
”4. La falsedad de un juicio no puede servirnos de objeción contra el mismo. La cuestión es saber cuánto ayuda tal juicio para favorecer y conservar la vida, la especie y todo lo necesario a su evolución.”

Friederich Nietzsche (1844-1900).

”Más allá del bien y del mal”.

Capítulo primero: Prejuicios de los filósofos.

Índice general

1. Introducción	1
1.1. Objetivos	2
1.2. Justificación	2
1.3. Restricciones	4
1.4. Descripción del documento	4
2. Antecedentes	7
2.1. Introducción	7
2.2. Inteligencia emergente	8
2.3. Sistemas Multiagentes	8
2.4. Razonamiento cualitativo	10
2.5. Computación genética	11
2.5.1. Programación genética	12
2.6. Sistemas dinámicos	13
2.7. Métodos de inteligencia artificial para predicción de series de tiempo	15
2.8. Casos de estudio	18
2.8.1. Sistema de Lorenz	19
2.8.2. Sistema de Mackey-Glass	19
2.8.3. Mercado de divisas	20
3. Lenguaje difuso dinámico	25
3.1. Conjuntos difusos dinámicos	26
3.2. Variación del valor modal de los conjuntos difusos	27
3.3. Variación en el ancho de banda de los conjuntos difusos	32
3.4. Estabilidad local	33
3.4.1. Semejanza/distancia suave de Gower	38

ÍNDICE GENERAL

3.5. Predicción de series de tiempo mediante conjuntos difusos dinámicos	39
3.5.1. Método de predicción difusa basado en series de tiempo	40
3.5.2. Estrategias	41
3.5.3. Resultados	42
4. Plataforma multiagente	45
4.1. Marco organizacional	45
4.1.1. Grupo de modelado	46
4.1.2. Grupo de coordinación	50
4.1.3. Grupo de interfaz	51
4.1.4. Modelado multiagente	53
4.2. Agregación difusa para la predicción colectiva	53
4.2.1. Métodos de fusión de datos	54
4.2.2. Método híbrido de agregación	54
4.2.3. Efecto de filtrado en la agregación difusa	56
5. Técnicas de razonamiento	61
5.1. Razonamiento difuso basado en series de tiempo	61
5.2. Redes neuronales <i>feedforward</i>	64
5.3. Redes neuronales <i>feedforward</i> difusas	65
5.4. Comparación entre redes FFNN cuantitativas y cualitativas	68
6. Programación genética de sistemas multiagente híbridos	71
6.1. Conceptos básicos	71
6.1.1. Diversidad coevolutiva	72
6.1.2. Comportamiento conveniente y evaluación colectiva de aptitud	73
6.1.3. Evolución filogenética y evolución ontogenética	74
6.2. Modelado evolutivo	75
6.2.1. Inicialización del proceso de PG	76
6.2.2. Entrenamiento genético	79
6.2.3. Reentrenamiento en línea	83
6.3. Predictores especializados	83
7. Resultados y discusión	85
7.1. Caso 1: Sistema de Lorenz	86

ÍNDICE GENERAL

7.2. Caso 2: Sistema de Mackey-Glass	96
7.3. Caso 1: Mercado de divisas	105
7.4. Discusión	118
8. Conclusiones, aportaciones y trabajo futuro	121
8.1. Aportaciones	123
8.2. Publicaciones y trabajos relacionados	126
8.3. Trabajo futuros	128

Capítulo 2

Antecedentes

2.1. Introducción

Una pretensión compartida por las diferentes técnicas de inteligencia artificial (IA), desde las redes neuronales artificiales (RNA) y los sistemas expertos (SE), hasta los algoritmos genéticos (AG) y los sistemas multiagente (SMA), consiste en el desarrollo de sistemas inteligentes artificiales mediante diversas analogías con la naturaleza. Sin embargo, aunque se ha hecho mucho trabajo en esta dirección, por lo general las diversas técnicas de IA se presentan como opciones independientes y en muchos casos contrapuestas para la solución de un mismo problema. Existen diversos intentos por integrar algunas de estas técnicas en aproximaciones híbridas, destacándose en estos trabajos el uso de redes neuronales artificiales, algoritmos genéticos, sistemas multiagente y lógica difusa.

La metodología que se presenta en este trabajo emplea una plataforma multiagente para la predicción de sistemas dinámicos, así como un algoritmo especializado de programación genética para configurar el sistema multiagente. Esta plataforma multiagente permite el empleo de diferentes técnicas de predicción para la simulación de un sistema dinámico. Las técnicas de predicción empleadas en este trabajo provienen del campo de la inteligencia artificial, sin embargo, es posible utilizar también otras técnicas más tradicionales. Se pone énfasis especial en las técnicas de predicción de sistemas descritos mediante series de tiempo. Finalmente, el lenguaje utilizado para describir los datos, así como para integrar las predicciones individuales se basa en una extensión del concepto de conjunto difuso. Dados estos elementos, la metodología presentada se enmarca dentro del campo de *soft computing*.

En este capítulo se presenta un *estado del arte* de técnicas de inteligencia artificial relacionadas con la metodología propuesta, así como una discusión de los casos de estudio que se abordarán.

2.2. Inteligencia emergente

La emergencia de nuevas propiedades en sistemas formados por una colección de individuos, a partir de propiedades más simples, es un concepto muy extendido en las ciencias físicas. En el campo de la inteligencia artificial, esta idea ha conducido a la llamada *inteligencia artificial distribuida*.

La inteligencia artificial distribuida (IAD) es un subcampo de la IA, originalmente enfocado a aplicar técnicas de IA a la solución de problemas del área de cómputo distribuido [Stone and Veloso, 2000]. En un enfoque más general, la IAD aborda la solución colaborativa de problemas globales mediante un grupo de entidades distribuidas, incluyendo aproximaciones como las redes neuronales, los algoritmos genéticos, los sistemas de vida artificial y los sistemas multiagente.

En las diversas aproximaciones de IAD, la inteligencia se considera como una consecuencia de la acción conjunta de individuos semiautónomos muy simples (con respecto al comportamiento inteligente que muestran en conjunto). La inteligencia resulta entonces como una propiedad emergente de la interacción de los individuos. Esto es, el resultado no es una suma (mecanicista) de las habilidades individuales, sino una propiedad nueva que surge, en parte de dicha suma y en parte de la interacción misma entre tales individuos.

En la metodología de simulación desarrollada en la presente investigación, la predicción de un sistema dinámico complejo determinado resulta de la colaboración de un conjunto de agentes dotados con modelos de predicción simples.

2.3. Sistemas Multiagentes

La *teoría de sistemas multiagente* (MAS), constituye una de las nuevas aproximaciones en inteligencia artificial y una de sus áreas de investigación y desarrollo con mayor actividad.

No obstante la popularidad del término *agente* en la literatura técnica y de divulgación en computación, no existe una acuerdo general acerca de la definición precisa de agente. Algunos diccionarios definen de manera genérica a un agente como *"uno que actúa o tiene el poder o la autoridad para actuar... o representar a otro"*¹. Una definición alternativa, más general es: *"Lo que actúa o ejerce acción por oposición a paciente, que la sufre"* [Merani, 1976]. En esta definición, se enfatiza el carácter proactivo y autónomo de los agentes.

En el contexto de la inteligencia artificial, las definiciones oscilan entre las dos defini-

¹Tomada del *American Heritage Dictionary* y citada en [Bradshaw, 1997].

ciones anteriores. Así, por ejemplo, Ndumu y Nwana definen un agente como ”*un programa autocontenido capaz de controlar su propia toma de decisiones y actuar, basado en su percepción de su ambiente, en busca de uno o más objetivos*” [Ndumu and Nwana, 1997]. Demezeau y Müller, por su parte, definen un agente como ”*una entidad inteligente, actuando razonablemente con respecto a sus propios objetivos y al estado actual de su conocimiento*” [Demezeau and Müller, 1990]. Beer define un *agente autónomo* como cualquier sistema integrado, diseñado para satisfacer objetivos internos o externos por sus propias acciones, mientras interactúa de manera continua a largo plazo con el ambiente en el que está situado [Beer, 1996].

Ante la ausencia de una definición precisa de agente, se acostumbra definir este concepto en términos de ciertas propiedades que debiera poseer un sistema para ser considerado como un agente inteligente. Sin embargo, tampoco existe un acuerdo entre los diferentes autores, acerca de las características concretas de los agentes. Sin embargo, se han venido identificando dos grupos de características principales. La primera de ellas, la menos exigente, se conoce como la *definición blanda* de agente e incluye características con las cuales la mayoría de los investigadores están de acuerdo. Por su parte, la segunda definición incluye características adicionales, por lo cual se conoce como la *definición dura* de agente [Jennings and Wooldridge, 1995b].

La definición blanda describe al agente como un sistema con las siguientes características [Jennings and Wooldridge, 1995a, Jennings and Wooldridge, 1995b, Jennings et al., 1998]:

- Habilidad social.
- Reactividad.
- Orientación a objetivos.
- Autonomía.
- Proactividad.
- Continuidad temporal.

La definición dura de agente, exige además que el agente presente algunas de las siguientes propiedades [Jennings and Wooldridge, 1995b]:

- Movilidad.

- Benevolencia.
- Racionalidad.
- Adaptabilidad.
- Colaboración.

En un sistema multiagente se destacan las siguientes características [Jennings, 1996, Moulin and Chaib-Draa, 1996, Müller, 1996, Durfee, 1996, Nwana et al., 1996]:

- Organización y estructura.
- Coordinación.
- Cooperación.
- Negociación.
- Planificación.
- Comunicación e interacción.

2.4. Razonamiento cualitativo

La mayor parte de los esfuerzos por dotar de capacidad predictiva a los agentes se han basado en el uso de la inducción lógica como mecanismo de razonamiento. Las técnicas de *razonamiento cualitativo* son una alternativa a las herramientas tradicionales de modelado y simulación en situaciones en que la información es imprecisa o incompleta. Existen diversas aproximaciones alrededor de lo que globalmente se conoce como modelado cualitativo y/o simulación cualitativa. Las principales aproximaciones de razonamiento cualitativo son las redes neuronales y la lógica difusa.

Se entiende por redes neuronales artificiales (RNA), un conjunto de técnicas de razonamiento cualitativo basadas en analogías con el funcionamiento del cerebro. De acuerdo con esta metáfora, la inteligencia es un resultado emergente de la acción de un gran número de elementos de procesamiento simples interconectados [Fu, 1994].

Las redes neuronales son capaces de realizar generalización, abstracción y extracción de características estadísticas a partir de los datos, aún cuando éstos sean imprecisos y/o incompletos [Fu, 1994]. Otra importante característica de las redes neuronales es una alta

tolerancia a fallas, la cual se basa en la distribución y en la redundancia de su representación de conocimiento [Fu, 1994]. Las redes neuronales son consideradas como una de las formas más adecuadas para simular el razonamiento natural.

La *lógica difusa* fue propuesta con el objeto de proporcionar una herramienta para modelado de fenómenos complejos en base a una representación que permitiera una cierta cantidad de ambigüedad [Yager and Filev, 1994]. Mediante el uso de conjuntos difusos se pueden desarrollar modelos con un mayor poder expresivo que el ofrecido por las representaciones convencionales. Los modelos difusos se pueden clasificar básicamente en dos categorías. La primera de ellas consiste en los Modelos Lingüísticos, en los que la información se codifica de manera difusa y se procesa mediante un conjunto de reglas IF-THEN difusas. La segunda categoría está constituida por modelos que se basan en el método de razonamiento de Takagi-Sugeno-Kang, en el que se combina la capacidad de los modelos lingüísticos con mecanismos para integrar información cuantitativa [Yager and Filev, 1994].

2.5. Computación genética

La importancia de la evolución como una manera de mejorar las expectativas de éxito de una especie es una idea ampliamente aceptada en biología. En el área de la inteligencia artificial, la idea de aplicar los conceptos de evolución a un sistema artificial condujo a la llamada *computación evolutiva*.

La computación evolutiva engloba distintas aproximaciones, basadas en analogías de los procesos evolutivos de los seres vivos, para la solución de problemas de búsqueda y optimización. Las principales líneas de investigación en esta área son: los algoritmos genéticos, las estrategias de evolución y la programación evolutiva (ver [Fogel, 1995], por ejemplo, para una discusión de estas técnicas). Estas aproximaciones abordan el proceso de búsqueda en base a un procedimiento iterativo en el que poblaciones de posibles soluciones son seleccionadas y modificadas aleatoriamente. Las diferencias principales entre los diferentes métodos residen en la elección de las formas de representación, en los procedimientos para generar nuevas poblaciones y en la forma de elegir qué soluciones mantener en futuras generaciones [Fogel, 1999].

Aunque las tres líneas de investigación en computación evolutiva se mantienen actualmente activas, son los algoritmos genéticos los que han despertado mayor interés en la comunidad de inteligencia artificial.

Haciendo una analogía directa del comportamiento natural, los algoritmos genéticos tra-

bajan con una población de individuos, cada uno de los cuales representa una posible solución a un problema dado. Cada individuo tiene asignada una *evaluación de aptitud*, que describe la calidad del individuo como solución para dicho problema. Los individuos con altas aptitudes reciben mayores oportunidades para *reproducirse* mediante la "cruza" con otros individuos de la población. Esto produce nuevos individuos que comparten algunas características tomadas de cada parente. Así se produce una población nueva de posibles soluciones, al seleccionar a los mejores individuos de la generación *actual* y aparearlos para producir un nuevo conjunto de individuos. Al favorecer el apareamiento de los individuos más aptos, es posible explorar las regiones de búsqueda más prometedoras sin tener que realizar una búsqueda exhaustiva. Si el algoritmo genético ha sido bien diseñado, la población deberá converger a una solución óptima para el problema.

2.5.1. Programación genética

La *programación genética* (PG) es una variante de los algoritmos genéticos en la que los individuos en el espacio de búsqueda son programas de computadora [Koza, 1998]. El objetivo de la PG es la evolución de una población de programas de computadora en base al mapeo que hacen de un conjunto de vectores de entrada.

La primera etapa en un proceso de PG consiste en generar la población inicial. Los principales parámetros en esta etapa son: el número máximo de individuos, el conjunto de elementos básicos (terminales y funciones) a utilizar en los programas, y el tamaño máximo permitido a un programa. Una vez determinados estos parámetros se generan los individuos de manera aleatoria.

Los individuos de la población inicial tienen, por lo general, una aptitud muy baja, lo cual tratará de mejorarse paulatinamente mediante un proceso de selección de los mejores individuos y la búsqueda constante de mejores soluciones. Los principales operadores utilizados por la PG para la evolución de la población son [Banzhaf et al., 1998]:

- **Cruza.** Mediante este operador se combina el contenido genético de dos padres, intercambiando partes de sus códigos.
- **Mutación.** Este operador se aplica de manera individual, y su objetivo es producir un cambio aleatorio en el contenido genético del individuo sobre el que opera.
- **Clonación.** El proceso realizado por este operador consiste en seleccionar y reproducir un individuo de una población en la siguiente.

Uno de los elementos más importantes de la PG es su algoritmo de *selección basada en la aptitud*. La aptitud es una medida de la calidad de un programa para predecir las salidas correspondientes a un conjunto de entradas dado [Banzhaf et al., 1998].

Una elección común para la función de aptitud se basa en el uso del llamado *error cuadrático medio* entre la salida predicha por el programa y la salida asociada al mismo ejemplo.

Dadas las medidas de aptitud de los individuos de una población, es necesario decidir que individuos serán utilizados como base de la evolución y que operadores genéticos les serán aplicados. A esta tarea se le denomina selección y la elección del algoritmo para realizarla es uno de los puntos críticos de una aplicación de PG. Uno de los métodos de selección más utilizados recientemente es la *selección por torneo*. En este método se selecciona al azar un subconjunto de la población completa y se realiza un torneo entre sus individuos. Las características de los mejores individuos en el torneo son conservadas (mediante alguna combinación de las diferentes operaciones genéticas), reemplazando las características de los individuos con menor aptitud [Banzhaf et al., 1998].

Un problema central en el desarrollo de sistemas multiagente es el mecanismo utilizado para establecer las reglas de interacción entre los agentes. Si el sistema es pequeño, las conexiones pueden ser establecidas por el programador durante el diseño del sistema. Sin embargo, cuando la complejidad del sistema crece, la automatización en la asignación de reglas de interacción entre agentes cobra una gran importancia. La necesidad de desarrollar comportamiento complejo en SMA por medios evolutivos ha sido señalada por diversos autores (ver por ejemplo [Iba, 1999] o [Zhang, 1999]).

Una de las aproximaciones utilizadas para desarrollar sistemas multiagente de manera evolutiva es la PG. En la mayoría de estos trabajos el objetivo ha sido el desarrollar comportamiento colectivo emergente como una combinación directa de acciones primitivas [Zhang, 1999]. En la metodología que se propone en el presente trabajo se empleará un algoritmo de PG para desarrollar los modelos de razonamiento que serán empleados por los agentes y las interdependencias entre tales modelos.

2.6. Sistemas dinámicos

Los sistemas dinámicos son sistemas que se caracterizan fundamentalmente por los cambios que experimentan a lo largo del tiempo, siendo el principal tema de investigación en esta área el de los *sistemas dinámicos no lineales*. En el contexto de la dinámica no lineal, el

termino de sistema dinámico generalmente se refiere a la representación abstracta del sistema en estudio.

Los métodos disponibles para el modelado y simulación de sistemas dinámicos son muy variados, dependiendo de la complejidad del sistema en estudio. La complejidad de un sistema dinámico está determinada por diversos factores, como son la no linealidad de su descripción, el número generalmente grande de variables involucradas y la información disponible acerca del sistema, que en muchos casos es incompleta y/o imprecisa. En términos de estas características se pueden distinguir, de una manera general, dos tipos de sistemas dinámicos complejos [Duch and Jankowski, 1994]: por un lado se encuentran aquellos sistemas en los que la dinámica es conocida y simple, pero el comportamiento resultante es complejo y desconocido. Tales sistemas se han estudiado ampliamente, particularmente mediante la llamada *teoría del caos*. Por otro lado se encuentran aquellos sistemas cuya dinámica es desconocida y cuyo comportamiento es sólo parcialmente conocido. Estos sistemas han sido menos estudiados y el objetivo de su análisis consiste en obtener descripciones simples del sistema y de ser posible encontrar su dinámica.

En ciertos casos se carece incluso de información acerca de las dependencias entre las diversas variables que intervienen en el problema y toda la información con que se cuenta se limita a un conjunto de observaciones presentada de manera tabular y a lo cual se denomina como una *serie de tiempo*. El proceso usual de modelado en tales casos consiste en mapear estas series de tiempo en un modelo, típicamente continuo, el cual se espera que dé mayor información que los mismos datos [Morrison, 1991]. Son estos últimos sistemas, denominados *débilmente estructurados*, cuya predicción se pretende abordar con la metodología propuesta en el presente trabajo.

Otro elemento importante que debe considerarse al abordar el análisis de un sistema dinámico es su carácter lineal o no lineal. Los sistemas dinámicos más interesantes son los sistemas no lineales, toda vez que éste es el modo natural de comportamiento de los sistemas físicos (reales) [Morrison, 1991]. En algunos casos, sin embargo, los sistemas en estudio (no lineales por naturaleza), pueden analizarse con toda la precisión requerida mediante modelos lineales aproximados.

La metodología que se propone aquí hará uso de técnicas de modelado cualitativo para atacar problemas en los que se carece de un modelo formal como descripción del sistema físico.

2.7. Métodos de inteligencia artificial para predicción de series de tiempo

Una serie de tiempo es un conjunto de datos medidos secuencialmente a lo largo del tiempo [Box et al., 1994], que describe el comportamiento de un sistema dinámico dado. Estas medidas se toman generalmente a intervalos de tiempo igualmente espaciados y con un método consistente de medición. Tal colección de datos representa una fuente de información importante sobre el comportamiento de los sistemas dinámicos. Su importancia reside en el hecho de que es una manera disponible para estudiar cualquier sistema observable, incluso aquellos que carecen de una descripción formal.

Una base de datos de series de tiempo es una base de datos cuyos registros tienen atributos asociados a una dimensión temporal [Last et al., 2001]. Una clase especial de bases de datos de series de tiempo se obtiene cuando un conjunto de series de tiempo simples se encuentran acopladas, de tal manera que los registros de la base de datos consisten en las medidas de diversas variables, tomadas simultáneamente en el tiempo.

La predicción de series de tiempo consiste en hacer un pronóstico de los valores futuros de una variable de interés, basándose en sus valores previos, disponibles en la serie de tiempo. Los llamados procesos de Box-Jenkins son los métodos más extensamente usados para el pronóstico de series de tiempo. Estos métodos se basan en conceptos estadísticos y pueden modelar el comportamiento de una amplia gama de series de tiempo. Pero estos métodos también exhiben importantes limitaciones para el análisis de datos provenientes de sistemas complejos reales, sobresaliendo las siguientes: 1) son de naturaleza lineal o lineal por trozos, mientras que la mayoría de los sistemas complejos son de carácter no lineal; 2), no pueden manejar correctamente los datos de series de tiempo no estacionarias [Kim and Lee, 1999]; 3) requieren datos tomados en períodos igualmente espaciados; 4) los datos se deben editar para tratar con valores extremos o faltantes, así como de otras distorsiones de los datos.

Las técnicas de IA han ganado aceptación recientemente para el pronóstico de series de tiempo, sobresaliendo la lógica difusa, las redes neuronales y los algoritmos genéticos.

Diversas técnicas de inteligencia artificial han sido ampliamente usadas para la predicción de series de tiempo. Destaca, en estos trabajos, el uso de técnicas basadas en redes neuronales. Las redes neuronales han mostrado tener habilidad para el reconocimiento de patrones en series de tiempo no lineales. El modelo de red neuronal más utilizado para la predicción de series de tiempo es la red neuronal *feedforward*, entrenada mediante el algoritmo *back-propagation* [Yao and Poh, 1995]. Si se selecciona como entrada a esta red, un vector formado

por elementos de la serie de tiempo, entonces la red neuronal puede considerarse como una generalización no lineal del modelo AR.

El primer trabajo importante de aplicación de las RNA al análisis de series de tiempo reales fue el desarrollado por Weigend et. al. [Weigend et al., 1991]. En dicho trabajo, se utiliza una red neuronal *feedforward*, entrenada mediante el algoritmo *back-propagation* y en la que se ha incorporado un mecanismo de penalización a la complejidad de la red, con el fin de reducir el problema de sobreentrenamiento. Con esta estructura, los autores abordan la predicción de dos series de tiempo: el número de manchas solares y el mercado de divisas. Para el caso del mercado de divisas, los autores se limitan a predecir valores para una divisa determinada, en un día específico (esto debido a los fuertes cambios entre diversos días de la semana). Específicamente, la RNA se diseña para predecir cambios en los valores del Marco Alemán los días martes, a partir de la serie de tiempo para esa divisa y el valor alcanzado el lunes previo. Los resultados de este trabajo son, sin embargo, bastante limitados, al predecir una sola variable en un día de la semana determinado (la predicción de las 5 divisas principales y durante los 5 días de la semana implicaría un sistema de 25 RNA). Por otra parte, no se establece ninguna relación de causalidad ni siquiera entre las variables consideradas, esto es entre las 5 divisas.

