheric sciences* 20 (1963), Nr. 2, S. 130–141

[Mamdani 1976] Mamdani, Ebrahim H.: Advances in the linguistic synthesis of fuzzy controllers. In: *International Journal of Man-Machine Studies* 8 (1976), Nr. 6, S. 669–678

[Mendel 2007] Mendel, Jerry M.: Type-2 fuzzy sets and systems: An overview [corrected reprint]. In: *IEEE Computational Intelligence Magazine* 2 (2007), Nr. 2, S. 20–29

[Moctezuma 2015] Moctezuma, M.B.O.: *Sistemas dinámicos en tiempo continuo: Modelado y simulación.* Omniascience, 2015. – ISBN 9788494467325

[Nauck und Kruse 1998] Nauck, Detlef ; Kruse, Rudolf: A neuro-fuzzy approach to obtain interpretable fuzzy systems for function approximation. In: *Fuzzy Systems Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on* Bd. 2 IEEE (Veranst.), 1998, S. 1106–1111

[Nauck und Kruse 1999] Nauck, Detlef ; Kruse, Rudolf: Neuro-fuzzy systems for function approximation. In: *Fuzzy sets and systems* 101 (1999), Nr. 2, S. 261–271

[Ogata 1996] Ogata, Katsuhiko: *Sistemas de control en tiempo discreto.* Pearson educación, 1996

[Olabe 1998] Olabe, Xabier B.: Redes neuronales artificiales y sus aplicaciones. In: *Publicaciones de la Escuela de Ingenieros* (1998)

[Olivares Caballero 1994] Olivares Caballero, Daniel: *Aplicación de sistemas caóticos en control automático*, Universidad Autónoma de Nuevo León, Dissertation, 1994

[Orchard 2004] Orchard, Bob: FuzzyCLIPS Version 6.10 d User's Guide. In: *National Research Council of Canada* (2004)

[Park u. a. 1995] Park, Young-Moon ; Moon, Un-Chul ; Lee, Kwang Y.: A self-organizing fuzzy logic controller for dynamic systems using a fuzzy auto-regressive moving average (FARMA) model. In: *IEEE Transactions on Fuzzy systems* 3 (1995), Nr. 1, S. 75–82

[Pedrycz u. a. 2016] Pedrycz, Witold ; Jastrzebska, Agnieszka ; Homenda, Wladyslaw: Design of fuzzy cognitive maps for modeling time series. In: *IEEE Transactions on Fuzzy Systems* 24 (2016), Nr. 1, S. 120–130

[Peitgen u. a. 2006] Peitgen, Heinz-Otto ; Jürgens, Hartmut ; Saupe, Dietmar: *Chaos and fractals: new frontiers of science*. Springer Science Business Media, 2006

[Perez 2010] Perez, Rigoberto: *Nociones básicas de Estadística*. Universidad de Oviedo, 2010

[Popov und Bykhanov 2005] Popov, Alexander A. ; Bykhanov, Kolya V.: Modeling volatility of time series using fuzzy GARCH models. In: *Science and Technology, 2005. KORUS 2005. Proceedings. The 9th Russian-Korean International Symposium on* IEEE (Veranst.), 2005, S. 687–692

[Riid und Rüstern 1998] Riid, Andri ; Rüstern, Ennu: Comparison of fuzzy function approximators. In: *Proc. 6th Biennal Baltic Electronic Conference*, 1998, S. 139–142

[Robert H. Shumway 2011] Robert H. Shumway, David S. S.: *Time Series Analysis and Its Applications*. Springer, 2011

[Ross 2009] Ross, Timothy J.: *Fuzzy logic with engineering applications*. John Wiley and Sons, 2009

[Rössler 1976] Rössler, Otto E.: An equation for continuous chaos. In: *Physics Letters A* 57 (1976), Nr. 5, S. 397–398

[Sachdev und Sharma 2015] Sachdev, Ajeeta ; Sharma, Vivek: Stock Forecasting Model Based on Combined Fuzzy Time Series and Genetic Algorithm. In: *Computational Intelligence and Communication Networks (CICN), 2015 International Conference on* IEEE (Veranst.), 2015, S. 1303–1307

[Schweizer und Sklar 2011] Schweizer, Berthold ; Sklar, Abe: *Probabilistic metric spaces*. Courier Corporation, 2011

[Sfetsos 2000] Sfetsos, Athanasios: A comparison of various forecasting techniques applied to mean hourly wind speed time series. In: *Renewable energy* 21 (2000), Nr. 1, S. 23–35

[Singh 1998] Singh, Sameer: Fuzzy nearest neighbour method for time-series forecasting. In: *Proc. 6th European Congress on Intelligent Techniques and Soft Computing* Citeseer (Veranst.), 1998

[Song u. a. 2010a] Song, Hengjie ; Miao, Chunyan ; Roel, Wuyts ; Shen, Zhiqi ; Catthoor, Francky: Implementation of fuzzy cognitive maps based on fuzzy neural network and application in prediction of time series. In: *IEEE Transactions on Fuzzy Systems* 18 (2010), Nr. 2, S. 233–250