Yao y Poh, por su parte, emplean un método basado en una red neuronal *feed-forward*, entrenada con el algoritmo *backpropagation*, para analizar el índice Kuala Lumpur Stock Exchange Composite Index (KLSE), del mercado Malasio [Yao and Poh, 1995]. Como entrada a la red emplean las series de tiempo del índice KLSE y una serie de indicadores técnicos, mientras que la salida de la red neuronal es el índice KLSE. Aunque los autores no reportan los niveles de error en la predicción, destacan que la eficacia del método, para predecir las tendencias en el mercado de valores, se restringe a una ventana de tiempo muy estrecha, cercana a los datos de entrenamiento. En otro trabajo [Yao et al., 1996], los investigadores abordan la predicción de las 5 divisas abordadas por Weigand et. al. En este trabajo se emplea una red neuronal feed-forward, alimentada con indicadores técnicos, para predecir la tendencia de las divisas, arrojando resultados que van del 46.59 % de éxito para el Yen, hasta el 76.14 % para el Dólar Australiano.

La razón para predecir el valor de una divisa sólo un día de la semana (como fue reportado en los trabajos citados anteriormente) es que la dinámica del mercado sufre variaciones muy grandes a lo largo de la semana. En el presente trabajo se aborda la predicción simultánea de las cinco divisas los 5 días de la semana.

Papageorgiou utilizó un modelo de Markov para encontrar regularidades en ciertas re-

giones locales de la serie de tiempo para el tipo de cambio del franco suizo con respecto al dólar estadounidense, con la intención de predecir la dirección de cambio de los datos [Papageorgiou, 1997]. En esta aproximación, cuando se agregan más datos a la base de conocimiento, es necesario redefinir las regiones locales. En el presente trabajo, esta tarea es innecesaria, ya que todos los datos se analizan de manera uniforme.

Kutsurelis aborda la predicción de la tendencia del mercado de valores usando redes neuronales *feedforward* entrenadas con algoritmo *backpropagation* [Kutsurelis, 1998]. El error reportado se encuentra entre el 5 % y 30 %. Sin embargo, el autor destaca, como dificultad principal para aplicar el método, la elección apropiada del conjuntos de entrenamiento y de los modelos que se utilizarán.

Una arquitectura más sofisticada se obtiene al retroalimentar algunas de las salidas, en lo que se llama una *red neuronal recurrente*. Entre los modelos de redes neuronales recurrentes más utilizados en la predicción financiera sobresalen las redes de Elman. Este tipo de redes consiste básicamente en una red *feed-forward* en la que se ha integrado un lazo de retroalimentación para tomar en consideración la dependencia temporal (ver [Elman, 1990] y [Chappelier and Grumbach, 1994], para una descripción de ésta y otras redes recurrentes y [Dorffner, 1996] para su aplicación al análisis de series de tiempo).

Lee-Giles y colaboradores utilizan una red neuronal mixta para predecir la dirección de cambio de las 5 divisas principales (en términos de los movimientos diarios con respecto al dólar estadounidense): el Dólar canadiense, el Marco alemán, la Libra esterlina, el Yen japonés y el Franco suizo [Lawrence et al., 1996, Giles et al., 1997]. La primera etapa en esta red combinada consiste en un mapa autorganizativo (SOM) usada para codificar los datos de la serie de tiempo en una secuencia de símbolos. La etapa siguiente es una red de Elman, usada para la predicción de las variables, sobre la información previamente codificada. La evaluación de los resultados obtenidos se realiza mediante una comparación indirecta con los resultados esperados para una caminata aleatoria. La mayoría de los autores consideran una caminata aleatoria como una variable con distribución de probabilidad uniforme, por lo que una predicción en su dirección de cambio debe resultar acertada en el 50 % de los casos (probabilísticamente). En contraste, Giles y colaboradores interpretan la hipótesis de caminata aleatoria como una predicción de que no habrá cambios en el valor de la variable, por lo que consideran que no es posible tener una medición del error en las predicciones de la red [Giles et al., 1997]. En consecuencia, la comparación de su método contra la caminata aleatoria la realizan probando el método propuesto por ellos contra los datos de una caminata aleatoria generada mediante un proceso artificial. En esta prueba obtienen errores entre 49.7 %

y 50.4 %. Por otra parte, con los datos reales los autores obtienen errores entre 47.1 % y 48.8 %, dependiendo del número de nodos en la red SOM.

Una limitante que presenta el modelado de sistemas mediante redes neuronales es que la red neuronal sólo será capaz de modelar el sistema mientras su dinámica no presente cambios drásticos. Esta característica las hace poco eficiente para modelar sistemas que, como en el caso financiero, presentan un comportamiento no estacionario.

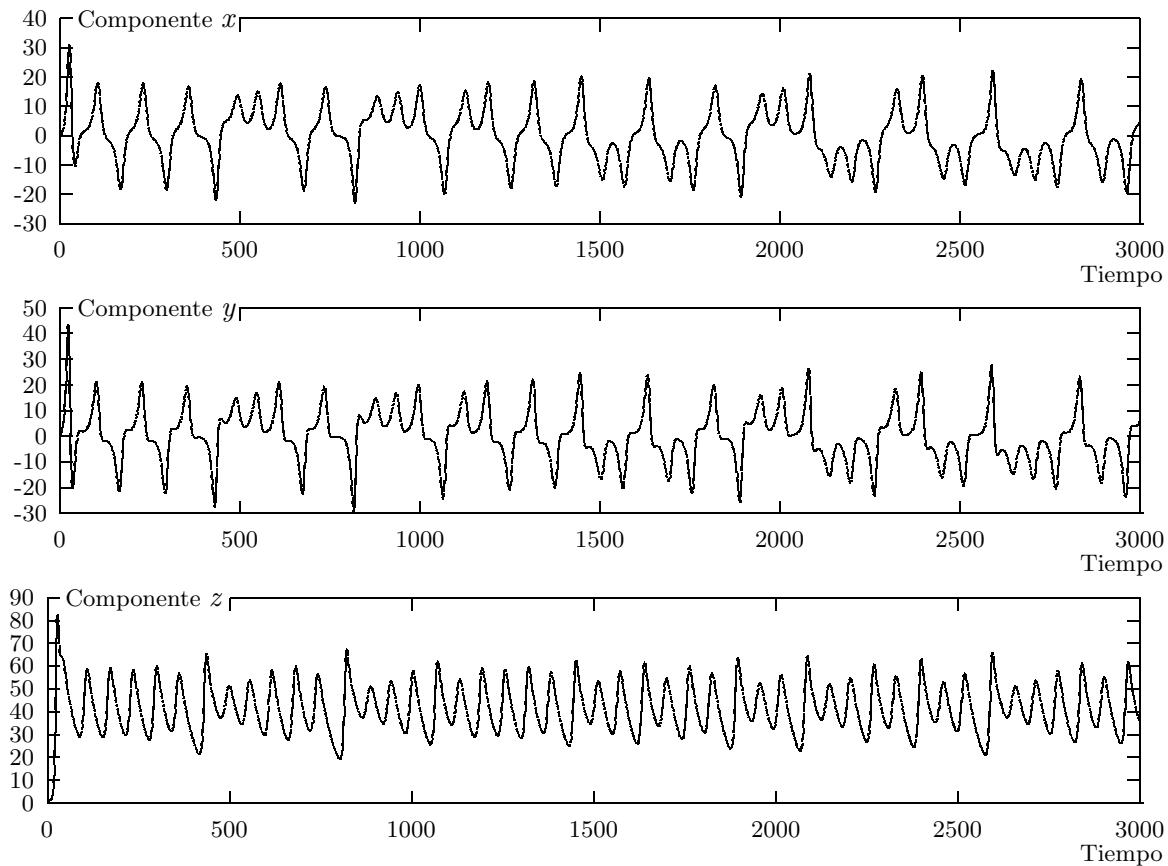


Figura 2.1: Series de tiempo para las tres componentes espaciales en el sistema de Lorenz.

2.8. Casos de estudio

Para verificar la validez de la metodología propuesta se han seleccionado tres problemas principales. Los primeros dos problemas son la predicción de dos problemas ampliamente conocidos en dinámica estocástica: el sistema de Lorenz y el sistema de Mackey-Glass; el tercer problema seleccionado es la predicción del mercado de divisas.

2.8.1. Sistema de Lorenz

El sistema de Lorenz es un sistema dinámico caótico que fue utilizado originalmente para simular el movimiento de una partícula en corrientes de convección y sistemas simplificados de clima. Este sistema está definido por el siguiente sistema de ecuaciones diferenciales [Mukherjee et al., 1997]:

$$\begin{aligned}\frac{dx(t)}{dt} &= \sigma(y - x) \\ \frac{dy(t)}{dt} &= rx - y - xz \\ \frac{dz(t)}{dt} &= xy - bz\end{aligned}\tag{2.1}$$

siendo x , y y z variables de estado y r , σ y b parámetros físicos positivos.

La figura 2.1 muestra un conjunto de datos de este sistema dinámico.

2.8.2. Sistema de Mackey-Glass

La ecuación de Mackey-Glass para describir el control de la producción de glóbulos blancos es un ejemplo bien conocido de las llamadas ecuaciones diferenciales no lineales con retraso. Esta ecuación genera una serie de tiempo caótica que se utiliza ampliamente como problema de prueba para los métodos de predicción de series de tiempo.

El sistema de Mackey-Glass está definido por la ecuación

$$\frac{dx(t)}{dt} = 0,2x(t - \tau)/(1 + x(t - \tau)^{10}) - 0,1x(t)\tag{2.2}$$

Debido a que el foco de la metodología presentada en este trabajo es la predicción de bases de datos de series de tiempo, se propone una variación sobre el sistema de Mackey-Glass, formado por tres ecuaciones diferenciales acopladas. Este sistema de ecuaciones fue utilizado para generar los datos de prueba. El sistema de ecuaciones diferenciales usado es para describir el sistema acoplado de Mackey-Glass es

$$\begin{aligned}\frac{dx(t)}{dt} &= 0,2y(t - \tau)/(1 + y(t - \tau)^{10}) - 0,1x(t) \\ \frac{dy(t)}{dt} &= 0,2z(t - \tau)/(1 + z(t - \tau)^{10}) - 0,1y(t) \\ \frac{dz(t)}{dt} &= 0,2x(t - \tau)/(1 + x(t - \tau)^{10}) - 0,1z(t)\end{aligned}\tag{2.3}$$

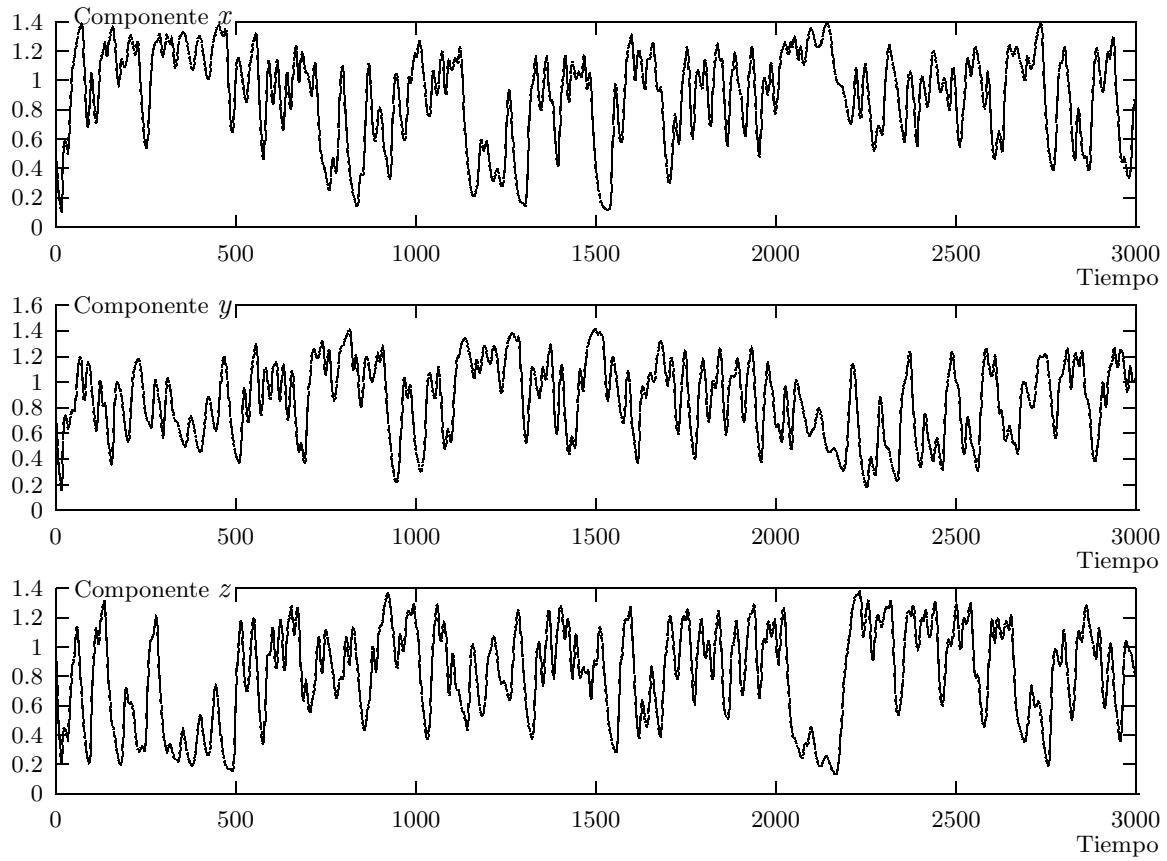


Figura 2.2: Serie de tiempo generada por el sistema de ecuaciones acopladas de Mackey-Glass.

La figura 2.2 muestra un conjunto de datos generados mediante este sistema dinámico.

2.8.3. Mercado de divisas

Los sistemas financieros son una clase particularmente interesante de sistemas dinámicos reales, que presentan muchos problemas al ser modelados, sobresaliendo la no linealidad de los datos, su carácter no estacionario y un alto nivel de ruido [Lee-Giles et al., 2001, Magdon-Ismail et al., 1998]. Los métodos de análisis financiero más empleados se basan en conceptos carentes de base científica (como la tendencia natural de los precios) y su verdadera utilidad ha sido cuestionada seriamente (véase, por ejemplo, [Hellström and Hellström, 1998]). Por otra parte, aunque el comportamiento de las variables económicas es determinado por factores diversos y complejos, lo que hace que algunos acontecimientos sean impredecibles [Yao and Poh, 1995, Şirin and Güvenir, 1993, Moody, 1995], tales sistemas presentan algunas características que los convierten en buenos casos de estudio para probar técnicas de

predicción, sobresaliendo la claridad de los objetivos a predecir y la disponibilidad de datos en diferentes frecuencias [Hellström and Hellström, 1998, Magdon-Ismail et al., 1998].

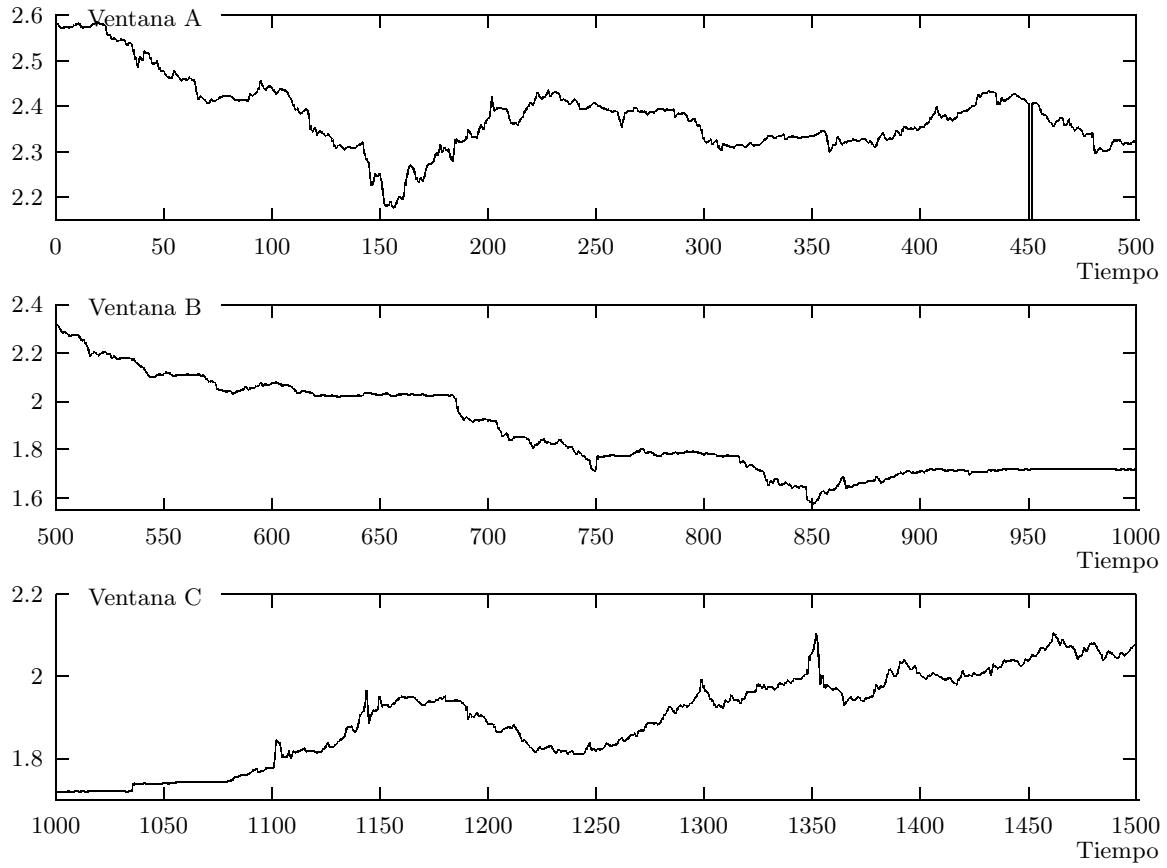


Figura 2.3: Tres ventanas consecutivas de datos de una serie de tiempo para la tasa de cambio de una divisa.

El sistema financiero específico que se analizará en este trabajo es el mercado de divisas. Una de las dificultades principales que se presentan, cuando se intenta simular tal sistema, es que se aproxima en gran medida a la llamada hipótesis de mercado eficiente (HME) [Hellström and Hellström, 1998, Lee-Giles et al., 2001, Yao et al., 1996]. La HME establece que la mejor predicción del precio de una mercancía es su valor actual, lo que significa que no es posible predecir su comportamiento. Sin embargo, la mayoría de los autores consideran que no hay evidencia que sustente la HME de una manera concluyente para cualquier mercado, en todo momento y toda circunstancia [Yao and Poh, 1995, Lawrence et al., 1996]. De hecho, es evidente que el mercado debe tener ciertas ineficiencias, lo que permite que a diario puedan hacerse predicciones limitadas sobre él.

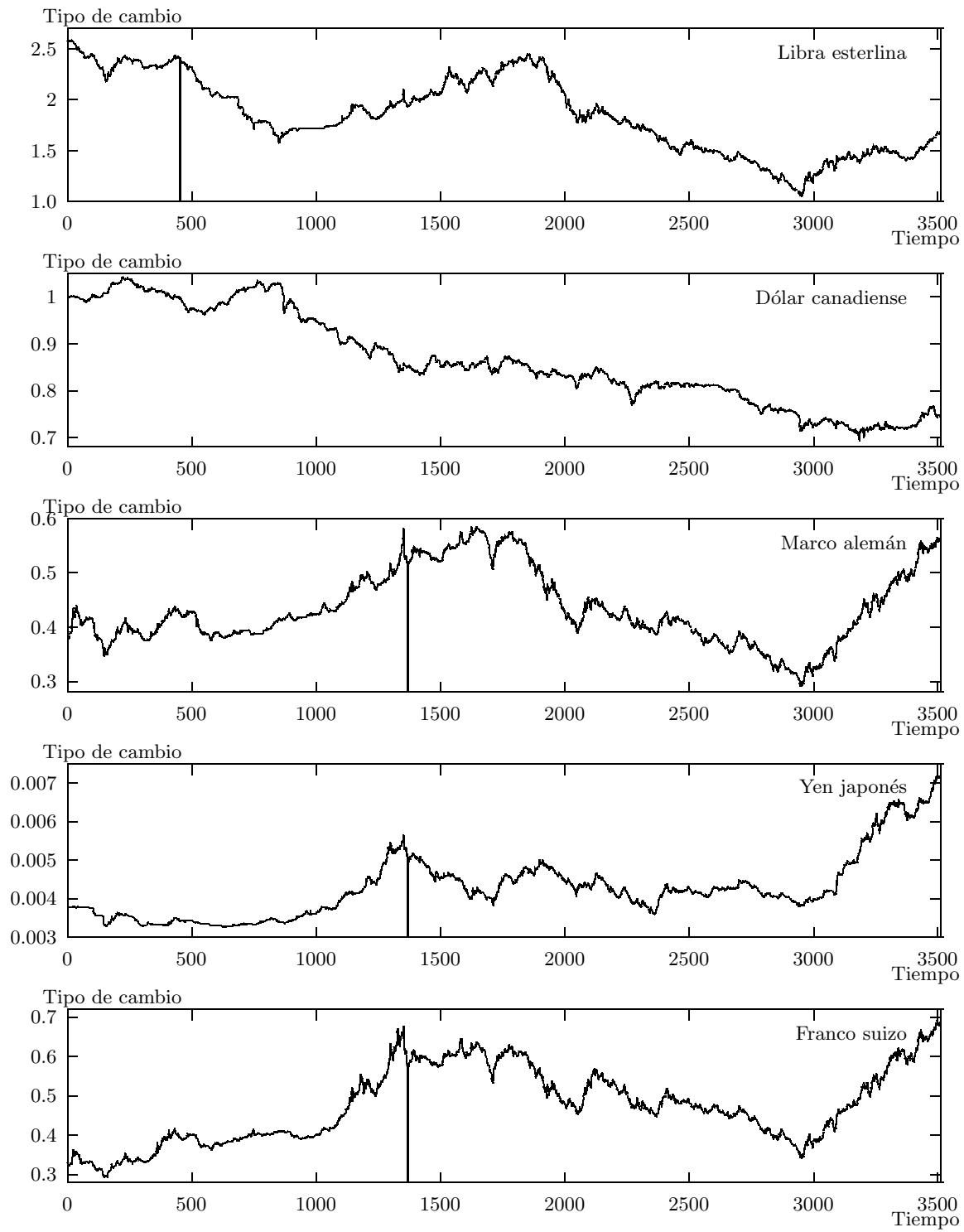


Figura 2.4: Series de tiempo del mercado de divisas.

La predicción de las divisas se basa en encontrar pequeñas ineficacias del mercado que se espera no durarán mucho tiempo. Por otra parte, tal sistema muestra un comportamiento no estacionario evidente que requiere analizar series de tiempo extensas. La dificultad de la generalización en este problema es evidente en la figura 2.3, que muestra tres ventanas sucesivas de los datos de una serie de tiempo de la tasa de cambio de una divisa: Si los datos de entrada en la ventana A y los datos objetivo en la ventana B se utilizan para entrenar a un sistema de modelado, el conocimiento adquirido en este proceso difícilmente se puede utilizar para predecir los datos en la ventana C.

En el presente trabajo, el análisis del mercado de divisas se realiza sobre la base de un conjunto de series de tiempo para las diversas variables que definen el sistema, suponiendo que pueden existir dependencias desconocidas entre esas variables. Los datos usados en este problema fueron tomados originalmente del *Monetary Yearbook of the Chicago Mercantile Exchange* y publicados en Internet por Andreas S. Weigend (datos tomados de <http://www.cs.colorado.edu/~andreas/Time-Series/Data/>). Esta base de datos contiene 3512 registros que cubren un período del 1 de junio de 1973 al 21 de mayo de 1987 del Marco Alemán, el Yen Japonés, el Franco Suizo, la Libra Esterlina y el Dólar Canadiense.

Capítulo 3

Lenguaje difuso dinámico

Los conjuntos difusos han mostrado ser una buena herramienta para representar las categorías cualitativas que describen el comportamiento de un sistema dinámico. Los conjuntos difusos estándar arrojan buenos resultados para el modelado de sistemas no lineales [Kim and Lee, 1999, Tsai and Wu, 1999], así como en el manejo de datos imprecisos. Los conjuntos difusos de segundo orden, por su parte, han arrojado buenos resultados al analizar datos con altos niveles de ruido (ver, por ejemplo, [Karnik and Mendel, 1999]). Sin embargo, la mayoría de las aproximaciones a la descripción difusa de datos son de naturaleza atemporal y no representan la evolución dinámica de esas categorías, por lo cual no arrojan buenos resultados con series de tiempo no estacionarias. Ozawa et. al. han propuesto variantes difusas de los modelos de Box-Jenkins para modelar sistemas no estacionarios [Ozawa et al., 1997, Ozawa and Niimura, 1999]. Sin embargo, una alternativa más natural y menos explorada consiste en incluir una dimensión temporal en la propia estructura de los conjuntos difusos. Los conjuntos difusos temporales y dinámicos se han propuesto con el objetivo de describir la naturaleza dinámica del mundo [Kosanović et al., 1996].

En este capítulo se presenta una extensión del concepto de conjunto difuso dinámico, que ha sido denominado como conjunto difuso dinámico suave. Estos conjuntos son empleados para la predicción de series de tiempo y, en un contexto más amplio en este trabajo, como un lenguaje descriptivo de datos que es usado en los SMA para modelado híbrido inteligente

3.1. Conjuntos difusos dinámicos

Los conjuntos difusos intentan describir los estados o categorías cualitativas del mundo. Dado un universo de discurso U , un subconjunto difuso $A \subseteq U$ se define como

$$A = \{(x, \mu_A(x)) \mid x \in U, \mu_A : U \rightarrow [0, 1]\} \quad (3.1)$$

Donde $\mu_A(x)$ es el grado *fijo* de pertenencia de x en el conjunto A .

Los conjuntos difusos estándar capturan adecuadamente representaciones instantáneas del mundo, pero se mantiene fijos a lo largo del intervalo de tiempo de observación, aún cuando la dinámica del sistema experimente cambios en ese mismo intervalo de tiempo. Ésta es una descripción válida para aquellos casos en los que la categoría física no cambia con el tiempo. Las categorías cualitativas que corresponden a los estados de sistemas dinámicos suelen mostrar, por otro lado, una evidente evolución temporal. De esta manera, los valores normales para el precio de un producto dado, la demanda de energía eléctrica y el nivel de actividad solar, varían a lo largo de tiempo, en algunos casos mostrando cierta periodicidad. Los conjuntos difusos estándar son una herramienta limitada para describir tales cambios en sistemas dinámicos, así pues, una nueva clase de conjunto difuso se introduce como sigue.

Si A es un conjunto difuso que describe una cierta categoría dinámica, es natural esperar que el valor de pertenencia de un valor dado $x \in U$ en el conjunto A pueda cambiar a lo largo del intervalo del tiempo T , según una cierta función temporal de pertenencia

$$\mu^x(t) : T \rightarrow [0, 1]$$

que describe la forma en que cambia la pertenencia del elemento $x \in U$ en el conjunto A conforme pasa el tiempo. De manera que la función de pertenencia del conjunto A toma entonces la forma

$$\mu_A : U \rightarrow \{\mu^x(t)\}$$

Definición 3.1. Un sistema difuso dinámico A_D puede definirse como

$$A_D = \{(x, \mu^{(x)}(t)) \mid \mu^{(x)}(t) \in [0, 1], t \in T\} \quad (3.2)$$

Los conjuntos difusos dinámicos definidos de esta manera representan superficies en el espacio $U \times T \times [0, 1]$. Una serie de tiempo $S \subset U \times T$ representa la proyección en el plano $U \times T$ de una curva en la superficie definida por el conjunto difuso dinámico. Tal curva

es la función de pertenencia de un conjunto difuso temporal, de acuerdo a la definición en [Kosanović et al., 1996].

La forma de un conjunto difuso paramétrico puede definirse por tres parámetros principales:

1. La función que define la forma general del conjunto difuso, que es una descripción abstracta del fenómeno modelado.
2. El valor modal que define la posición del conjunto difuso en el dominio de la variable que se codificará, que se toma a menudo como el valor representativo de la categoría o el punto de pertenencia 1.
3. El valor del ancho de banda del conjunto difuso, que se relaciona con la tolerancia sobre la variabilidad en el valor cuantitativo de la variable.

Aunque estos tres parámetros pueden cambiar a lo largo del tiempo para un mismo fenómeno, la forma general del conjunto difuso difícilmente cambiará a lo largo del tiempo. Por otra parte, el valor modal es el parámetro que presenta más cambios en un conjunto difuso. En las secciones siguientes se analizarán los efectos de la variación en el valor modal y el valor del ancho de banda de un conjunto difuso, así mismo, se comparan los efectos sobre los conjuntos difusos estándar y los conjuntos difusos dinámicos.

3.2. Variación del valor modal de los conjuntos difusos

Una situación usual cuando se emplean conjuntos difusos para describir conceptos en una serie de tiempo, es que se desconozca el valor modal adecuado *a priori*. Una aproximación común para elegir el valor modal de un conjunto difuso en tales condiciones consiste en seleccionar el valor medio de los datos disponibles. Esta solución es adecuada si la serie de tiempo es estacionaria. En esta sección se analizarán los efectos de esta aproximación cuando la serie es no estacionaria.

En la figura 3.1 se muestran los resultados obtenidos al codificar una serie de tiempo que muestra una tendencia de crecimiento cuadrática mediante cuatro estrategias diferentes para la elección del valor modal de los conjuntos difusos. Los datos han sido generados por la función

$$x(t) = \frac{1}{2000}t^2 + u(t) \quad (3.3)$$

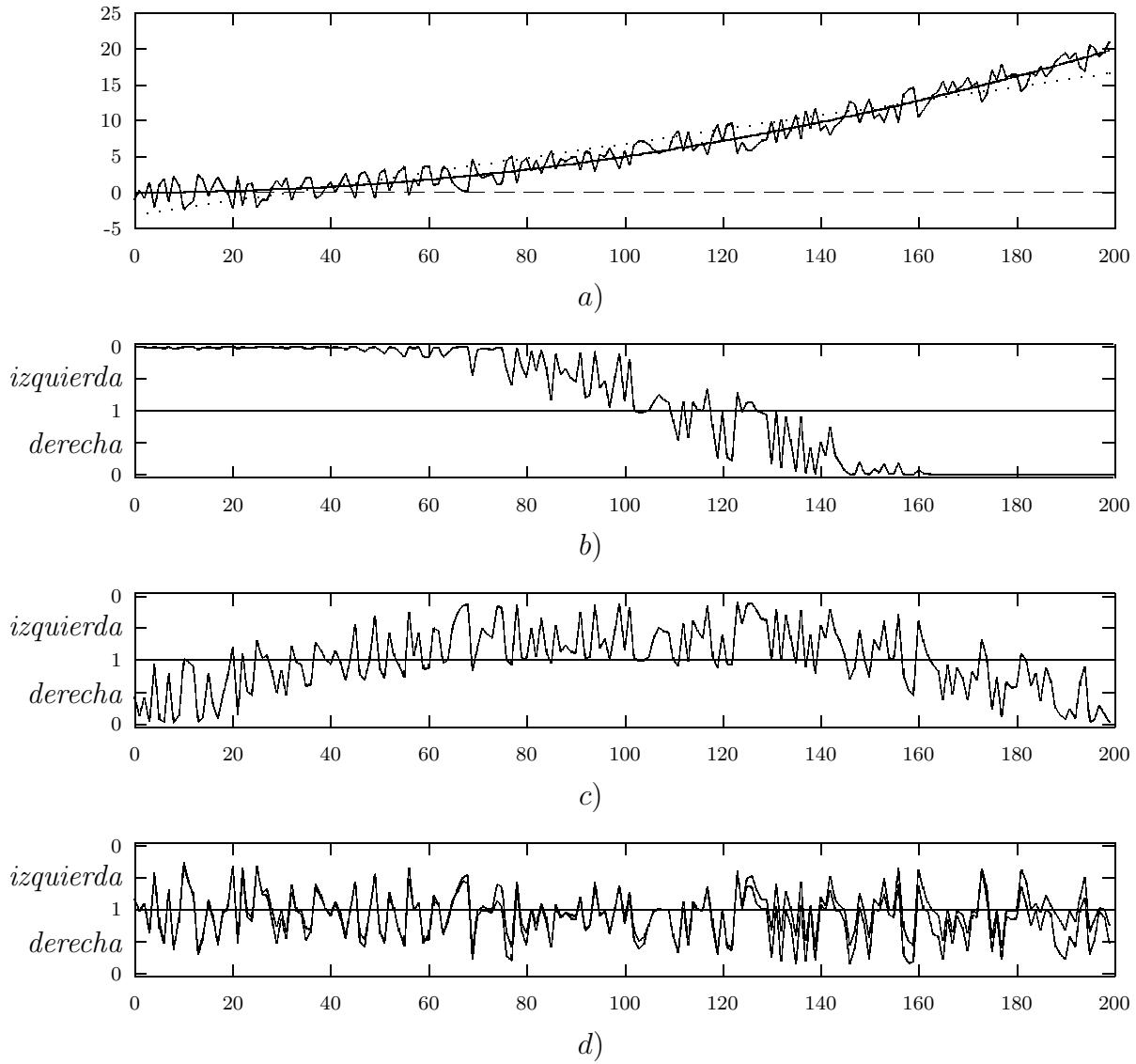


Figura 3.1: Comportamiento dinámico del valor "normal" de categorías difusas.

- a) Datos originales y la curva que describe su tendencia cuadrática. b) Datos codificados usando un conjunto difuso fijo cuyo valor modal es el valor medio global de los datos. c) Codificación usando conjuntos difusos cuyo valor modal es el valor correspondiente de la regresión lineal. d) En esta gráfica se muestran los resultados de dos estrategias de codificación: en una de ellas se utilizó un conjunto difuso cuyo valor modal cambia de acuerdo a la curva cuadrática. La segunda codificación utiliza conjuntos difusos cuyo valor modal en cada punto es el valor medio sobre los 10 valores pasados en la serie de tiempo.

donde $u(t)$ es una función generadora de números pseudoaleatorios con distribución uniforme, tal que $u(t) \in [-2,5, 2,5]$. Estos datos se muestran en la gráfica 3.1.a.

En las cuatro estrategias de codificación, se emplearon conjuntos difusos descritos mediante funciones gaussianas para describir qué tan "normal" es un valor cuantitativo determinado para la variable, es decir, qué tanto se aproxima el valor cuantitativo en la serie de tiempo al valor cuadrático esperado.

En la gráfica 3.1.b se muestran los resultados obtenidos al usar el valor medio del conjunto completo de datos como valor modal. En este caso la función de pertenencia se define como:

$$\mu(x(t)) = e^{-((x(t) - \bar{x})/\sigma)^2} \quad (3.4)$$

donde \bar{x} es el valor medio sobre el conjunto completo de valores cuantitativos y σ su desviación estándar.

Los valores codificados hacia el lado izquierdo de la curva gaussiana corresponden a valores cuantitativos por abajo del valor esperado correspondiente, mientras que los valores cuantitativos por arriba del valor esperado se codifican a la derecha de la curva. Como puede apreciarse en esta gráfica, los primeros 100 valores cuantitativos en la serie de tiempo son codificados como si fueran más pequeños que el valor esperado, sin importar su comparación con el valor cuadrático correspondiente, mientras que los últimos 70 valores se codifican como si fueran superiores a los valores cuadráticos. Pero los datos se encuentran distribuidos de manera uniforme alrededor de la curva cuadrática, por lo que deberían distribuirse por igual a derecha e izquierda. De esta manera, tal codificación no parece reflejar adecuadamente el comportamiento real de los datos.

Una aproximación usual para analizar series no estacionarias, particularmente entre economistas, consiste en suponer una tendencia de crecimiento de los datos, típicamente lineal y eliminarla de los datos originales para obtener una serie de tiempo estacionaria. La segunda estrategia de codificación en el presente experimento equivale a eliminar una supuesta tendencia lineal de los datos en la serie de tiempo. Para ello, se obtiene mediante el método de ajuste por mínimos cuadrados, la línea recta que mejor aproxima a los datos. A continuación, cada punto en la serie de tiempo es codificado mediante un conjunto difuso diferente, cuyo valor modal coincide con el valor de la línea recta, calculado para el mismo tiempo. Los resultados de esta aproximación se presentan en la gráfica 3.1.c. Tal secuencia de conjuntos difusos conforma un conjunto difuso dinámico. Esta estrategia de codificación también arroja una distribución de los datos codificados alrededor del conjunto difuso que no parecen reflejar el comportamiento real de los datos. El error obtenido cuando se realiza la codificación

de los datos asumiendo una tendencia incorrecta es evidente de la gráfica, donde los datos codificados muestran una distribución cuadrática en torno al valor modal.

En la gráfica 3.1.d, se presentan los resultados obtenidos cuando se usan los valores de la tendencia cuadrática real como valores modales de los conjuntos difusos dinámicos.

En la imagen 3.1.e los datos son codificados mediante un conjunto difuso dinámico que toma como valor modal en cada punto el valor promedio de los últimos 10 datos (promedios móviles), mientras que su ancho de banda es determinado por la desviación estándar sobre los mismos datos. La pertenencia de $x(t)$ al conjunto difuso "normal" está dado por:

$$\mu(x(t)) = e^{-((x(t) - \bar{x}_t)/\sigma_t)^2} \quad (3.5)$$

siendo

$$\bar{x}_t = \bar{x}_{t-1} + \frac{1}{n}(x(t-1) - x(t-n-1))$$

y

$$\sigma_t = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x(t-i) - \bar{x}_t)^2}$$

La representación de conceptos mediante conjuntos difusos tiene dos objetivos principales: 1) abordar problemas de información imprecisa, y 2) analizar un problema mediante variables lingüísticas, de manera semejante a como lo hacen los humanos. De acuerdo con el segundo de estos objetivos, los mejores resultados en la codificación mediante conjuntos difusos dinámicos son obtenidos usando la tendencia real de los datos, si es que ésta es conocida. Por otra parte, el uso de promedios móviles sobre los valores cuantitativos también arroja buenos resultados, sin el uso de conocimiento exacto sobre la dinámica del sistema. De esta manera, se define una especialización de los conjuntos difusos dinámicos en los siguientes términos

Definición 3.2. Un conjunto difuso dinámico suave es un conjunto difuso dinámico cuyos parámetros son definidos por los valores cuantitativos anteriores en la serie de tiempo.

Una observación importante de este experimento es la siguiente: aunque la mejor aproximación para elegir los valores modales de los conjuntos difusos dinámicos se obtiene cuando se ajustan a la dinámica del sistema, cuando esta dinámica es desconocida es preferible emplear una aproximación dinámica suave que intentar ajustar los datos a una tendencia simple (por ejemplo a la regresión lineal) o a una tendencia presupuesta.

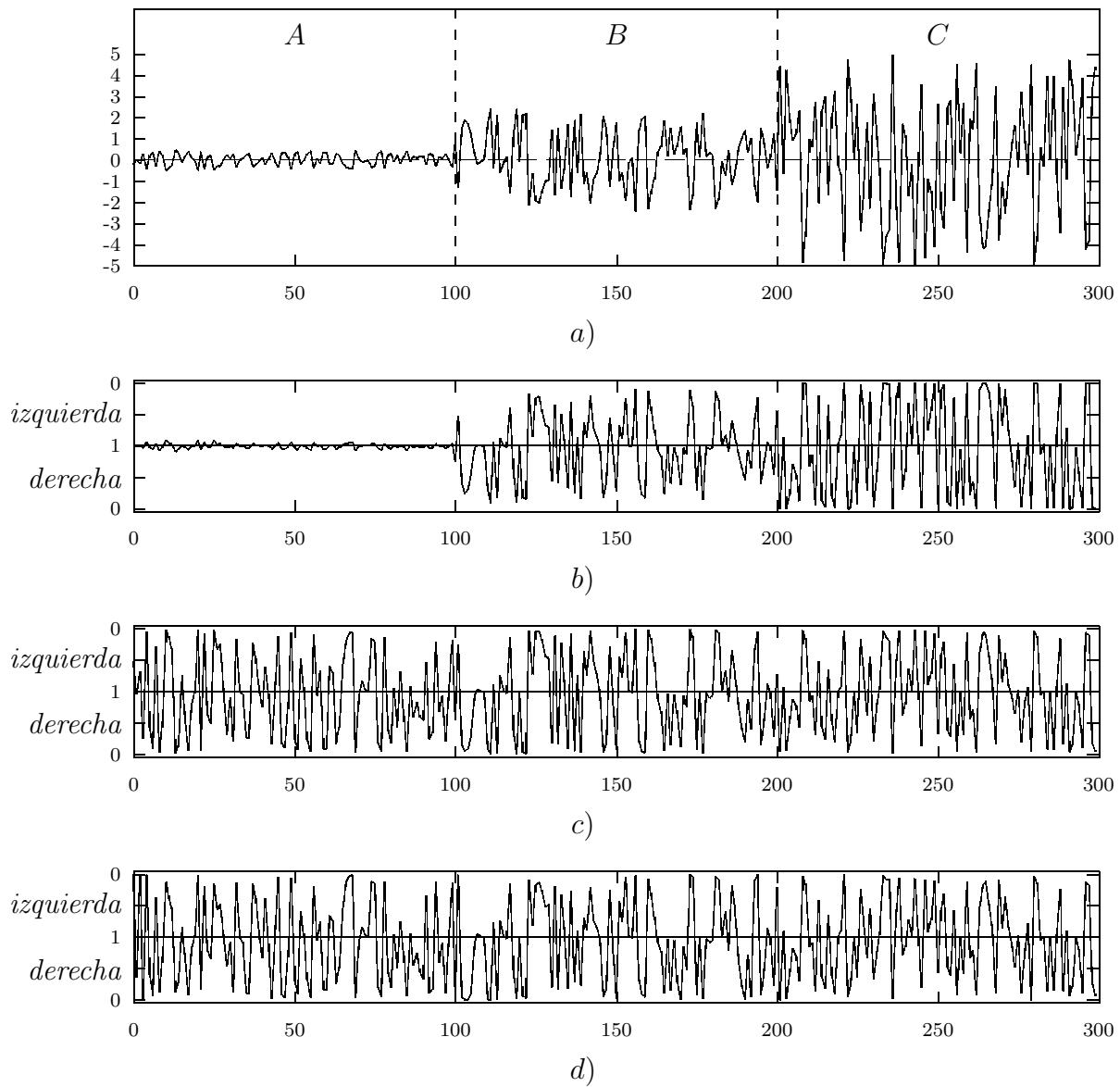


Figura 3.2: Comportamiento dinámico del ancho de banda de los conjuntos difusos.
 a) Datos originales divididos en 3 regiones con diverso rango de variación. b) Codificación empleando un conjunto difuso con ancho de banda fijo. c) Codificación utilizando un valor de ancho de banda diferente en cada región, calculada a partir de los datos en la región. d) Codificación realizada mediante conjuntos difusos cuyo valor de ancho de banda se calcula según la variación local de los datos.

3.3. Variación en el ancho de banda de los conjuntos difusos

En la figura 3.2 se presentan los resultados obtenidos al codificar un conjunto de datos con cambios en el rango de variación, mediante tres estrategias diferentes para la elección del ancho de banda de los conjuntos difusos. Los datos originales, que se muestran en la imagen 3.2.a), fueron generados como:

$$x(t) = g(t) \quad (3.6)$$

donde $g(t)$ es una función pseudoaleatoria con distribución gausiana tal que

$$g(t) = \begin{cases} [-0,5, 0,5] & t < 100 \\ [-5, 5] & 100 \leq t < 200 \\ [-10, 10] & t > 200 \end{cases} \quad (3.7)$$

Como puede observarse, los datos pueden fragmentarse en tres regiones A , B y C con diferentes rangos de variación. Los datos se codifican usando tres estrategias basadas en la desviación estándar de las diferencias adyacentes de los datos originales. La diferencia adyacente de la variable x en el tiempo t_k está dada por

$$r_k = x_k - x_{k-1} \quad (3.8)$$

En la primera estrategia (gráfica 3.2.b) se utiliza un conjunto difuso cuyo ancho de banda es proporcional a la desviación estándar calculada sobre el conjunto completo de diferencias adyacentes (ecuación 3.4). El valor modal se toma como el valor medio sobre todos los datos. La codificación mediante esta estrategia describe adecuadamente la variación de los datos en un contexto global. Sin embargo, en sistemas dinámicos con comportamiento no estacionario, suele ser más importante la información local.

La segunda estrategia (gráfica 3.2.c) utiliza valores distintos de ancho de banda para codificar los valores cuantitativos en cada región. Cada valor de ancho de banda elegido es proporcional al valor de la desviación estándar calculado sobre las diferencias adyacentes de la región correspondiente. El valor modal se toma también como el valor medio sobre los datos en la misma región (hay que notar que los datos para este experimento se distribuyen simétricamente en torno a 0, por lo cual es de esperar que éste sea una elección adecuada de valor modal). Como puede observarse, la codificación ignora las diferencias en los rangos de variación de las diferentes regiones y, en cambio, refleja las variaciones locales.