[Song u. a. 2010b] Song, HJ ; Miao, CY ; Shen, ZQ ; Roel, W ; Maja, DH ; Francky, C: Design of fuzzy cognitive maps using neural networks for predicting chaotic time series. In: *Neural Networks* 23 (2010), Nr. 10, S. 1264–1275

[Song und Chissom 1993a] Song, Qiang ; Chissom, Brad S.: Forecasting enrollments with fuzzy time series – Part I. In: *Fuzzy Sets and Systems* 54 (1993), Nr. 1, S. 1 – 9. – URL `http://www.sciencedirect.com/science/article/pii/016501149390355L`. – ISSN 0165-0114

[Song und Chissom 1993b] Song, Qiang ; Chissom, Brad S.: Fuzzy time series and its models. In: *Fuzzy sets and systems* 54 (1993), Nr. 3, S. 269–277

[Song und Chissom 1994] Song, Qiang ; Chissom, Brad S.: Forecasting enrollments with fuzzy time series – Part II. In: *Fuzzy Sets and Systems* 62 (1994), Nr. 1, S. 1 – 8. – URL `http://www.sciencedirect.com/science/article/pii/0165011494900671`. – ISSN 0165-0114

[Stach u. a. 2008] Stach, Wojciech ; Kurgan, Lukasz A. ; Pedrycz, Witold: Numerical and linguistic prediction of time series with the use of fuzzy cognitive maps. In: *IEEE Transactions on Fuzzy Systems* 16 (2008), Nr. 1, S. 61–72

[Sulandari und Yudhanto 2015] Sulandari, Winita ; Yudhanto, Yudho: Forecasting trend data using a hybrid simple moving average-weighted fuzzy time series model. In: *Science in Information Technology (ICSITech), 2015 International Conference on* IEEE (Veranst.), 2015, S. 303–308

[Sánchez u. a. 2007] Sánchez, José Manuel B. ; Lugilde, Diego N. ; Linares Fernández, Concepción de ; Guardia, Consuelo D. de la ; Sánchez, Francisca A. u. a.: Forecasting airborne pollen concentration time series with neural and neuro-fuzzy models. In: *Expert Systems with Applications* 32 (2007), Nr. 4, S. 1218–1225

[Toolbox 1995–2015] Toolbox, Fuzzy L.: *User's Guide© Copyright 1995–2015, The Math Works*. 1995–2015

[Tseng und Tzeng 2002] Tseng, Fang-Mei ; Tzeng, Gwo-Hshiung: A fuzzy seasonal ARIMA model for forecasting. In: *Fuzzy Sets and Systems* 126 (2002), Nr. 3, S. 367–376

[Tseng u. a. 2001] Tseng, Fang-Mei ; Tzeng, Gwo-Hshiung ; Yu, Hsiao-Cheng ; Yuan, Benjamin J.: Fuzzy ARIMA model for forecasting the foreign exchange market. In: *Fuzzy sets and systems* 118 (2001), Nr. 1, S. 9–19

[Walpole 2012] Walpole, Ronald E.: *Probability statistics for engineers scientists*. 2012

[Wolf u. a. 1985] Wolf, Alan ; Swift, Jack B. ; Swinney, Harry L. ; Vastano, John A.: Determining Lyapunov exponents from a time series. In: *Physica D: Nonlinear Phenomena* 16 (1985), Nr. 3, S. 285–317

[Xing u. a. 2017] Xing, Frank Z. ; Cambria, Erik ; Zou, Xiaomei: Predicting evolving chaotic time series with fuzzy neural networks. In: *International Joint Conference on Neural Networks (IJCNN)*, 2017

[Yadav und Balakrishnan 2014] Yadav, Rajnish K. ; Balakrishnan, Manoj: Comparative evaluation of ARIMA and ANFIS for modeling of wireless network traffic time series. In: *EURASIP Journal on Wireless Communications and Networking* 2014 (2014), Nr. 1, S. 1–8

[Yang und Huang 1998] Yang, Hong-Tzer ; Huang, Chao-Ming: A new short-term load forecasting approach using self-organizing fuzzy ARMAX models. In: *IEEE Transactions on Power Systems* 13 (1998), Nr. 1, S. 217–225

[Yu 2005] Yu, Hui-Kuang: Weighted fuzzy time series models for {TAIEX} forecasting. In: *Physica A: Statistical Mechanics and its Applications* 349 (2005), Nr. 3–4, S. 609 – 624.

– URL `http://www.sciencedirect.com/science/article/pii/S0378437104014128`.

– ISSN 0378-4371

[Zadeh 1996] Zadeh, Lotfi A.: Fuzzy logic= computing with words. In: *IEEE transactions on fuzzy systems* 4 (1996), Nr. 2, S. 103–111