En la última estrategia (gráfica 3.2.c), la codificación se realiza mediante conjuntos difusos dinámicos cuyo ancho de banda se calcula individualmente para cada punto, siendo su valor proporcional a la desviación de estándar sobre las diferencias adyacentes de los 10 últimos valores cuantitativos anteriores al valor que se codificará.

En la figura 3.3 se presentan los resultados de una tarea más difícil. En este caso, los datos que se codificarán (gráfica 3.3.a) se generan como:

$$x(t) = \frac{1}{2}t + g(t) \quad (3.9)$$

donde $g(t)$ es una función pseudoaleatoria con distribución gausiana tal que

$$g(t) = \begin{cases} [-0,5,0,5] & t < 100 \\ [-5,5] & 100 \leq t < 200 \\ [-10,10] & t > 200 \end{cases} \quad (3.10)$$

La codificación mostrada en la gráfica 3.3.b fue realizada usando un conjunto difuso fijo cuyo valor modal es el valor medio del conjunto completo de datos y el valor de ancho de banda del conjunto fue elegido como la desviación estándar sobre el conjunto completo de diferencias adyacentes. En la figura 3.3.c se observa los resultados cuando se emplea un conjunto difuso dinámico cuyos valores modales coinciden con los puntos sobre la línea que define la tendencia de los datos, mientras que el valor de ancho de banda se calcula localmente en cada región ($t < 100$, $100 \leq t < 200$ y $t \geq 200$). Los resultados en la gráfica 3.3.d) se obtienen usando un conjunto difuso dinámico suave cuyo valor modal corresponde al valor anterior y el valor de ancho de banda se elige localmente como la desviación estándar sobre los 10 valores cuantitativos anteriores.

3.4. Estabilidad local

La manera usual de definir categorías difusas consiste en elegir una etiqueta lingüística que describa el fenómeno a modelar y especificar la función de pertenencia correspondiente. De esta manera, un horno casero con una temperatura, por ejemplo, de 50°C se podría describir como "no muy caliente" y un hombre de 120 años de edad se diría que es "un hombre muy viejo". Pero, las variables financieras se analizan generalmente en diferentes términos. Tales variables se describen comúnmente en términos de su estabilidad local: los precios, por ejemplo, se dice que están a la alza o a la baja. Partiendo de esta metáfora, los datos generados por otros sistemas dinámicos pueden ser codificados en términos de tres conjuntos

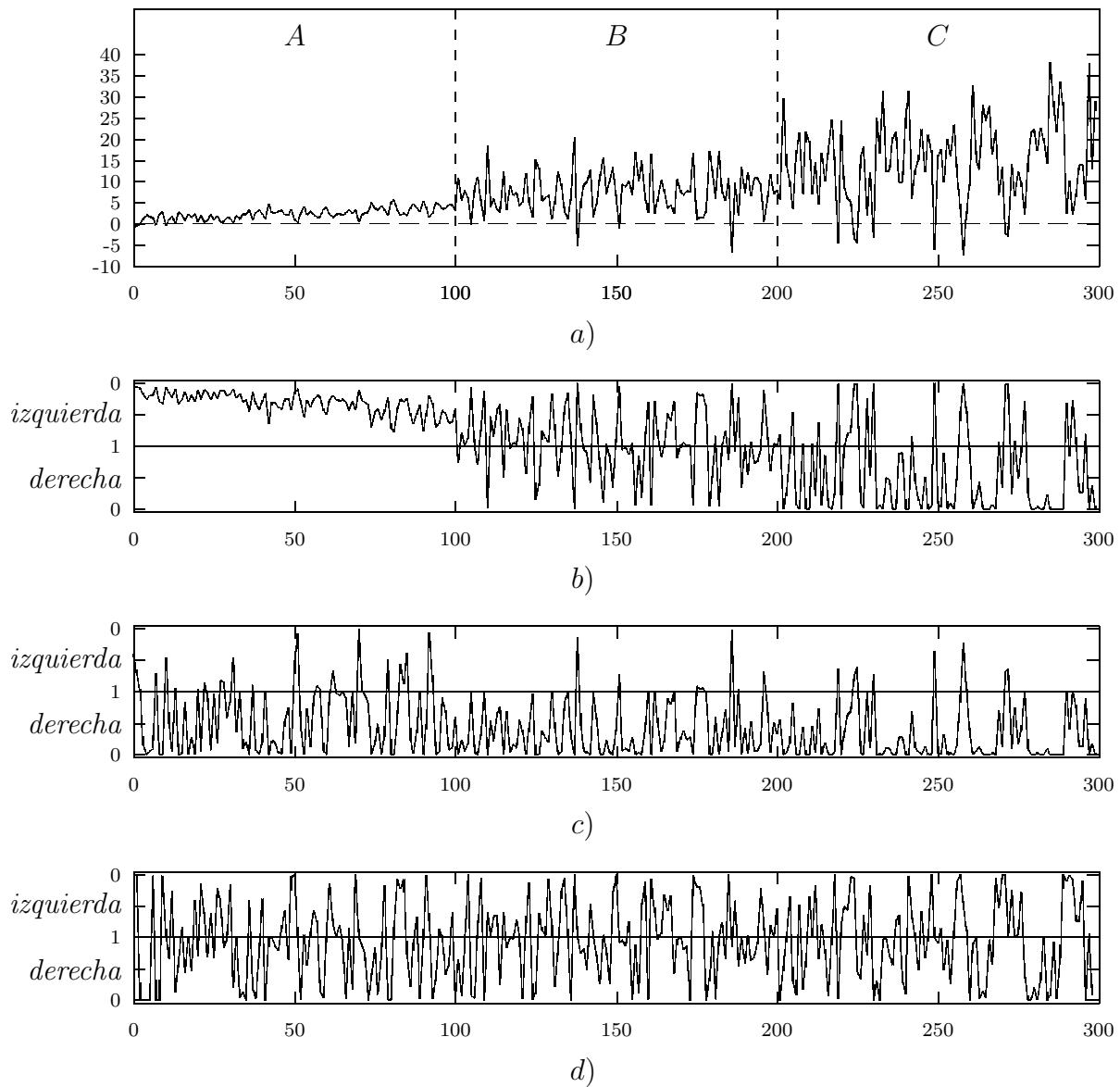


Figura 3.3: Comportamiento dinámico de las categorías difusas que describen datos con variación combinada del ancho de banda y del valor esperado.

a) Datos originales. b) La codificación usando un conjunto difuso fijo. c) Codificación utilizando un conjunto difuso fijo en cada región. d) Codificación usando conjuntos difusos que se adaptan al comportamiento local de los datos.

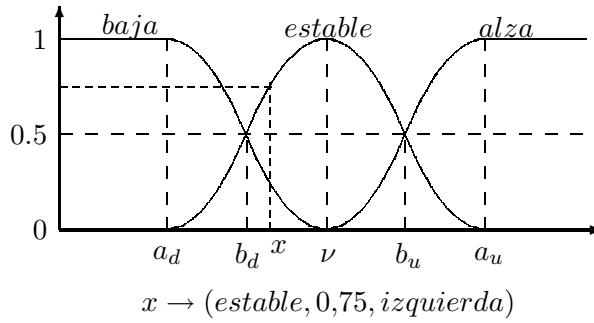


Figura 3.4: Conjuntos difusos para describir la estabilidad local de un sistema dinámico.

difusos, denominados como "baja", "estable" y "alza", como se muestra en la figura 3.4. Tales conjuntos difusos describen la estabilidad local de la variable, en relación a sus valores anteriores, y se define como sigue:

- El conjunto difuso "estable" está definido por una curva logística creciente y una curva logística decreciente. El valor de pertenencia de la variable x en el conjunto difuso estable está dado entonces por

$$\mu_E(x) = \begin{cases} 0 & x \leq a_d \\ 2 \left[\frac{a_d - x}{\nu - a_d} \right]^2 & a_d < x \leq b_d \\ 1 - 2 \left[\frac{\nu - x}{\nu - a_d} \right]^2 & b_d < x \leq b_u \\ 2 \left[\frac{a_u - x}{a_u - \nu} \right]^2 & b_u < x < a_u \\ 0 & x \geq a_u \end{cases} \quad (3.11)$$

donde ν es el valor modal del conjunto y representa un valor de referencia para los tres conjuntos difusos; a_d, b_d, b_u y a_u son valores tales que

$$|\nu - a_d| = |\nu - a_u|$$

$$b_d = \frac{\nu + a_d}{2}$$

$$b_u = \frac{\nu + a_u}{2}$$

Esta función aproxima la forma de una curva gaussiana, presentando la característica adicional de tomar el valor cero fuera de un rango determinado. Tal estrategia de codificación describe una variable como estable siempre que su variación, referida al valor anterior, no sea *tan alta*. Conforme el valor de la variable se separa del valor modal, se describe como menos estable.

- El conjunto "baja" se elige como un escalón unitario y una curva logística decreciente cuyo punto de pertenencia cero coincide con el centro del conjunto difuso estable. La pertenencia de la variable x en este conjunto difuso está dada por

$$\mu_B(x) = \begin{cases} 1 & x < a_d \\ 1 - 2 \left[\frac{a_d - x}{\nu - a_d} \right]^2 & a_d \leq x < b_d \\ 2 \left[\frac{\nu - x}{\nu - a_d} \right]^2 & b_d \leq x < \nu \\ 0 & x \geq \nu \end{cases} \quad (3.12)$$

El valor de pertenencia en el conjunto "baja" es 0 cuando la variable toma un valor igual o superior al valor modal ν , y toma el valor de pertenencia 1 cuando el valor cuantitativo disminuye considerablemente. Conforme el valor de la variables x disminuye, se califica como estando a la "baja" y al mismo tiempo como permaneciendo "estable", siempre que la reducción sea pequeña .

- El conjunto "alza" se elige como una curva logística creciente cuyo punto de pertenencia 0 coincide con el centro del conjunto difuso estable y con el punto de pertenencia 0 del conjunto "baja" y un escalón unitario. Así, la función de pertenencia al conjunto está dada por

$$\mu_A(x) = \begin{cases} 0 & x < \nu \\ 2 \left[\frac{\nu - x}{\nu - a_u} \right]^2 & \nu \leq x \leq b_u \\ 1 - 2 \left[\frac{a_u - x}{\nu - a_u} \right]^2 & b_u \leq x \leq a_u \\ 1 & x \geq a_u \end{cases} \quad (3.13)$$

El valor de pertenencia en el conjunto difuso "alza" es cero cuando la variable mantiene el mismo valor que el del valor modal, o cuando toma un valor inferior. A medida que el valor de la variable x aumenta, se designa como estando a la "alza" y al mismo tiempo como estable, si el incremento es pequeño.

La codificación difusa en este trabajo se basa en el método cualitativo de codificación usado en la técnica de razonamiento inductivo difuso (FIR) [Cellier, 1991, de Albornoz, 1996]. Tal algoritmo de codificación describe un valor x en el universo de discurso mediante una tercia $r = (clase, pertenencia, lado)$. Esta descripción está formada por un valor cualitativo que define la clase difusa a la que pertenece x , un valor numérico de pertenencia a la clase y un valor cualitativo adicional de lado de función, que especifica hacia que lado de la función de pertenencia se localiza r (véase la figura 3.4). El valor de lado toma los siguientes valores en cada uno de los conjuntos:

- Conjunto "estable"

$$lado_E(x) = \begin{cases} izquierda & x < \nu \\ centro & x = \nu \\ derecha & x > \nu \end{cases} \quad (3.14)$$

- Conjunto "baja"

$$lado_B(x) = \begin{cases} izquierda & x < a_d \\ centro & x = a_d \\ derecha & x > a_d \end{cases} \quad (3.15)$$

- Conjunto "alza"

$$lado_A(x) = \begin{cases} izquierda & x < a_u \\ centro & x = a_u \\ derecha & x > a_u \end{cases} \quad (3.16)$$

Ésta es una descripción conveniente para la estabilidad dinámica local: Como ejemplo, una variable declarada como *estable* con tendencia *a la baja*, se puede codificar como (*estable*, μ , *izquierda*). En este esquema de codificación, un valor cuantitativo se trata como perteneciente solamente al conjunto difuso donde es más grande su valor de pertenencia.

Así pues, un valor cuantitativo se puede codificar según el esquema siguiente:

$$\mathcal{X} = \begin{cases} (baja, 1, izquierda) & x \leq a_d \\ (baja, 1 - 2 \left[\frac{a_d - x}{\nu - a_d} \right]^2, derecha) & a_d < x \leq b_d \\ (estable, 1 - 2 \left[\frac{\nu - x}{\nu - a_d} \right]^2, izquierda) & b_d < x < \nu \\ (estable, 1, centro) & x = \nu \\ (estable, 1 - 2 \left[\frac{\nu - x}{\nu - a_u} \right]^2, derecha) & \nu < x < b_u \\ (alza, 1 - 2 \left[\frac{a_u - x}{\nu - a_u} \right]^2, izquierda) & b_u \leq x < a_u \\ (alza, 1, derecha) & x \geq a_u \end{cases} \quad (3.17)$$

Si el valor modal para un registro dado no está disponible al momento de codificarlo, la variable se asigna al conjunto estable con un valor de pertenencia 1, lo que equivale, en el caso de sistemas financieros, a aceptar la hipótesis de mercado eficiente.

Esta codificación es equivalente, pero más económica y comprensible que la aproximación usada generalmente, consistente en codificar un valor cuantitativo basado en sus valores de pertenencia a varios conjuntos difusos. De esta manera, el ejemplo de la figura 3.4, (*estable*, 0, 75, *izquierda*), se podría representar de una manera más convencional como ($\mu_{baja} = 0,25$, $\mu_{estable} = 0,75$, $\mu_{alza} = 0$). Dos conceptos complementarios a esta representación, que serán utilizados ampliamente en este trabajo, son el de "conjuntos vecinos" y "apuntar hacia un conjunto". Estos conceptos se definen como sigue:

Definición 3.3. Dos conjuntos difusos α y β se dice que son conjuntos difusos vecinos si y sólo si $\alpha \neq \beta$ y $\alpha \cap \beta \neq \phi$, donde ϕ es el conjunto vacío. Esta relación es denotada por $\mathcal{N}(\alpha, \beta)$.

Definición 3.4. El registro r se dice que apunta hacia el conjunto difuso α si y sólo si: 1) $r \in \beta$ y $\mathcal{N}(\alpha, \beta)$, y 2) el conjunto difuso α está situado al mismo lado del conjunto difuso β donde se localiza r . Esta relación se denota como $r \mapsto \alpha$.

De esta manera

- $\mathcal{N}(baja, estable)$ y $\mathcal{N}(alza, estable)$.
- Si $r = (estable, \mu, derecha)$, entonces $r \mapsto alza$.

3.4.1. Semejanza/distancia suave de Gower

La proximidad entre dos tercias cualitativas se mide mediante una medida de semejanza derivada del llamado "coeficiente de similitud de Gower". El coeficiente de similitud de Gower [Gower, 1971] ha sido ampliamente utilizado para medir la semejanza entre datos no cuantitativos y entre datos mezclados, en diversas áreas de investigación (ver, por ejemplo, [Abbadeni, 1996, Tudhope and Taylor, 1996]). La variante utilizada en este trabajo enfatiza la naturaleza difusa de los datos codificados al introducir valores de semejanza parcial para las componentes "clase" y "lado" en las tercias difusas.

La nueva medida de semejanza se define como sigue:

Definición 3.5. La semejanza suave de Gower entre dos registros r_i y r_j es

$$\mathcal{S}(r_i, r_j) = \frac{1}{3} \cdot (1 - |\mu_i - \mu_j| + g_c(r_i, r_j) + g_s(r_i, r_j)) \quad (3.18)$$

donde

- μ es el valor de pertenencia de r , dado por la tripleta difusa $(clase, \mu, lado)$.
- $g_c(r_i, r_j)$ es la contribución a la semejanza entre los dos registros en relación a los conjuntos difusos a los que pertenecen. Este término se define como sigue:

$$g_c(r_i, r_j) = \begin{cases} 1 & r_i \in \alpha \text{ y } r_j \in \alpha \\ 0,5 & r_i \in \alpha, r_j \in \beta \text{ y } \mathcal{N}(\alpha, \beta) \\ 0 & \text{c.o.c.} \end{cases} \quad (3.19)$$

siendo α y β conjuntos difusos. De esta manera, la semejanza entre r_i y r_j , en relación al conjunto difuso a que pertenecen, es total ($g_c(r_i, r_j) = 1$) si ambos registros pertenecen al mismo conjunto. Esta semejanza es nula ($g_c(r_i, r_j) = 0$) si los registros pertenecen a conjuntos ajenos ($\alpha \cap \beta = \emptyset$). Finalmente, la semejanza con respecto al conjunto difuso es parcial ($g_c(r_i, r_j) = 0,5$) si los registros pertenecen a conjuntos vecinos.

- Y $g_s(r_i, r_j)$ es la semejanza entre los registros r_i y r_j , en términos al lado de función donde se encuentran y está dada por:

$$g_s(r_i, r_j) = \begin{cases} 1 & r_i \in \alpha, r_j \in \alpha \text{ y } S(r_i) = S(r_j) \\ 0,5 & r_i \in \alpha, r_j \in \beta, \mathcal{N}(\alpha, \beta), r_i \mapsto \beta \text{ y } r_j \mapsto \alpha \\ 0 & \text{c.o.c.} \end{cases} \quad (3.20)$$

siendo $S(r)$ una función que devuelve el lado de función del registro r . En este caso, la semejanza entre r_i y r_j , con respecto al lado de función donde se encuentran, es total ($g_s(r_i, r_j) = 1$) si ambos registros pertenecen al mismo conjunto y se encuentran al mismo lado de la función; es parcial ($g_s(r_i, r_j) = 0,5$) si se encuentran en conjuntos vecinos, cada registro colocado en el lado hacia donde se encuentra el conjunto al que pertenece el otro registro; la semejanza es nula en cualquier otro caso.

A partir de la ecuación 3.18 se deriva una medida de distancia entre dos tripletas cualitativas, de la siguiente manera:

Definición 3.6. La distancia suave de Gower entre dos registros r_i y r_j está dada por

$$\mathcal{D}(r_i, r_j) = \sqrt{1 - \mathcal{S}(r_i, r_j)} \quad (3.21)$$

3.5. Predicción de series de tiempo mediante conjuntos difusos dinámicos

El análisis de series de tiempo es un campo de aplicación conveniente para probar la eficacia de los conjuntos difusos dinámicos suaves como lenguaje de modelado de datos provenientes de sistemas dinámicos.

En esta sección, se presentan los resultados obtenidos en la predicción de series de tiempo usando un algoritmo de k vecinos próximos. Este método de predicción utiliza la estabilidad local de los datos para buscar los registros más cercanos al valor codificado actual de la variable.

3.5.1. Método de predicción difusa basado en series de tiempo

El método de predicción de series de tiempo empleado en esta sección calcula el nuevo valor de la variable a partir de los registros en la serie de tiempo cuya estabilidad local es la más parecida a la estabilidad local de la variable en el tiempo actual. Si $X(t_1), X(t_2), \dots, X(t_k)$ son los k registros codificados más cercanos al valor difuso actual $X(t)$, entonces la predicción de la estabilidad local en el paso de tiempo $t + 1$ se calcula por medio de una operación difusa de agregación sobre los registros codificados $X(t_1 + 1), X(t_2 + 1), \dots, X(t_k + 1)$.

Para medir la semejanza local del registro actual $X(t)$ a los registros previos en la serie de tiempo, se conforma un cúmulo de registros con los registros inmediatos anteriores en la serie:

$$C(X(t)) = (X(t), X(t-1), X(t-2), \dots, X(t-m-1)) \quad (3.22)$$

Siendo m la longitud del cúmulo, esto es, el número de registros considerados para medir la semejanza local.

A continuación se buscan los k cúmulos $C(X(t_1)), C(X(t_2)), \dots, C(X(t_k))$ más próximos y éstos son utilizados para calcular la predicción del valor de salida, para lo cual se calculan los contadores siguientes, para cada categoría difusa α :

$$I_\alpha = \frac{\sum_{j=1}^k \text{Sim}(C(X(t)), C(X(t_j))) \cdot V_j}{\sum_{j=1}^k \text{Sim}(C(X(t)), C(X(t_j)))} \quad (3.23)$$

donde $\text{Sim}(C(X(t)), C(X(t_j)))$ es la semejanza entre los cúmulos $C(X(t))$ y $C(X(t_j))$, dada por:

$$\text{Sim}(C(X(t)), C(X(t_j))) = \frac{1}{m} \sum_{i=0}^{m-1} \mathcal{S}(X(t-i), X(t_j-i)) \quad (3.24)$$

Si $X(t_j) = (clase_j, \mu_j, lado_j)$, entonces V_j se calcula como sigue:

$$V_j = \begin{cases} \mu_j & clase_j = \alpha \\ 1 - \mu_j & clase_j = \beta, \mathcal{N}(\beta, \alpha) \text{ y } X(t_j) \mapsto \alpha \\ 0 & \text{c.o.c.} \end{cases}$$

La predicción de la variable en el tiempo $t + 1$ está dada por

$$X(t+1) = (\alpha_{max}, I_{\alpha_{max}}, lado_{\alpha_{max}}) \quad (3.25)$$

siendo

- α_{max} el conjunto difuso cuyo contador acumuló el máximo valor en la agregación.

- $I_{\alpha_{max}}$ valor del contador para α_{max} , que se toma como el nuevo valor de pertenencia.
- $lado_{\alpha_{max}}$ el nuevo valor de lado de la función, el cual se toma como apuntando hacia el conjunto vecino del conjunto ganador que haya obtenido la puntuación más alta en la agregación.

Finalmente, el valor cualitativo se decodifica para obtener el valor cuantitativo de salida.

3.5.2. Estrategias

Con el fin verificar la utilidad de los conjuntos dinámicos suaves y de elegir la mejor opción, se probaron tres estrategias diferentes para elegir los parámetros de los conjuntos difusos:

- La *estrategia A* utiliza conjuntos difusos fijos, cuyo valor modal ν se calcula como el valor medio sobre los datos del conjunto de entrenamiento completo, mientras que su ancho de banda se calcula como la desviación estándar de los mismos datos:

$$\nu = \frac{1}{n} \sum_{j=1}^n x_j \quad \text{y} \quad \varpi = \sqrt{\frac{1}{n-1} \sum_{j=1}^n (x_j - \nu)^2}$$

donde $\{x_j : j = 1 \dots n\}$ es el conjunto de entrenamiento.

- En la *estrategia B* se emplea un conjunto difuso dinámico suave. Los valores de los parámetros del conjunto se calculan localmente como el valor medio y la desviación estándar de los 10 registros previos.

$$\nu = \frac{1}{w} \sum_{j=1}^w x_j \quad \text{y} \quad \varpi = 20 \cdot \sqrt{\frac{1}{w-1} \sum_{j=1}^w (x_j - \nu)^2}$$

donde $x_{t-j} = x(t - j\Delta t)$ es el valor de la variable j pasos de antes al tiempo actual t . w es el número de registros considerados para calcular localmente los parámetros. En las pruebas presentadas en esta sección, $w = 10$.

- La *estrategia C* es una variante de la estrategia B, obtenida usando el valor anterior en la serie de tiempo como valor modal para los conjuntos difusos ($\nu = x(t - \Delta t)$). El ancho de banda del conjunto difuso se calcula como la desviación estándar de los 10 registros previos.

3.5.3. Resultados

Las estrategias para la predicción de serie de tiempo descritas previamente fueron utilizadas para predecir las siguientes series de tiempo.

- Atractor de Lorenz. Para este problema, se generó un conjunto de 3000 registros mediante integración numérica de las ecuaciones 2.1, usando un método Runge-Kutta de cuarto orden. Esta serie se etiquetó como "Lorenz 1" en el cuadro 3.1. Siguiendo a la mayoría de los autores, los valores de los parámetros usados en este trabajo son $\sigma = 10$, $r = 28$ y $b = \frac{8}{3}$. La tasa de muestreo fue $\tau = 0,01$. En la prueba se realiza la predicción de la componente z del sistema de Lorenz.
- La serie de tiempo etiquetada como "Lorenz 2" en el cuadro 3.1 se generó agregando una tendencia de crecimiento lineal y una dispersión creciente lineal en la componente z del atractor de Lorenz. Esta serie de tiempo dada, entonces, por la ecuación:

$$\check{z}_k = \frac{k(1 + z_k)}{100}; \quad k = 1 \dots 3000 \quad (3.26)$$

- Una tercera serie de tiempo fue generada simulando una tendencia de crecimiento cuadrática y una dispersión creciente cuadrática en la componente z del atractor de Lorenz. La serie de tiempo se etiqueta como "Lorenz 3" en el cuadro 3.1, y está dada por la ecuación:

$$\tilde{z}_k = \frac{k^2(1 + z_k)}{50000}; \quad k = 1 \dots 3000 \quad (3.27)$$

- El número de manchas solares. Los datos usados en este trabajo corresponden al rango de observación del 1 de enero de 1947 al 12 de agosto de 1956.
- Mercado divisas. Las variables que se predecirán en este caso de estudio son los tipos de cambio, con respecto al dólar estadounidense, de las siguientes monedas: la libra esterlina, el dólar canadiense, el marco alemán, el yen japonés y el franco suizo. Los datos usados en este problema cubren el período del 1 de junio de 1973 al 21 de mayo de 1987.
- Los datos trimestrales de la producción australiana de electricidad, en millones de kWh, de marzo de 1956 a septiembre de 1994.

Serie de tiempo	Estrategia		
	A	B	C
Lorenz 1	0.1345930	0.1627156	0.0336144
Lorenz 2: lineal	0.9570831	0.1449291	0.0323558
Lorenz 3: cuadratica	1.5498921	0.1217218	0.0225338
Número de manchas solares	0.2701580	0.2950078	0.2448777
Libra esterlina	2.2653646	0.1021100	0.1035266
Dólar canadiense	0.9332337	0.0658772	0.0692485
Marco alemán	0.6274184	0.0428661	0.0433149
Yen Japonés	0.0832565	0.0617559	0.0635505
Franco Suizo	0.9886812	1.0522985	0.9332467
Producción australiana de electricidad	2.0897527	0.3217207	0.4471190
NMSE promedio	0.9899433	0.2371003	0.1993388

Cuadro 3.1: Error NMSE al predecir diferentes series de tiempo.

El cuadro 3.1 presenta la comparación en el desempeño del algoritmo de predicción al utilizar las 3 estrategias para la representación difusa de los datos. La comparación se hace a través del error cuadrático medio normalizado (NMSE, por las siglas en inglés) obtenido al predecir cada serie de tiempo. También se presenta el error NMSE promedio de todas las pruebas realizadas para cada estrategia. El NMSE se define como

$$NMSE = \frac{\sqrt{\frac{1}{n-1} \sum_{i=0}^n (x_i - \hat{x}_i)^2}}{\sqrt{\frac{1}{n-1} \sum_{i=0}^n (x_i - \bar{x})^2}} \quad (3.28)$$

donde \hat{x}_i es la predicción de x_i , y \bar{x} el valor medio de x_i .

Como puede observarse, la estrategia de usar los conjuntos difusos dinámicos suaves referidos al valor anterior (estrategia C) muestra un mejor desempeño, en todos los casos, que el obtenido al usar los conjuntos difusos fijos (estrategia A). Esta superioridad es más notable para las series de tiempo que muestra tendencias claramente crecientes o decrecientes, como se puede apreciar de los resultados obtenidos con las 3 series de tiempo del sistema de Lorenz, la producción australiana de electricidad, la libra esterlina o el dólar canadiense. Aunque en 5 casos, usar el valor medio local como valor modal de los conjuntos difusos arroja un NMSE más bajo, en 3 casos sus resultados son incluso peores que al emplear conjuntos difusos fijos.

De esta manera, en el presente trabajo se emplean conjuntos difusos dinámicos suaves cuyo valor modal es el valor inmediato anterior en la serie de tiempo.

Capítulo 4

Plataforma multiagente

Los sistemas multiagente ofrecen una plataforma adecuada para la integración de técnicas diversas en una misma aplicación [Plaza and Ontañón, 2003]. En este trabajo, se ha utilizado un sistema multiagente como base para la integración de técnicas de simulación diferentes en un sistema de modelado híbrido inteligente de sistemas dinámicos. La estrategia a seguir consiste en aprovechar las características de los agentes para encapsular modelos diferentes. La solución a un problema dado se obtiene como resultado de la colaboración entre los individuos del sistema multiagente.

En este capítulo, se presenta la arquitectura multiagente específica utilizada en este trabajo, así como un mecanismo de agregación difusa de las opiniones individuales.

4.1. Marco organizacional

La arquitectura SMA utilizada para el modelado híbrido inteligente de sistemas dinámicos se basa en el modelo Aalaadin para sistemas multiagente [Gutknecht and Ferber, 1997] (véase la figura 4.1). Tal modelo proporciona dos características apropiadas para el desarrollo de la metodología de simulación propuesta: un modelo genérico de agente, que no hace ninguna suposición ni impone restricciones sobre su configuración interna, y un marco de organización basado en los conceptos de agente, grupo y rol. Estas características ofrecen una gran libertad en el diseño de los agentes, fijando la atención en las técnicas de predicción que se emplearán. Por otra parte, el esquema organizacional simplifica la estructuración del SMA y ofrece mecanismos para administración del ciclo de vida de agentes y grupos, y para el manejo de paso de mensajes.

Los agentes se encuentran organizados en los grupos de modelado, coordinación e interfaz, siendo los principales roles de agente los de predictor, árbitro local, planificador de trabajos

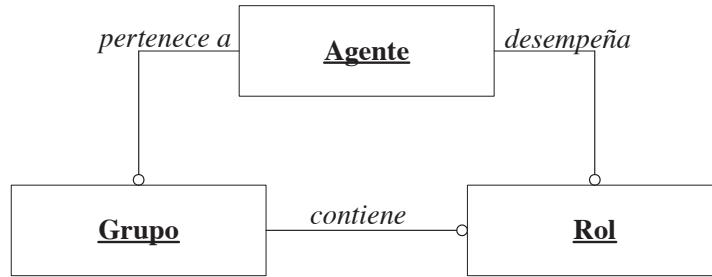


Figura 4.1: Modelo Aalaadin para sistemas multiagente.
(tomado de [Gutknecht and Ferber, 1997])

(*scheduler*), árbitro central, facilitador de recursos (*broker*), y asistente de usuario (véase la figura 4.2). Estos elementos se integran en un modelo SMA especializado, definido como sigue:

Definición 4.1. Un SMA para modelado híbrido inteligente de sistemas dinámicos es una tripleta $\langle M, Cg, Ig, R \rangle$, donde

- M es el conjunto de grupos de modelado.
- Cg es el grupo de coordinación.
- Ig es el grupo de interfaz.
- R es el conjunto de reglas sociales que gobiernan las interacciones del SMA.

Esta estructura imita una organización conformada por un conjunto de equipos de trabajo especializados, cada uno de ellos responsables de tareas específicas de modelado, coordinación e interfaz.

4.1.1. Grupo de modelado

El grupo de modelado es la base para la distribución de tareas. Esta clase de grupo es la entidad responsable de predecir el comportamiento de un subsistema dado. Cada grupo de modelado consta de un conjunto de predictores y un árbitro local. La tarea asignada a cada grupo de modelado es la predicción del comportamiento de una serie de tiempo.

El predictor es un agente especializado provisto de una técnica de simulación de sistemas dinámicos. En este trabajo, estas técnicas se utilizan para predecir el nuevo valor de una variable dada, basada en la serie de tiempo de la variable y las series de tiempo de otras

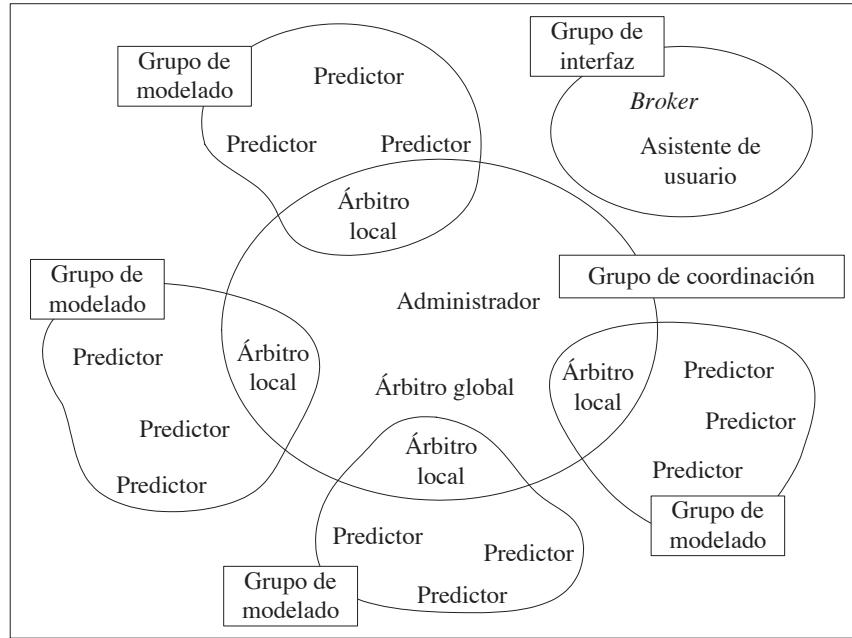


Figura 4.2: Estructura organizacional de un SMA para modelado híbrido inteligente.

variables de entrada al sistema. En la figura 4.3 se muestra el diagrama de estados que describe el comportamiento del agente predictor. La acción del agente predictor comienza cuando envía un mensaje con la performativa "Listo" al árbitro local de su grupo. A continuación, el agente predictor entra en un ciclo de operación en espera de una solicitud de predicción. En la solicitud de predicción se especifica el tiempo para el cual se deberá predecir el valor de la variable t , así como el rango de registros (R) a considerar como base de conocimiento. El agente realiza una serie de solicitudes de los datos de entrada necesarios para la predicción. En caso de alcanzar un número máximo de intentos (MI) sin lograr obtener los datos requeridos, el agente realiza la predicción con los datos disponibles. En cualquier caso, el predictor envía su resultado al árbitro local (acompañada de una medida de confianza). Una vez hecha la predicción, el agente queda en espera de otra solicitud. El ciclo de vida del predictor termina cuando el sistema es finalizado, o cuando el agente es eliminado del grupo de modelado.

Varias *especies* de predictor cohabitan en el mismo grupo de modelado, cada una de ellas caracterizada por la técnica de simulación usada por sus individuos. Por otra parte, los individuos de una especie determinada se diferencian entre sí por los datos que utilizan en su entrenamiento y el modelo específico usado para simular un sistema dinámico. Tal diversidad de predictores permite integrar y explotar las ventajas de las diferentes técnicas de simulación

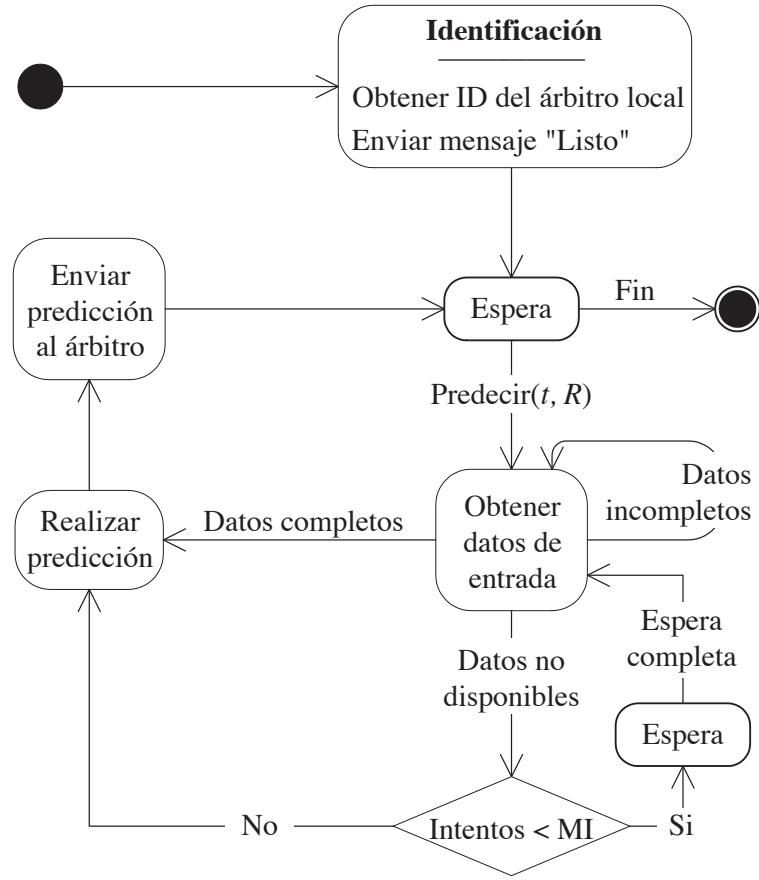


Figura 4.3: Diagrama de estados del agente predictor.

disponibles. De esta manera, el conjunto de agentes predictores en cada grupo de modelado emula a un equipo de especialistas, provistos con mecanismos de razonamiento diversos y cuyos miembros tienen experiencias diferentes.

El árbitro local es responsable de coordinar el proceso de predicción, y de calcular y publicar el valor global de salida al interior de cada grupo de modelado, basado en los resultados individuales de los predictores. El diagrama de estados del árbitro local se muestra en la figura 4.4. Su ciclo de vida inicia cuando envía un mensaje con la performativa "Listo" al árbitro global, quedando a continuación en espera de una solicitud de predicción. En la solicitud, nuevamente se incluyen los valores t y R que el árbitro local deberá transmitir a lo predictores en su grupo, al solicitarles su "opinión" sobre la predicción solicitada por el árbitro global. El árbitro local realiza una serie de intentos por recabar las predicciones individuales. Si después de un cierto número máximo de intentos no logra obtener todas las predicciones individuales, realiza la predicción global con los datos disponibles. Finalmente,

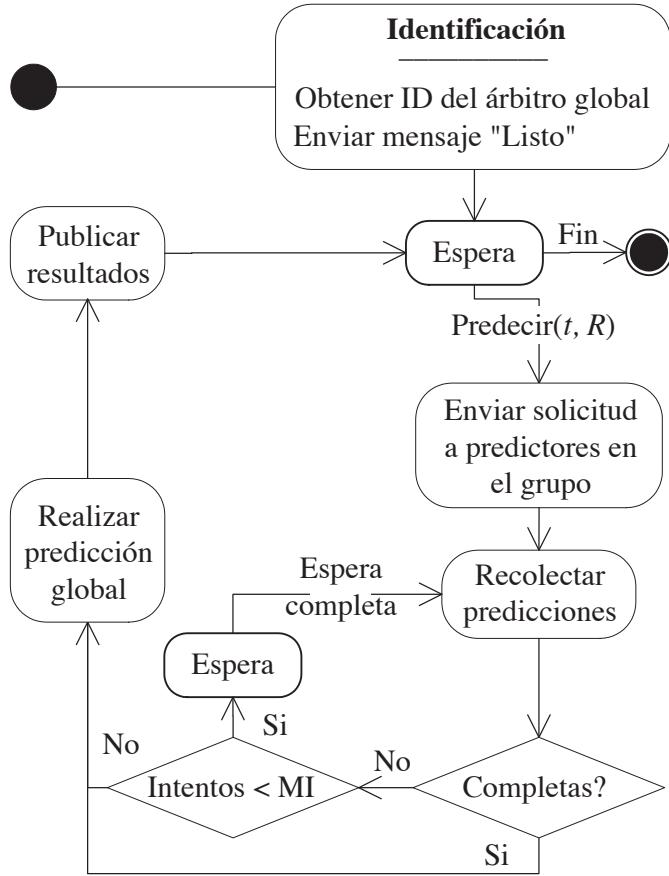


Figura 4.4: Diagrama de estados del árbitro local.

publica los resultados y queda en espera de una nueva solicitud.

Los árbitros locales no distinguen entre las diferentes especies de predictores, para la predicción colectiva (aunque tal información está disponible para el árbitro).

Definición 4.2. Un *grupo de modelado* se define como una 5-tupla $M_i = \langle P_i, mT_i, R_i, V_i, Ref_i \rangle$, donde

- P_i es el conjunto de agentes predictores.
- mT_i es el conjunto de técnicas de modelado usadas en el grupo.
- $R_i : P_i \rightarrow mT_i$ es una relación que describe la técnica de modelado usada por cada agente predictor.
- V_i es la variable que se predecirá por el grupo de modelado.

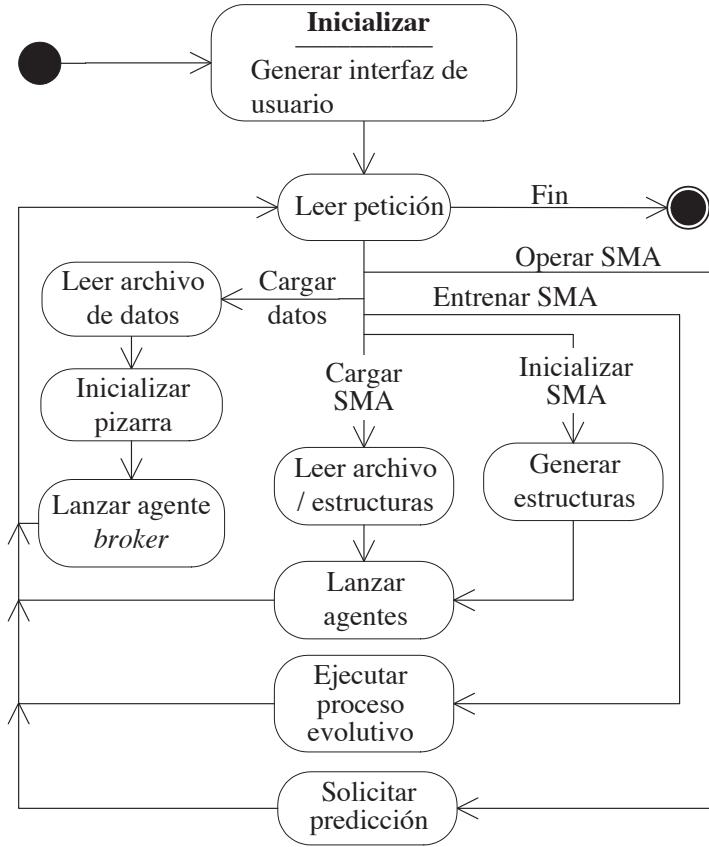


Figura 4.5: Diagrama de estados del *scheduler*.

- Ref_i es el agente árbitro local.

4.1.2. Grupo de coordinación

El grupo de coordinación incluye a todos los árbitros locales, al agente *scheduler* y al árbitro central. El *scheduler* es el primer agente generado en el sistema. Su función consiste en generar todos los demás agentes en el SMA, controlar las etapas de entrenamiento y la actualización de la población en cada grupo de modelado. La figura 4.6 muestra el diagrama de interacción durante la etapa de creación de SMA.

El árbitro central es responsable de coordinar el proceso de la predicción de todos los grupos de modelado en el sistema y de publicar los valores finales de cada variable de salida (la figura 4.7 muestra el diagrama de interacción durante la etapa de la operación de SMA).

Definición 4.3. Un *grupo de coordinación* es una cuádrupla $Cg = < L, C, S, V >$, donde

- L es el conjunto de árbitros locales.

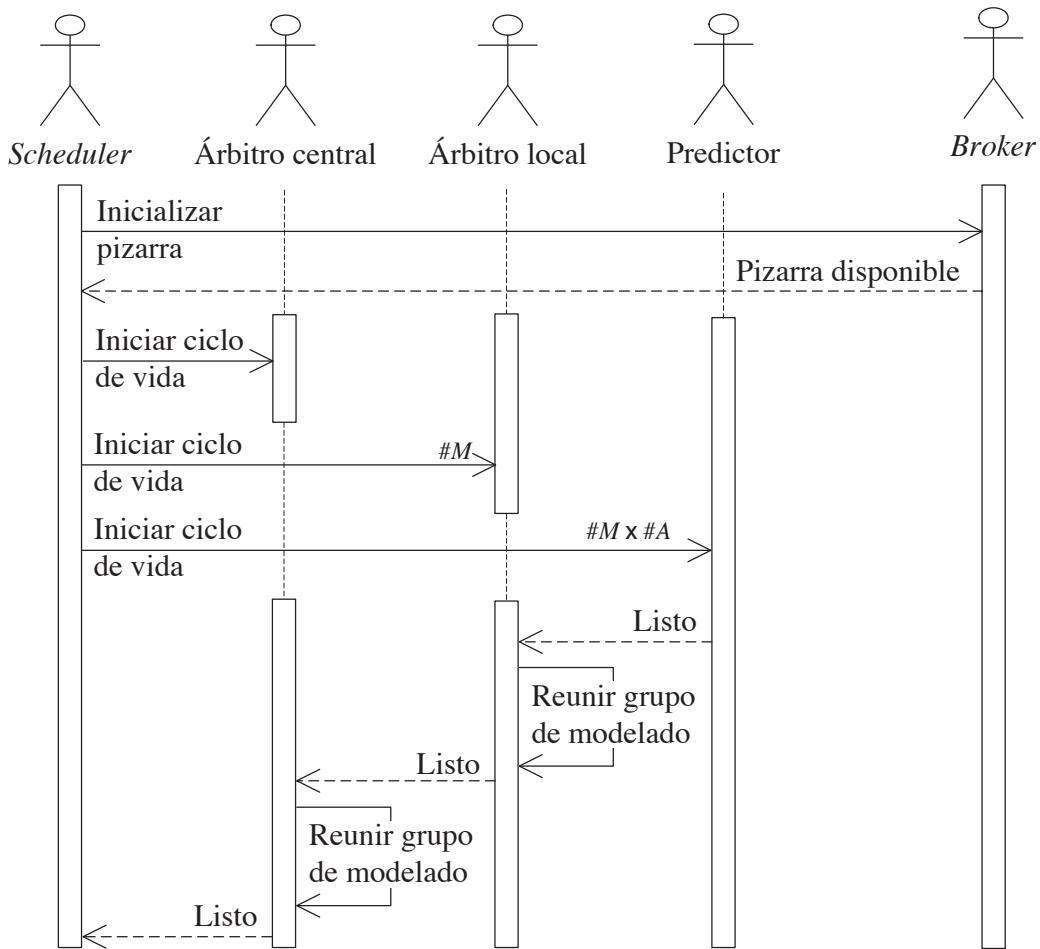


Figura 4.6: Diagrama de interacción para la etapa de inicialización del SMA.

- C es el árbitro central.
- S es el agente *scheduler*.
- V es el conjunto de variables que serán modeladas por el SMA.

4.1.3. Grupo de interfaz

El grupo de interfaz está formado por un conjunto de agentes asistentes de usuario y un conjunto de agentes *broker*. La función de los asistentes de usuario consiste en la obtener las peticiones y las restricciones impuestas por los usuarios, y transferir esta información al grupo de coordinación. Por otra parte, los agentes *broker* son responsables de recopilar y de archivar toda la información disponible sobre el sistema dinámico. El intercambio de información entre

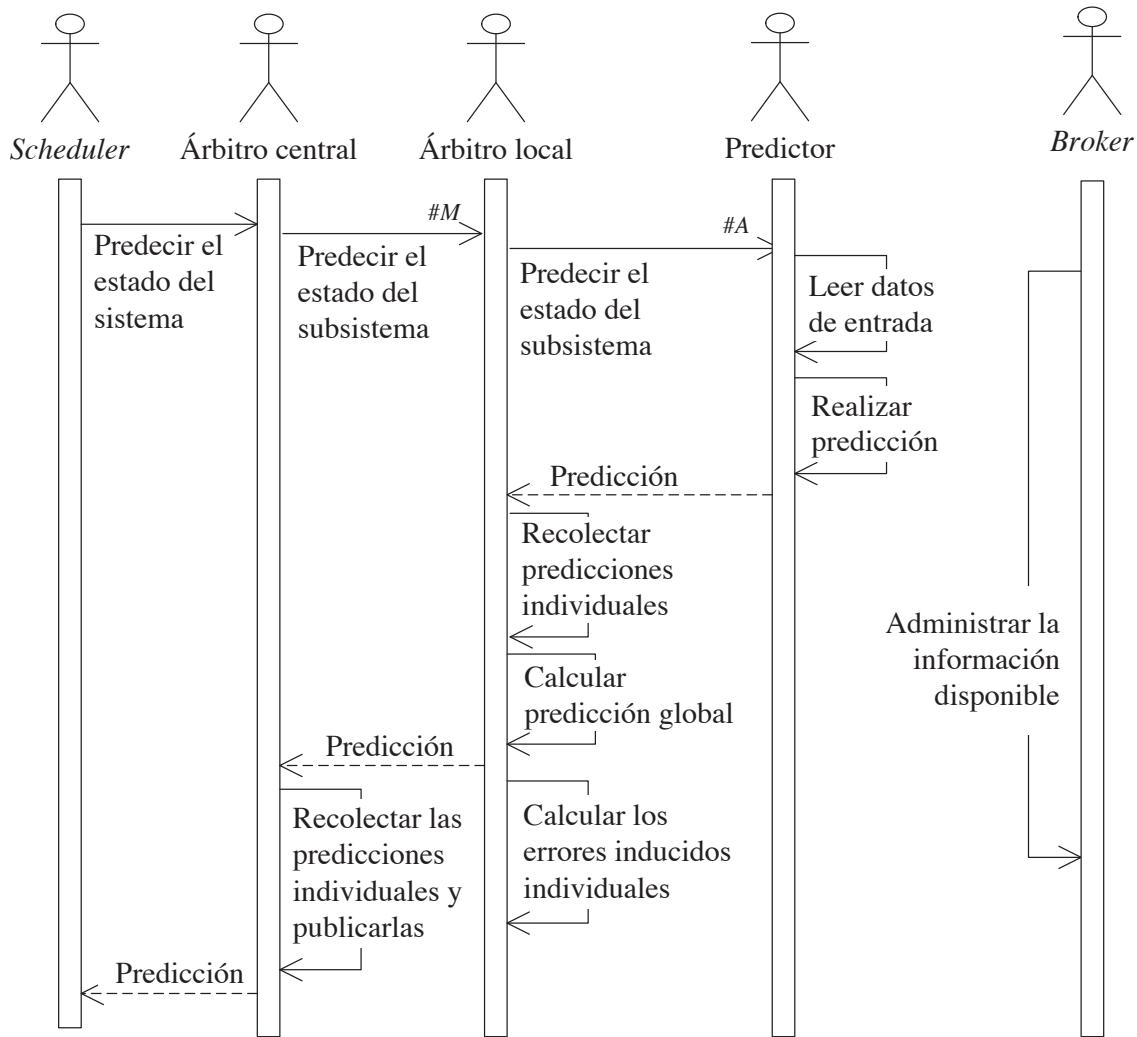


Figura 4.7: Diagrama de interacción para la etapa de operación del SMA.

los agentes se realiza por medio de los mecanismos siguientes: 1) acceso directo del *scheduler* a los métodos internos de los agentes, usados principalmente para conducir el proceso genético; 2) paso de mensaje entre el árbitro central y los árbitros locales, y entre los árbitros locales y los agentes predictor en cada grupo de modelado, usado para transferir peticiones y resultados del pronóstico; y 3) una pizarra global, usada por los predictores para obtener los datos de entrada de información.

Definición 4.4. Un *grupo de interfaz* es un cuádrupla $Ig = < I, W, DB, Bb >$, donde

- I es el conjunto de agentes de interfaz.
- W es el conjunto de agentes *broker*.

- Db es la base de conocimiento.
- Bb es la pizarra.

4.1.4. Modelado multiagente

Cada grupo de predictores en un sistema multiagente para modelado híbrido inteligente juega el papel de una ecuación diferencial, en el sentido de que su responsabilidad es la predicción del comportamiento de una variable. Sin embargo, contrariamente a la aproximación estándar en ecuaciones diferenciales, en los grupos de modelado cohabitan varios modelos diferentes o complementarios para cada variable.

El uso de un sistema multiagente para realizar la simulación de un sistema dinámico, permite realizar la predicción simultánea de varias variables de salida, siendo posible hacer la simulación de un sistema dinámico por medio de un conjunto de grupos de modelado acoplados, que serían el equivalente cualitativo a un sistema de ecuaciones diferenciales acopladas. Otra ventaja que proporciona el uso de un SMA es la posibilidad natural de combinar técnicas diferentes para predecir una misma variable de salida. También, es posible sustituir, de una manera simple, aquellos modelos que han dejado de ser útiles.

A diferencia del modelado basado en ecuaciones diferenciales, en un SMA para modelado híbrido inteligente los "términos" en estas especies de ecuaciones diferenciales son variables incluso en número. De esta manera un sistema de modelado híbrido inteligente representaría una especie de modelo autoajustable.

4.2. Agregación difusa para la predicción colectiva

El tema más importante en el desarrollo de sistemas multiagente es la forma de garantizar una adecuada interacción entre los individuos del sistema. Sin embargo, y a pesar del esfuerzo realizado en el área, las técnicas utilizadas para implementar la interacción entre agentes siguen siendo muy rígidas [Romero Cortés and Sheremetov, 2001]. Cuando los agentes son clasificadores cuya tarea es obtener una solución colectiva, el problema consiste en como combinar los clasificadores [Duin, 2002, Yang and Zou, 2004].

La metodología de modelado híbrido inteligente emplea un mecanismo de agregación de las opiniones individuales de los agentes que integra varios elementos de las técnicas más avanzadas para combinación de clasificadores. Por otro lado, este mecanismo aprovecha las características de la estrategia de codificación difusa y el algoritmo de programación genética

del SMA para mantener una estructura tan simple como la de los métodos de agregación más simples.

4.2.1. Métodos de fusión de datos

La fusión de datos (FD) es una forma de procesamiento de datos cuyo objetivo es la obtención de decisiones basada en fuentes de información distribuidas [Gorodetski et al., 2002]. Los principales métodos para combinar decisiones a partir de clasificadores diversos son: 1) algoritmos de votación, 2) algoritmos difusos o probabilísticas, 3) algoritmos de meta-aprendizaje basados en generalización apilada y 4) algoritmos de meta-aprendizaje basados en competencia [Gorodetski et al., 2002]. Los métodos más simples son los de votación, mientras que los más complejos son los de meta-aprendizaje.

Los métodos de generalización apilada utilizan conocimiento acerca de los clasificadores básicos para integrar las opiniones individuales, con el inconveniente de que no pueden garantizar que se mantenga la respuesta del sistema cuando se agrega (o reemplaza) un nuevo clasificador [Gorodetski et al., 2002].

Los métodos de combinación de clasificadores basados en la competencia asignan a cada clasificador particular una región en el espacio de atributos donde es más competente [Duin, 2002, Gorodetski et al., 2002, Gorodetski et al., 2003]. La principal responsabilidad del árbitro es determinar la competencia de cada clasificador en el sistema, de acuerdo a los datos de entrada [Duin, 2002, Gorodetski et al., 2002, Ortega et al., 2001]. La combinación se obtiene mediante la selección de la clasificación producida por el agente más competente.

4.2.2. Método híbrido de agregación

La predicción del comportamiento de un sistema dinámico, en el presente trabajo, se basa en la agregación de la opinión individual dada por los diversos agentes que conforman el sistema multiagente.

El resultado global en cada grupo de modelado, es decir, el nuevo valor de la variable de salida, se determina por medio de una operación difusa de agregación que integra los resultados individuales obtenidos por los diversos agentes. Aunque el árbitro local no hace ninguna distinción en la técnica de la predicción usada por los agentes, las opiniones individuales desempeñan diversos papeles, en un sentido conceptual, para la predicción colectiva, dependiendo de la técnica de modelado que fue utilizada por el agente: algunos agentes se entrenan para dar una predicción independiente, mientras que las predicciones de otros agentes tienen

sentido solamente en un contexto colectivo.

Cada agente expresa su voto por un conjunto difuso específico y ofrece una medida de confianza sobre su resultado. Si el predictor no cuenta con un mecanismo para medir la confianza sobre su propia predicción, envía un valor por defecto, específico para cada especie. Por otra parte, cada voto tiene un valor proporcional al valor correspondiente de pertenencia. El valor de lado de función asociado a la codificación difusa se toma como apuntando al conjunto vecino (en caso de que haya alguno) y a esta señal se toma como una segunda opción. El peso de este segundo voto se calcula como el complemento del valor de pertenencia que corresponde a la opción principal.

La operación de agregación es realizada mediante un proceso de votación ponderada con opción múltiple. Para calcular los votos en este proceso, se calculan los contadores siguientes para cada clase α_i :

$$I_{\alpha_i} = \frac{\sum_{j=1}^{N_G} W_j \cdot V_j}{\sum_{j=1}^{N_G} W_j} \quad (4.1)$$

donde N_G es el número de agentes que han emitido un voto en el grupo de modelado G ; W_j es el peso dado a la opinión del agente A_j en función de su desempeño y está dado por

$$W_j = 2 \left(\frac{1}{1 + e^{-5\Omega_j^2}} - \frac{1}{2} \right) \cdot (1 - \bar{E}_j) \quad (4.2)$$

En esta expresión, Ω_j es la medida de confianza en la predicción dada por el agente A_j . De esta manera, el término

$$2 \left(\frac{1}{1 + e^{-5\Omega_j^2}} - \frac{1}{2} \right)$$

asigna un peso sigmoideo a la predicción del agente, dependiendo de la medida proporcionada de confianza. \bar{E}_j es una medida del error cuadrático normalizado medido sobre el desempeño histórico del agente y está dado por

$$\bar{E}_j = \frac{1}{N_T} \sum_{n=1}^{N_T} \min \left(\left| \frac{O_n - \hat{O}_{nj}}{d} \right|, 1 \right) \cdot \xi_n^j \quad (4.3)$$

donde N_T es el número de tareas de predicción en las cuales el agente ha participado. O_n es el valor objetivo en la n -esima tarea de predicción y \hat{O}_{nj} la predicción correspondiente hecha por el agente A_j . d se define como sigue:

$$d = \begin{cases} O_n & \text{Si } O_n \neq 0 \\ 1 & \text{c.o.c.} \end{cases} \quad (4.4)$$

El término ξ_n^j en la ecuación 4.3 es una medida del error que el agente ha inducido en la predicción colectiva y está dado por

$$\xi_n^j = \min \left(\left| \frac{O_n - \hat{O}_n}{d} \right|, 1 \right) - \min \left(\left| \frac{O_n - \hat{O}_n^{(-j)}}{d} \right|, 1 \right) \quad (4.5)$$

donde \hat{O}_n es la predicción colectiva y $\hat{O}_n^{(-j)}$ es la predicción colectiva que se obtendría al eliminar la opinión del agente A_j . Si $\xi_n^j > 0$ entonces la opinión del agente A_j incrementa el error colectivo, y por lo tanto, su error histórico se incrementa, en caso contrario, el error histórico se reduce.

V_j es el voto del agente A_j . Si $O_{nj} = (clase_j, \mu_j, lado_j)$, entonces V_j se calcula según la relación siguiente:

$$V_j = \begin{cases} \mu_j & clase_j = \alpha_i \\ 1 - \mu_j & clase_j = \beta, \mathcal{N}(\beta, \alpha_i) \text{ y } \hat{O}_{nj} \mapsto \alpha_i \\ 0 & \text{c.o.c.} \end{cases} \quad (4.6)$$

La predicción de la variable en el tiempo $t + 1$ está dada por

$$X(t + 1) = (\alpha_{max}, I_{\alpha_{max}}, lado_{\alpha_{max}}) \quad (4.7)$$

siendo

- α_{max} el conjunto difuso cuyo contador acumuló el máximo valor en la agregación.
- $I_{\alpha_{max}}$ valor del contador para α_{max} , que se toma como el nuevo valor de pertenencia.
- $lado_{\alpha_{max}}$ el nuevo valor de lado de la función, el cual se toma como apuntando hacia el conjunto vecino del conjunto ganador que haya obtenido la puntuación más alta en la agregación.

4.2.3. Efecto de filtrado en la agregación difusa

El algoritmo de codificación difusa utilizado en este trabajo realiza un mapeo de valores cuantitativos $x(t_k) \in \mathcal{X} \subseteq (-\infty, +\infty)$ en la serie de tiempo a valores difusos $X(t_k) = (\alpha, \mu, lado)$ tal que $\alpha \in \{baja, estable, alza\}$, $\mu \in [0, 1]$, $lado \in \{izquierda, centro, derecha\}$. La decodificación, por otro lado, convierte un valor difuso $X(t_k)$ al valor cuantitativo $x(t_k) \in [x(t_{k-1}) - 2\varpi_{k,w}, x(t_{k-1}) + 2\varpi_{k,w}]$, siendo $\varpi_{k,w}$ el ancho de banda del conjunto difuso, calculado sobre el subconjunto de los w valores previos a x_k en la serie de tiempo (véase la sección 3.2). Este método de codificación-decodificación ocasiona una pérdida de información, como

puede ser observado en el cuadro 4.1, donde se presentan los errores NMSE entre los datos decodificados y los originales para las series de tiempo del capítulo 3.

Serie de tiempo	Error
Lorenz 1	0.066659745
Lorenz 2: lineal	0.034183831
Lorenz 3: cuadrática	0.027385146
Número de manchas solares	0.172133517
Libra esterlina	0.149439043
Dólar canadiense	0.017037359
Marco alemán	0.180704628
Yen japonés	0.146950901
Franco suizo	0.146189558
Producción australiana de electricidad	0.087247378
NMSE promedio	0.102793111

Cuadro 4.1: Error en el proceso de fuzificación-defuzificación de diferentes series de tiempo.

El error que se presenta durante el proceso de codificación-decodificación se debe, parcialmente, a que el valor $x(t_k)$ se decodifique con base en los valores $x(t_{k-1})$ y $\varpi_{k,w}$, con lo cual el nuevo valor no puede alejarse demasiado del valor previo. Esta característica se utiliza ventajosamente durante el proceso de agregación. El árbitro local recibe, de parte de los agentes predictores, las predicciones individuales acerca del cambio que debe experimentar el valor de la variable, con respecto a su valor actual y a los cambios recientes en los datos. El árbitro realiza la agregación difusa de estas predicciones, la cual es tomada como predicción global. Esta predicción es decodificada, tomando como referencia los valores conocidos de $x(t_{k-1})$ y $\varpi_{k,w}$. Esta manera de realizar la integración de las opiniones individuales tiene dos consecuencias: 1) una predicción cuantitativo alejada del valor actual se limita al rango cualitativo $[(baja, 1, 0, izquierda), (alza, 1, 0, derecha)]$, reduciendo de esta manera el impacto de tal predicción en el resultado global, 2) una predicción localmente correcta siempre se integra adecuadamente a la predicción colectiva.

El comportamiento del proceso de agregación se muestra en el cuadro 4.2, donde se presentan los errores NMSE al agregar cuatro predicciones individuales simuladas para varias series de tiempo (las del capítulo 3). Las predicciones simuladas $\tilde{x}_i(t_k)$ son generadas artificialmente a partir del valor objetivo $x(t_k)$, como sigue:

- Predicción A: $\tilde{x}_A(t_k) = 2x(t_k)$
- Predicción B: $\tilde{x}_B(t_k) = x(t_k) + \bar{x}$

Serie de tiempo	NMSE en la predicción					
	A	B	C	D	Promedio	Aggregación
Lorenz 1	4.108199	3.970541	13.298844	2.201878	5.1806636	0.1328268
Lorenz 2: lineal	1.876813	1.582948	7.931265	1.313573	2.8137433	0.0465090
Lorenz 3: quadratica	1.464046	1.062508	7.900963	1.314045	2.5685649	0.0361280
Número de manchas solares						
	1.674592	1.283782	8.763070	1.455456	2.7412663	0.2374062
Libra esterlina	5.076301	4.973223	11.997946	1.992912	5.2910872	0.1526566
Dólar canadiense	8.744220	8.686305	6.090641	1.029232	5.6932689	0.0205275
Marco alemán	6.439682	6.359468	14.831155	2.494854	6.6730499	0.1836752
Yen Japonés	5.256278	5.159441	15.068286	2.481897	6.2079620	0.1487024
Franco suizo	5.105934	5.005592	12.606105	2.109362	5.5416595	0.1489003
Producción australiana de electricidad	2.080282	1.833439	5.849628	0.974199	2.4100382	0.1602712
NMSE						
Promedio	4.182635	3.991725	10.433790	1.736741	4.5121304	0.1267603

Cuadro 4.2: Error al agregar predicciones erroneas de diferentes series de tiempo.

- Predicción C: $\tilde{x}_C(t_k) = x(t_k) + \frac{3R}{N}t$, siendo R el rango de valores de los datos y N el número de predicciones.
- Predicción D: $\tilde{x}_D(t_k) = \rho \cdot x(t_k)$, siendo $\rho \in [-0.5, 0.5]$ un valor generado al azar usando una distribución de probabilidad uniforme.

Las predicciones A, B, C y D, que evidentemente son erróneas, se utilizan para calcular dos valores globales: el primero (etiquetado como **promedio** en el cuadro 4.2) es el valor promedio de las predicciones cuantitativas individuales; el segundo valor (etiquetado como **Aggregación**) es generado usando el método de agregación descrito anteriormente, sobre los valores difusos. Como puede apreciarse, el error es considerablemente superior en el promedio cuantitativo al obtenido mediante la agregación difusa (11.5 veces en el mejor de los casos,

para el número de manchas solares y 277 veces para el peor caso, el tipo de cambio del dólar canadiense).

De este experimento se desprende que el proceso difuso de agregación puede ser útil incluso para integrar los resultados de predicción obtenidos con técnicas cuantitativas, aunque aún se requiere más trabajo en esta dirección para precisar el grado de dicha utilidad.

Capítulo 5

Técnicas de razonamiento

La simulación de un sistema dinámico, en el presente trabajo, es realizada mediante modelos generados directamente a partir de observaciones de su comportamiento previo, registrado en un conjunto de series de tiempo. Las técnicas específicas empleadas en este trabajo son:

- Una técnica de modelado difuso llamada razonamiento difuso basado en series de tiempo, que será denotada como RDST.
- Una red neuronal *feedforward* entrenada mediante el algoritmo *backpropagation* estándar, que en lo subsecuente será denotada por RNA.
- Una red neuronal *feedforward* entrenada mediante una variante difusa del algoritmo *backpropagation* estándar, que será denotada por RNAD.

En este capítulo se introducen las técnicas RDST y RNAD, y se presenta la técnica RNA, así como la forma de integrarlas en el SMA.

5.1. Razonamiento difuso basado en series de tiempo

La primera técnica de modelado que se discutirá es una técnica cualitativa que se ha designado como razonamiento difuso basado en series de tiempo (RDST). Esta técnica emplea una aproximación de razonamiento basado en casos, es decir, la salida correspondiente a una cierta entrada de información dada se calcula por medio de una comparación con los registros más similares en una base de casos.

La técnica RDST fue inspirada en la técnica de razonamiento inductivo difuso (FIR - fuzzy inductive reasoning) [Cellier, 1991, de Albornoz, 1996]. Como en FIR, la técnica de

razonamiento RDST opera sobre datos cualitativos dados como tercias formadas por un valor cualitativo de clase, un valor numérico de pertenencia y un valor cualitativo de lado de función (véase la figura 3.4). La codificación difusa se realiza utilizando tres conjuntos difusos dinámicos [Kosanović et al., 1996, Soto and Núñez, 2003], que se han designado como "baja", "estable" y "alza", como se muestra en la figura 3.4. El centro del conjunto "estable" se toma como el valor anterior de la variable en la serie de tiempo. Estos conjuntos describen el grado de estabilidad local de la variable, en relación a su valor anterior.

Un modelo RDST establece una relación entre variables cualitativas. La relación en la ecuación 5.1, por ejemplo, indica que el valor de la variable y_1 en el tiempo actual t es determinado por el valor de la variable u_1 en el mismo tiempo (t), los valores de u_2 y y_1 una unidad de tiempo atrás ($t - \delta t$) y el valor de y_1 dos unidades de tiempo antes ($t - 2\delta t$):

$$y_1(t) = \tilde{F}(u_1(t), u_2(t - \delta t), y_1(t - \delta t), y_3(t - 2\delta t)) \equiv \tilde{F}(t) \quad (5.1)$$

donde δt es el intervalo de muestreo. El grupo de variables $C(t) = u_1(t), u_2(t - \delta t), y_1(t - \delta t), y_3(t - 2\delta t)$, que se designará en adelante como un *cúmulo de registros*, es la información de entrada al modelo $\tilde{F}(t)$, y $y_1(t)$ es la variable de salida.

Si el sistema dinámico es determinista, es de esperar que antecedentes similares conduzcan a estados consecuentes semejantes. La predicción de la variable de salida al tiempo t es realizada, entonces, buscando un cúmulo de registros que correspondan con los datos de entrada en el modelo. En el ejemplo dado en la ecuación 5.1, $y_1(t)$ es obtenido buscando un cúmulo de registros tales que $u_1(t) = u_1(t')$, $u_2(t - \delta t) = u_2(t' - \delta t)$, $y_1(t - \delta t) = y_1(t' - \delta t)$ y $y_3(t - 2\delta t) = y_3(t' - 2\delta t)$. Si existe tal cúmulo de registros, entonces el valor de la variable de salida en el tiempo actual se asume será igual que el de la variable correspondiente en el cúmulo, en el ejemplo, $y_1(t) = y_1(t')$. Debido a que en problemas reales, tales coincidencias ocurren raramente, se buscan los k -cúmulos más próximos y éstos son utilizados para calcular la predicción del valor de salida.

La proximidad entre dos tercias cualitativas se mide mediante el coeficiente de semejanza suave de Gower (ecuación 3.18).

La predicción final del agente se obtienen mediante un proceso de agregación similar al usado en el método de predicción presentado en la sección 3.5. Nuevamente, se calculan los contadores siguientes para cada clase $\alpha_i \in \{baja, estable, alza\}$:

$$I_{\alpha_i} = \frac{\sum_{j=1}^k \text{Sim}(C(t_j), C(t)) \cdot V_j}{\sum_{j=1}^k \text{Sim}(C(t_j), C(t))} \quad (5.2)$$

donde

$$\text{Sim}(C(t_j), C(t)) = \frac{1}{m} \sum_{i=0}^m \mathcal{S}(v(t_j), v(t)) \quad (5.3)$$

es la semejanza entre el cúmulo de registros $C(t_j)$ correspondiente al j -ésimo vecino y el cúmulo $C(t)$ que es la entrada actual al modelo, ambos conformados por m variables, y es empleada aquí como valor de ponderación; $\mathcal{S}(v(t_j), v(t))$ es la semejanza suave de Gower entre las variables $v(t_j)$ y $v(t)$. V_j es el voto del j -ésimo cúmulo vecino próximo; Si $y_j(t) = (clase_j, \mu_j, lado_j)$ es la salida dado el cúmulo $C(t_j)$, entonces V_j se calcula (empleando las definiciones 3.3 y 3.4), como:

$$V_j = \begin{cases} \mu_j & clase_j = \alpha_i \\ 1 - \mu_j & clase_j = \beta, \mathcal{N}(\beta, \alpha_i) \text{ y } y_j(t) \mapsto \alpha_i \\ 0 & \text{c.o.c.} \end{cases} \quad (5.4)$$

La interpretación final de los contadores es como en la ecuación 3.25 y se repite aquí por comodidad.

La predicción de la variable de salida $y(t)$ está dada por

$$y(t) = (\alpha_{max}, I_{\alpha_{max}}, lado_{\alpha_{max}})$$

siendo

- α_{max} el conjunto difuso cuyo contador acumuló el máximo valor en la agregación.
- $I_{\alpha_{max}}$ valor del contador para α_{max} , que se toma como el nuevo valor de pertenencia.
- $lado_{\alpha_{max}}$ el nuevo valor de lado de la función, el cual se toma como apuntando hacia el conjunto vecino del conjunto ganador que haya obtenido la puntuación más alta en la agregación.

El método de predicción de series de tiempo RDST ofrece una manera directa de evaluar la calidad de una predicción: puesto que mientras más semejantes sean los registros utilizados para la predicción a los datos de entrada, mayor será la certeza de que la predicción es correcta. Entonces, se puede derivar una medida de confianza en la predicción hecha por el agente como:

$$\tilde{C} = \frac{1}{k} \sum_{j=1}^k \mathcal{S}(C(t_j), C(t)) \quad (5.5)$$

Este será el valor de confianza que el predictor enviará al agente árbitro local.

5.2. Redes neuronales *feedforward*

El segundo método de razonamiento usado en este trabajo, como herramienta de predicción del agente, es la red neuronal *feedforward* entrenada mediante el algoritmo *backpropagation* estándar presentado en [Rumelhart et al., 1986] (RNA).

Una red neuronal *feedforward* es un sistema integrado por un conjunto de dispositivos de proceso simples llamados neuronas, que se ordenan en grupos llamados capas. Hay una capa de entrada, que obtiene los datos de entrada, una capa de salida que arroja la salida de la red y un conjunto de capas intermedias llamadas capas ocultas. Las neuronas en cada capa, desde la capa de entrada a la capa inmediata anterior a la de salida, se interconectan solamente a las neuronas de la capa siguiente. La salida de cada neurona es utilizada como entrada a las neuronas con las que se conecta. La salida i -ésima de la neurona en cualquier capa está dada por

$$x_i^{(l)} = g(a_i^{(l)}) \quad (5.6)$$

donde, $a_i^{(l)}$ es denominado como el nivel de activación de la neurona. Este valor describe el efecto de los datos de entrada en la actividad de la neurona. En este trabajo, la activación de la neurona i -th en la capa l se define como

$$a_i^{(l)} = \left(\sum_{j=1}^{n^{(l-1)}} w_{ij}^{(l)} x_j^{(l-1)} \right) - \Theta_i^{(l)} \quad (5.7)$$

donde $w_{ij}^{(l)}$ representa el peso de interconexión entre la j -ésima neurona en la capa $l-1$ y la i -ésima neurona en la capa l , y se llama peso sináptico entre esas neuronas, $n^{(l-1)}$ es el número de neuronas en la capa $l-1$, $x_j^{(l-1)}$ es la salida de la neurona j en la capa $l-1$ y $\Theta_i^{(l)}$ es el valor de *bias* o valor umbral de la neurona i en la capa l . Por otra parte, la función de salida $g(\cdot)$ usada en este trabajo está dada por

$$g(a) = \frac{1}{1 + e^{-Ga}} \quad (5.8)$$

donde G es la ganancia de la red.

El entrenamiento de la red neuronal consiste en ajustar los pesos sinápticos entre las neuronas, de tal manera que las salidas de las neuronas en la capa de salida correspondan a una cierta salida prevista, asociada a los datos de entrada. El error al asociar los datos de entrada al valor de salida se calcula individualmente para cada neurona en la capa de salida y se propaga hacia la primera capa oculta.

El peso sináptico $w_{ij}^{(l)}$ que conecta la j -ésima neurona en la capa $l-1$ a la i -ésima neurona en la capa l se actualiza de forma iterativa, mediante la ecuación

$$w_{ij}^{(l)}(k+1) = w_{ij}^{(l)}(k) + \Delta w_{ij}^{(l)}(k) \quad (5.9)$$

donde

$$\Delta w_{ij}^{(l)}(k) = \lambda \delta_i^{(l)}(k) x_j^{(l-1)} + \alpha \Delta w_{ij}^{(l)}(k-1) \quad (5.10)$$

$\lambda \in (0, 1)$ es la razón de aprendizaje y $\alpha \in (0, 1)$ es el momento. $\delta_i^{(l)}$ es la señal de error de la neurona i en la capa l y se define como

$$\delta_i^{(l)} = \begin{cases} g'(a_i^{(l)}) (y_i - x_i^{(l)}) & \text{Para neuronas en la capa de salida} \\ g'(a_i^{(l)}) \sum_{m=1}^{n^{(l+1)}} \delta_m^{(l+1)} w_{mi}^{(l+1)} & \text{Para neuronas en las capas anteriores a la de salida} \end{cases} \quad (5.11)$$

Para una función sigmoidea de salida

$$g'(a) = Gg(a) (1 - g(a)) \quad (5.12)$$

Las redes neuronales carecen de un mecanismo para la evaluación del error que cometan al tratar de identificar un patrón. En este trabajo, se emplea una medida de calidad de la red neuronal entrenada, derivada de la señal de error y definida como:

$$\tilde{C} = 1 - \frac{1}{G \cdot h \cdot n_O} \sum_{j=1}^h \sum_{i=1}^{n_O} \delta_{ij}^{(O)} \quad (5.13)$$

donde $\delta_{ij}^{(O)}$ es la señal de error de la i -ésima neurona en la capa de salida, al predecir el j -ésimo valor objetivo de un conjunto de datos de prueba de longitud h . El valor de calidad \tilde{C} se utiliza como un valor fijo de confianza en la red neuronal al predecir valores objetivo desconocidos.

Finalmente, la salida de la red neuronal es codificada como una tercia difusa antes de enviarla al agente árbitro local, con el fin de que éste pueda integrarla con las otras predicciones.

5.3. Redes neuronales *feedforward* difusas

El tercer mecanismo de razonamiento empleado en este trabajo es una red neuronal entrenada mediante una adaptación difusa del algoritmo *backpropagation* (RNAD). Esta

variante busca superar dos restricciones del algoritmo de entrenamiento estándar: El primero de ellos, el menos importante, es la naturaleza cuantitativa de este método de entrenamiento, que contrasta con la aproximación cualitativa usado en el resto del sistema.

La segunda limitación que se intenta superar con el enfoque difuso es el comportamiento de la función de salida sigmoidea para rangos grandes de valores de los datos. De la ecuación 5.8 puede observarse que $g(a)$ es muy cercano a la unidad para valores grandes de la activación a , mientras que los valores pequeños de activación conducen a $g(a) \approx 0.5$ (véase la figura 5.1). De esta manera, los valores grandes en los datos de entrada requieren valores pequeños de pesos sinápticos en la capa de entrada, mientras que valores pequeños de entrada necesitan pesos grandes, haciendo difícil la tarea de ajuste de los pesos sinápticos. Por otra parte, si la salida de la red neuronal es un valor en el mismo rango de datos de entrada, la función de salida se debe modificar para dar valores fuera del rango $[0, 1]$. Una manera común de solucionar estos problemas consiste en normalizar los datos, lo cual requiere conocer su rango completo de valores. El método de entrenamiento *backpropagation* difuso propuesto en esta sección, opera sobre los valores de pertenencia, que ya están en el rango $[0.5, 1]$ (debido a la estrategia de codificación difusa usada en este trabajo).

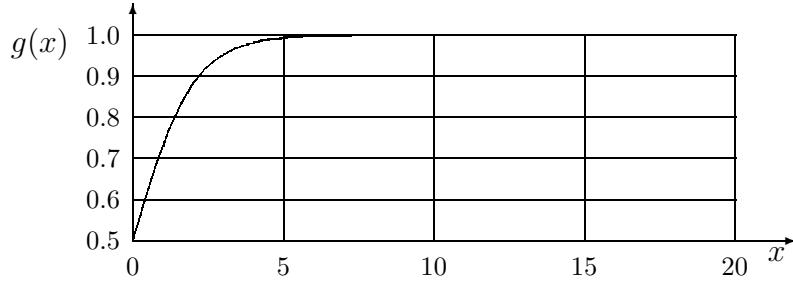


Figura 5.1: Comportamiento de la función de salida de una neurona, para valores de activación grandes.

Los datos de entrada a una red neuronal difusa son tercias cualitativas $r = (Class, \mu, Side)$, de manera que, la red neuronal debe operar sobre los tres elementos de la tercia para mantener la coherencia de los datos. La salida en la i -ésima neurona en la capa l está dada por

$$X_i^{(l)} = (\alpha_i^{(l)}, \mu_i^{(l)}, lado_i^{(l)}) \quad (5.14)$$

- La clase $\alpha_i^{(l)}$ se obtiene agregando las señales provenientes de las neuronas en la capa $l - 1$, de acuerdo a los contadores

$$I_{\alpha_s} = \sum_{j=1}^n w_{i,j}^{(l)} \cdot \chi_j \quad (5.15)$$

donde $\alpha_s \in \{baja, estable, alza\}$, $w_{ij}^{(l)}$ es el peso sináptico de la j -ésima neurona en la capa $l-1$ a la neurona i -ésima neurona en la capa l . χ_j se calcula mediante la relación siguiente:

$$\chi_j = \begin{cases} g(a_j^{(l-1)}) & X_j^{(l-1)} \in \alpha_s \\ 1 - g(a_j^{(l-1)}) & X_j^{(l-1)} \in \beta, \mathcal{N}(\beta, \alpha_s) \text{ y } X_j^{(l-1)} \mapsto \alpha_s \\ 0 & \text{c.o.c.} \end{cases} \quad (5.16)$$

$\alpha_i^{(l)}$ se elige como el conjunto difuso cuyo contador obtiene el valor más alto en la agregación.

- El nuevo valor de pertenencia, $\mu_i^{(l)}(\alpha_i^{(l)})$, es la salida de la función $g(\cdot)$, definida por la ecuación 5.8, esto es

$$\mu_i^{(l)} = \frac{1}{1 + e^{-G a_i^{(l)}}}$$

siendo G la ganancia de la red. La activación $a_i^{(l)}$, en este caso, está dada por

$$a_i^{(l)} = \max_s(I_{\alpha_s}) - \vartheta_i^{(l)} \quad (5.17)$$

$\vartheta_i^{(l)}$ es el valor umbral. Puesto que el valor de pertenencia de la salida en todas las neuronas está en el rango $[0,5, 1,0]$, en este trabajo $\vartheta_i^{(l)}$ se fija a 0.5 para todas las neuronas.

- El nuevo valor de lado de función $lado_i^{(l)}$ se toma como apuntando al conjunto difuso cuyo contador obtuvo el segundo valor más alto en la agregación.

La señal de error de la neurona i en la capa l , $\delta_i^{(l)}$, se define como

$$\delta_i^{(l)} = \begin{cases} g'(a_i^{(l)}) \mathcal{D}(X_i^{(l)}, X_i) & \text{Para neuronas en la} \\ & \text{capa de salida} \\ g'(a_i^{(l)}) \sum_{m=1}^{n^{(l+1)}} \delta_m^{(l+1)} w_{mi}^{(l+1)} & \text{Para neuronas anteriores} \\ & \text{a la capa de salida} \end{cases} \quad (5.18)$$

siendo $\mathcal{D}(\cdot)$ la función de distancia suave de Gower definida en la ecuación 3.21, X_i es el valor deseado de salida en el i -ésimo nodo en la capa de salida y $X_i^{(l)}$ el valor real de salida en el mismo nodo.

La medida de calidad se calcula, nuevamente, mediante la ecuación 5.13.

5.4. Comparación entre redes FFNN cuantitativas y cualitativas

El desempeño de la red neuronal entrenada con el algoritmo difuso es comparado con el funcionamiento de la red neuronal entrenada con el algoritmo cuantitativo estándar, mediante los problemas siguientes (definidos en el capítulo 3):

- La componente z del atractor de Lorenz.
- La componente z del atractor de Lorenz, con tendencia de crecimiento lineal y dispersión con crecimiento lineal.
- La componente z del atractor de Lorenz, con tendencia de crecimiento cuadrática y dispersión con tendencia de crecimiento cuadrática.

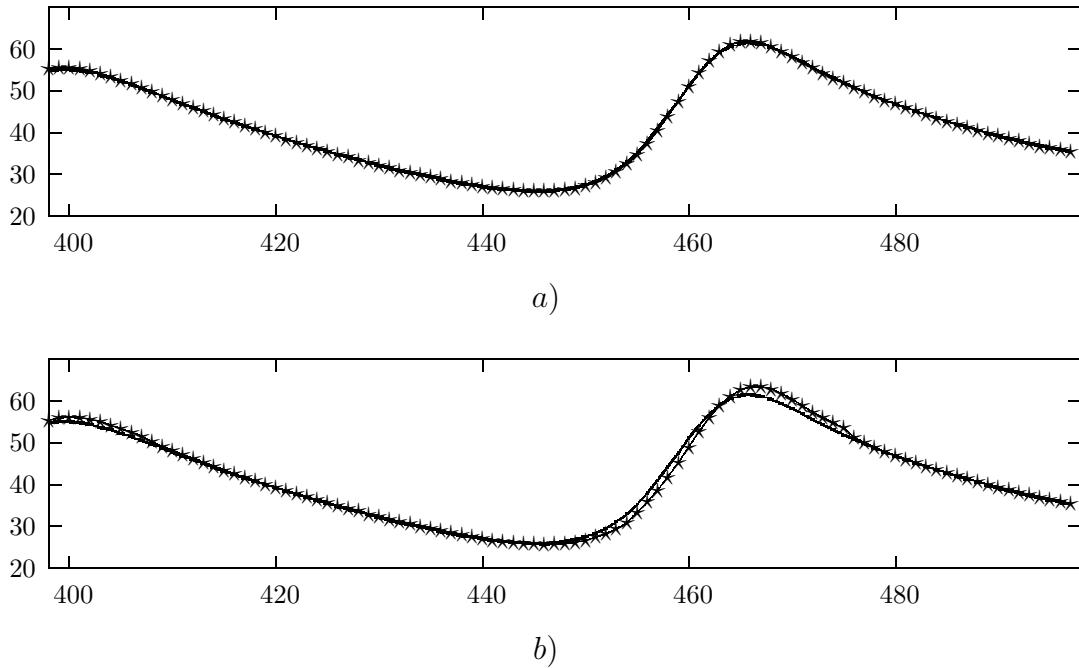


Figura 5.2: Prediccion neuronal de la serie Lorenz.

a) Predicción mediante una red neuronal entrenada con el algoritmo *backpropagation* estándar. Predicción mediante una red neuronal entrenada con el algoritmo *backpropagation* difuso. En ambos casos, la línea continua representa los puntos objetivo, mientras que la línea señalada con estrellas (*) es la predicción correspondiente.

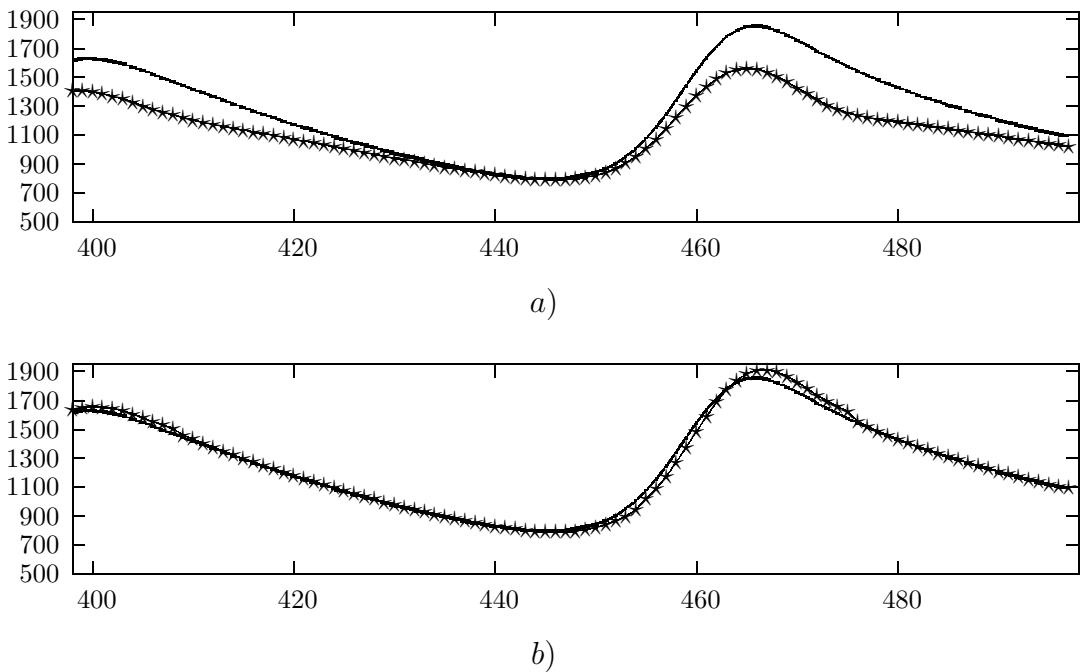


Figura 5.3: Predicción neuronal de la serie Lorenz 2: lineal.

a) Predicción mediante una red neuronal entrenada con el algoritmo *backpropagation* estándar. Predicción mediante una red neuronal entrenada con el algoritmo *backpropagation* difuso. En ambos casos, la línea continua representa los puntos objetivo, mientras que la línea señalada con estrellas (*) es la predicción correspondiente.

Como puede apreciarse en la figura 5.2, la predicción de las series de tiempo estacionarias, hecha por la red neuronal entrenada con el método backpropagation estándar es ligeramente mejor que la predicción obtenida con la red neuronal entrenada con el algoritmo difuso. Por otra parte, cuando desaparecen las condiciones de comportamiento estacionaria, el desempeño de la red neuronal entrenada con el algoritmo difuso es claramente superior (figuras 5.3 y 5.4).

En ambos casos se empleó una red neuronal con 15 nodos de entrada, 20 nodos en la única capa oculta, y un nodo de salida. Los parámetros de entrenamiento fueron: ganancia 1, razón de aprendizaje 0.05, momento a 0.5 y *bias* igual a 1 para los modelos RNA y 0.5 para los modelos RNAD.

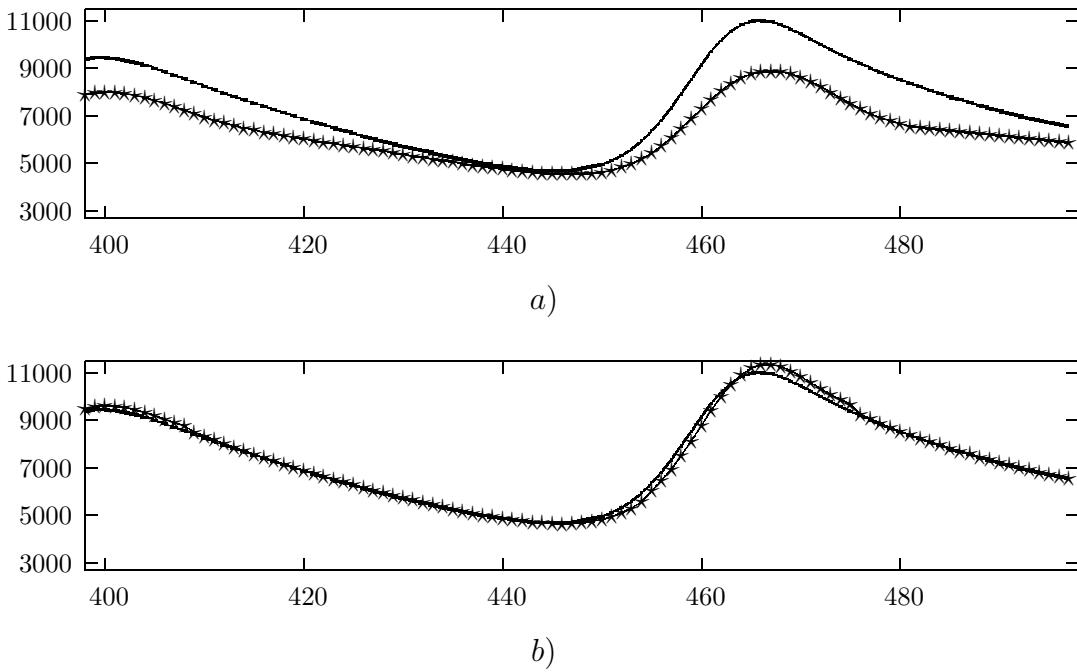


Figura 5.4: Predicción neuronal de la serie Lorenz 3: cuadrática.

a) Predicción mediante una red neuronal entrenada con el algoritmo *backpropagation* estándar. Predicción mediante una red neuronal entrenada con el algoritmo *backpropagation* difuso. En ambos casos, la línea continua representa los puntos objetivo, mientras que la línea señalada con estrellas (*) es la predicción correspondiente.

Conjunto de datos	Método de entrenamiento	
	backpropagation estándar	backpropagation difuso
Lorenz 1	0.1202752	0.0960687
Lorenz 2: lineal	0.3987142	0.0933912
Lorenz 3: cuadrático	0.5365449	0.0869285
Número de manchas solares	0.2661172	0.2782048
Libra esterlina	0.3169981	0.0972515
Dólar canadiense	0.4199859	0.1927423
Marco alemán	0.2506006	0.0495888
Yen japonés	0.3608898	0.0386783
Franco suizo	0.1830959	0.0525080
NMSE promedio	0.3170246	0.1094846

Cuadro 5.1: Error al predecir diferentes series de tiempo usando el algoritmo *backpropagation* estándar y el algoritmo *backpropagation* difuso.

Capítulo 6

Programación genética de sistemas multiagente híbridos

Los datos de observación disponibles acerca de un sistema dinámico pueden dar lugar a un conjunto muy grande de modelos, dependiendo de las variables de entrada y de salida consideradas. Encontrar los mejores modelos para un sistema cuya dinámica se conoce de manera incompleta es una tarea difícil, debido al problema de explosión combinatoria que se presenta cuando crece el número de variables que serán consideradas.

Un concepto clave en la metodología de simulación desarrollada en el presente trabajo es la programación genética (PG) del sistema multiagente. El uso de esta aproximación para generar la población de un SMA resuelve el problema de la explosión combinatoria que se presenta al elegir las variables para modelar el sistema físico. Así mismo, la aproximación de modelado genético busca superar otra limitación de la mayoría de las técnicas de modelado y simulación: la tendencia impuesta por los datos de entrenamiento. Los mecanismos evolutivos empleados para desarrollar el sistema multiagente, también permiten la actualización automática de los modelos de razonamiento utilizados, cuando la dinámica de los datos cambia drásticamente.

En este capítulo, se presenta un algoritmo no estándar de programación genética utilizado para desarrollar los sistemas multiagente para el modelado híbrido inteligente de sistemas dinámicos.

6.1. Conceptos básicos

El algoritmo de programación genética empleada para configurar un SMA para modelado híbrido inteligente se basa en los conceptos de diversidad coevolutiva, aptitud colectiva,

comportamiento conveniente, evolución filogenética y evolución ontogenética.

6.1.1. Diversidad coevolutiva

La mayoría de las aplicaciones de computación evolutiva suponen una cierta homogeneidad en la población de individuos que participen en el proceso evolutivo. De tal manera que todos los individuos son seleccionados para evolución en el mismo tiempo, o en términos de un sorteo o de un certamen. Los SMA empleados como base del modelado híbrido inteligente, por otro lado, incluyen una variedad de predictores, mostrando una naturaleza heterogénea. Dicha variedad de modelos mejora la posibilidad de obtener una buena solución [Gorodetski et al., 2002, Ortega et al., 2001], sin embargo, también dificulta la obtención del modelo colectivo [Duin, 2002]. En el caso del modelado híbrido inteligente, cohabitan agentes provistos de modelos de configuración simple (como los predictores RDST) con agentes cuyos modelos sólo quedan sintonizados después de un largo proceso (por ejemplo, los predictores RNA y RNAD). Al tratar estos agentes como los individuos de un algoritmo genético, las diferencias en la complejidad de la sintonización representan una forma de diversidad coevolutiva [Ridley, 2003].

Para abordar el problema de selección de individuos, dando el tiempo suficiente para la sintonización de los modelos complejos, pero sin mantener ociosos a los modelos simples, se emplean tres períodos de evolución, que se definen como sigue:

Definición 6.1. Equipo. Éste es el nivel de población básico en el que debe sobrevivir el agente: Los agentes predictores deben mostrar un buen desempeño para solucionar un problema dado mientras que colaboran en diversos equipos a lo largo de su ciclo de vida. Durante el proceso de entrenamiento, se van seleccionando grupos diferentes de agentes de la población completa, para formar equipos de trabajo. El desempeño de los agentes es registrado mientras participan en cada uno de los equipos de trabajo en los que es asignado.

Definición 6.2. Generación. Periodo de tiempo a lo largo del cual un conjunto fijo de individuos de una especie dada vive, cohabita y compite por los recursos disponibles. Este concepto describe el cambio básico en la composición de la población de un SMA. Al final de cada generación se genera una nueva población, basándose en el desempeño mostrado por los agentes en la población actual a lo largo de los diversos equipos en los cuales han participado. Las generaciones de diversas especies pueden tener diferentes duraciones.

Definición 6.3. Era. Este concepto describe un cambio global en la población de un SMA. Tales acontecimientos involucran cambios generacionales de todas las especies en el SMA.

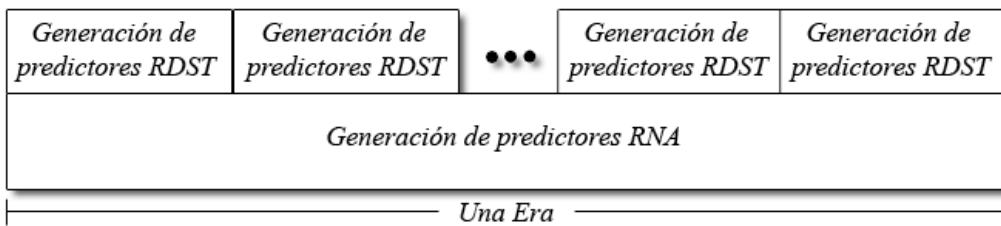


Figura 6.1: Relación entre la duración de una generación y una era.

Este conjunto de conceptos evolutivos describe y emula la diversidad evolutiva mostrada por la naturaleza. Las diversas especies en un ecosistema natural muestran diversa tasas de evolución, dependiendo de su complejidad orgánica y de los recursos naturales disponibles.

En este trabajo se utilizan dos especies de predictores: La primera especie utiliza modelos RDST y la segunda modelos RNA (y RNAD). A lo largo de una era transcurren varias generaciones de predictores RDST y una sola generación de predictores RNA, como se muestran en figura 6.1.

6.1.2. Comportamiento conveniente y evaluación colectiva de aptitud

Dos objetivos de la metodología de diseño desarrollada en este trabajo son: 1) realizar la simulación de un sistema dinámico complejo mediante una combinación de modelos simples, y 2) tener la capacidad de reemplazar en tiempo de ejecución aquellos modelos simples que han dejado de ser de utilidad. Para lograr el primer objetivo, se emplea un proceso de programación genética, basado en el concepto de evaluación colectiva de aptitud, para desarrollar el SMA. La opinión de cada agente se evalúa parcialmente en términos de su impacto en la predicción global, de manera que modelos individuales muy simples e incluso erróneos pueden ser considerados útiles, si mejoran el desempeño del SMA. El segundo objetivo se logra conservando a aquellos agentes que muestran el mejor desempeño promedio después de colaborar en diversos equipos.

Debido a la estrategia de evaluación colectiva, es de esperar que solamente el modelo colectivo de cada grupo de modelado sea apropiado para la descripción del sistema físico. Así pues, la solución óptima que se busca mediante la programación genética del SMA, consiste en una colección de individuos con la capacidad para colaborar en la solución de una clase dada de problemas, más que un individuo capaz de resolver el problema completo por sí mismo. La búsqueda del modelo global óptimo corresponde en seleccionar esos agentes que, al trabajar en diferentes equipos, conduzcan a un buen desempeño colectivo.

Definición 6.4. El comportamiento de un agente es conveniente si y solamente si tal comportamiento conduce a mejorar el desempeño de la especie.

Definición 6.5. Una evaluación colectiva de aptitud es un proceso de evaluación de aptitud que favorece a los agentes con un comportamiento conveniente.

La aptitud del predictor A_j se evalúa a partir de dos medidas de error: el error directo (e_j), que describe la calidad individual del modelo y el error inducido (\tilde{e}_j), que describe su habilidad para colaborar en la predicción colectiva. Estas medidas de error se definen como

$$e_j = \frac{1}{N_T} \sum_{n=0}^{N_T} \min \left(\left| \frac{O_n - \hat{O}_{nj}}{d} \right|, 1 \right) \quad (6.1)$$

y

$$\tilde{e}_j = \frac{1}{N_T} \sum_{n=0}^{N_T} \left[\min \left(\left| \frac{O_n - \hat{O}_n}{d} \right|, 1 \right) - \min \left(\left| \frac{O_n - \hat{O}_n^{(-j)}}{d} \right|, 1 \right) \right] \quad (6.2)$$

donde N_T es el número de tareas de predicción en las cuales el agente ha participado; O_n es el valor objetivo en la n^{th} tarea de predicción; \hat{O}_{nj} la predicción correspondiente hecha por el agente A_j ; \hat{O}_n es la predicción colectiva y $\hat{O}_n^{(-j)}$ es la predicción colectiva que se obtendría al eliminar la opinión del agente A_j . d se define como:

$$d = \begin{cases} O_n & \text{Si } O_n \neq 0 \\ 1 & \text{c.o.c} \end{cases} \quad (6.3)$$

El error global (E_j) del agente se calcula como:

$$E_j = \begin{cases} e_j \cdot \tilde{e}_j & \text{Si } \tilde{e}_j > 0, e_j \neq 0 \\ \tilde{e}_j/e_j & \text{Si } \tilde{e}_j < 0, e_j \neq 0 \\ \min(\tilde{e}_j, e_j) & \text{c.o.c} \end{cases} \quad (6.4)$$

Los mejores individuos en cada generación son aquellos que acumulan un menor error global.

Estos conceptos subrayan la importancia del contexto para evaluar la aptitud de un individuo. Debido a esta aproximación a la evaluación de aptitud, la organización de la población global cambia también a lo largo del proceso genético.

6.1.3. Evolución filogenética y evolución ontogenética

La naturaleza emplea dos mecanismos básicos de adaptación que difieren en tiempo y espacio, pero que son complementarios para mejorar las expectativas de éxito de los individuos [Floreano and Mondada, 1996, Nolfi and Parisi, 1997, Nolfi and Floreano, 1999]. El

primero de tales mecanismos es la evolución a nivel filogenético, mediante la cual se modifica gradualmente la estructura de los individuos de una especie [Floreano and Mondada, 1996, Nolfi and Floreano, 1999], a lo largo de muchas generaciones, siendo la especie la beneficiaria de esta adaptación. Este concepto es la base de los algoritmos genéticos estándar: partiendo de una población dada, el AG va produciendo nuevos individuos que se espera sean cada vez más aptos para interactuar con su medio que lo que fueron sus predecesores.

El segundo mecanismo de adaptación es el aprendizaje, que se desarrolla a un nivel ontogenético y que consiste en la evolución que cada individuo experimenta a lo largo de su vida [Piaget, 1983, Harter, 2001, Bongard and Pfeifer, 2003, Floreano and Mondada, 1996, Nolfi and Floreano, 1999]. En este caso, es el individuo el principal beneficiario del proceso de adaptación y sólo una parte muy pequeña de estos recursos aprendidos se heredan a las siguientes generaciones. Este concepto ha sido ampliamente utilizado en aprendizaje automático [Harter, 2001], especialmente en el ámbito de las redes neuronales. Dada la importancia de este concepto, Bongard y Pfeifer introducen el concepto de ontología artificial [Bongard and Pfeifer, 2003].

Sin embargo, y a pesar de que diversos autores han destacado la importancia y complementariedad de ambos mecanismos de adaptación, ha habido pocos experimentos para combinar, explícitamente, ambos conceptos evolutivos [Floreano and Mondada, 1996].

En el presente trabajo se emplean ambos conceptos de manera selectiva, dependiendo de la especie de predictores: Los modelos RDST carecen de un mecanismo de entrenamiento, por lo cual solo emplea el mecanismo de adaptación filogenético proporcionado por el algoritmo genético. Los predictores RNA y RNAD, por otro lado, cuentan con un mecanismo de entrenamiento eficiente, por lo cual, la adaptación de tales agentes utiliza, principalmente el proceso de evolución a nivel ontogenético que proporciona el algoritmo de entrenamiento.

6.2. Modelado evolutivo

La programación genética de un SMA, en el presente trabajo, consiste en elegir los modelos parciales que se utilizarán como mecanismo de razonamiento para todos los predictores en cada grupo de modelado, que describan colectivamente el sistema dinámico de acuerdo a un conjunto de series de tiempo conocidas.

El proceso del PG se realiza de acuerdo a las etapas siguientes:

1. Inicialización del proceso de PG.

2. Entrenamiento.
3. Reentrenamiento en línea.

6.2.1. Inicialización del proceso de PG

La inicialización del proceso PG consiste en la generación de la población inicial de predictores en cada grupo de modelado en el SMA. Los agentes que desempeñan papeles del planificador, interfaz de usuario, *wrapper* y árbitro, no cambian a lo largo del proceso genético.

Los primeros parámetros que se definen son el número y tamaño de los grupos de modelado. Estos parámetros son determinados por la complejidad y la segmentación hechas del problema. En este trabajo, los problemas abordados consisten en la predicción de bases de datos de series de tiempo. La tarea asignada a cada grupo de modelado consiste en predecir una de tales series de tiempo. Así pues, se crea un grupo de modelado para cada serie de tiempo.

La elección del tamaño de la población para un dominio particular es uno de los aspectos críticos en el desarrollo de un algoritmo genético [Harik et al., 1999, Mandischer, 1993]. Se suele asumir, como regla intuitiva, que lo deseable es buscar una solución de compromiso que permita tener el máximo número de individuos posibles en cada población (a fin de tener una alta diversidad genética), a la vez que se mantiene un desempeño aceptable (en tiempo y en calidad) [Harik et al., 1999, Cantú-Paz and Goldberg, 2003]. Sin embargo, existen pocos estudios que ofrezcan reglas formales para la elección de este parámetro [Harik et al., 1999, Mandischer, 1993]. Harik et. al., por ejemplo, presentan un método probabilístico para elegir el tamaño de la población en el algoritmo genético simple [Harik et al., 1999]. En el mismo estudio se identifican dos factores dependientes del tamaño de la población que influyen en la calidad del algoritmo genético: 1) el suministro inicial de individuos y 2) la calidad de estas soluciones iniciales. De esta manera, mientras mejores sean las soluciones individuales, menor es el tamaño de población necesario para garantizar la convergencia del algoritmo.

En el presente trabajo se explota la alta calidad que se puede obtener de modelos con un buen mecanismo de adaptación ontogenética, como las redes neuronales, a fin de reducir el tamaño de las poblaciones. De esta manera se reduce el problema generado por las restricciones del equipo físico usado. El tamaño de cada grupo de modelado se elige igual para todos los grupos de modelado.

Una vez definidos estos parámetros, se generan los predictores como sigue.

Modelos RDST

Los diferentes modelos RDST usados para la predicción de una variable de salida dada se diferencian por las variables de entrada consideradas. En el trabajo actual, las variables de entrada a los modelos parciales se toman de las series de tiempo.

El primer paso en la generación de los modelos RDST consiste en definir los valores para el ancho máximo de un modelo W_{max} (el número máximo de series de tiempo consideradas en el modelo) y la profundidad máxima D_{max} (la medición más temprana que se considerará). A continuación, se generan los modelos RDST, de la manera siguiente:

1. Para cada modelo RDST, se escogen aleatoriamente los valores de ancho $W \in [1, W_{max}]$ y de profundidad $D \in [1, D_{max}]$. Si $W = 1$, el modelo consiste en un segmento de la serie de tiempo de la variable de salida. Si $D = 1$, la relación no describe causalidad temporal entre las variables de entrada y la variable de salida. Si $W = D = 1$, se eligen nuevos valores para W y D . Esto genera una matriz que contendrá una estructura general del modelo. Así, por ejemplo, la matriz

$$\begin{array}{c} t \setminus x \\ \hline v_i & u_j & u_k \\ t - 2\delta t & \dots & \dots & \dots \\ t - \delta t & \dots & \dots & \dots \\ t & \dots & \dots & \dots \end{array} \quad (6.5)$$

permite generar modelos para la variable v_i en el tiempo actual t , en los que pueden aparecer cualquiera de las variables de entrada $u_j(t), u_k(t), v_i(t - \delta t), u_j(t - \delta t), u_k(t - \delta t), v_i(t - 2\delta t), u_j(t - 2\delta t)$ y $u_k(t - 2\delta t)$. Aquí, v_i es la variable para la cual se está generando el modelo, u_j y u_k son otras dos variables cualquiera en las series de tiempo.

2. Cada serie de tiempo es etiquetada con un índice. Dado el número total de series de tiempo, $NumVars$, de donde se seleccionarán las variables de entrada al modelo, se genera un conjunto de $W - 1$ números aleatorios en el rango $[1, NumVars]$, diferentes al índice de la variable de salida. Estos números corresponden a las variables de entrada al modelo. De esta manera, si $i = 1, j = 2$ y $k = 4$, entonces el modelo en 6.5 tendría la estructura general

$$\begin{array}{c} t \setminus x \\ \hline v_1 & v_2 & v_4 \\ t - 2\delta t & \dots & \dots & \dots \\ t - \delta t & \dots & \dots & \dots \\ t & \dots & \dots & \dots \end{array} \quad (6.6)$$

3. Para cada pareja ordenada (v_j, t_k) , que define la medición de la variable v_j en el tiempo t_k , se define aleatoriamente si ésta debe aparecer en el modelo para $v_i(t)$. Así, si se señala con \times las parejas que deben aparecer en el modelo, entonces la matriz

$$\begin{array}{c|ccc} t \setminus x & v_1 & v_2 & v_4 \\ \hline t - 2\delta t & \times & \times & \dots \\ t - \delta t & \dots & \times & \times \\ t & - & \dots & \times \end{array} \quad (6.7)$$

corresponde al modelo

$$v_1(t) = \tilde{F}(v_4(t), v_2(t - \delta t), v_4(t - \delta t), v_1(t - 2\delta t), v_2(t - 2\delta t)) \quad (6.8)$$

La pareja (v_1, t) está señalada con $-$, ya que ésta es la variable de salida.

4. El modelo es registrado como un vector cuyos elementos son las parejas seleccionadas.
5. Para cada modelo se crea un agente que utilice el modelo para simular el sistema dinámico.

Modelos RNA

Los algoritmos genéticos y las redes neuronales se han combinado de dos maneras principales [Cantú-Paz and Kamath, 2003]: 1) para elegir los pesos de interconexión, y 2) para elegir la estructura de la red. Uno de los problemas que surgen cuando se combinan redes neuronales es el gran tiempo de cómputo requerido [Cantú-Paz and Kamath, 2003]. Este problema es particularmente notable en el presente trabajo, dado que la solución que se busca no es una red individual, sino un conjunto de redes neuronales capaces de colaborar con un grupo más amplio y diverso de predictores. Este problema se reduce imponiendo restricciones a la estructura de la red y confiando en la diversidad de modelos.

El primer paso para generar un modelo RNA o RNAD consiste en definir el número de capas y número de nodos en una capa dada, y los valores para los parámetros de ganancia, razón de aprendizaje, momento y *bias*. Sin embargo, no existen reglas formales para la elección de ninguno de estos valores, situación que ya se ha señalado como un tema importante a investigar (ver, por ejemplo [Yao and Tan, 2000, De Falco et al., 1998, Kolarik and Rudorfer, 1994]). De esta manera, las restricciones a la estructura de las redes y los valores de parámetros se establecen en base a los valores reportados usualmente en la literatura.

A continuación, los modelos RNA se generan como sigue:

1. Se elige aleatoriamente el número de capas ocultas (HL). Puesto que en la literatura la mayoría de las redes neuronales suelen reportarse con 1 a 3 capas ocultas, este valor se restringe a $HL \in \{1, 2, 3\}$.
2. Se elige aleatoriamente el número de nodos (n^l) en cada capa (l), excepto para la capa de salida que en este trabajo siempre tiene una sola neurona. Nuevamente, siguiendo los trabajos de predicción de series de tiempo mediante redes neuronales, el número de nodos en cada capa se fija $n^l \in [10, 30]$.
3. Los valores de los parámetros se fijan igual para todos las redes neuronales, de acuerdo a valores sugeridos en la literatura (ver por ejemplo [Velazquez, 2002]). Así, la ganancia se fija a 1, la razón de aprendizaje a 0.05, el momento a 0.5 y el *bias* a 1 para los modelos RNA y 0.5 para los modelos RNAD.
4. Se escogen al azar los pesos de interconexión.
5. finalmente, para cada modelo se crea un agente que utilice el modelo para simular el sistema dinámico.

6.2.2. Entrenamiento genético

La población inicial en cada grupo de modelado está compuesta por agentes dotados con modelos generados al azar. Tales modelos no reflejan el comportamiento del sistema dinámico bajo estudio, y por lo tanto, los agentes muestran una aptitud muy baja para predecir el comportamiento del sistema. Para generar agentes con modelos que se ajusten a los datos de observación, se aplica un proceso evolutivo a partir de esta población inicial. Pero, como sucede con especies diferentes en la naturaleza, los predictores RNA y los predictores RDST evolucionan de diferente manera: Los modelos RDST carecen de mecanismos de autoajuste, confiando así en la evolución filogenética para mejorar el desempeño de la especie. Las redes neuronales, por otra parte, pueden alcanzar un buen desempeño cuando son entrenadas con el método backpropagation, de manera que los predictores neuronales pueden evolucionar ontogenéticamente.

El proceso de entrenamiento genético (figura 6.2) se basa en la estructura *equipo-generación-era* presentada en la sección 6.1.1. De acuerdo a tal estructura, la evolución de los predictores se realiza en dos niveles. En el nivel generacional, cada especie se desarrolla de manera aislada.

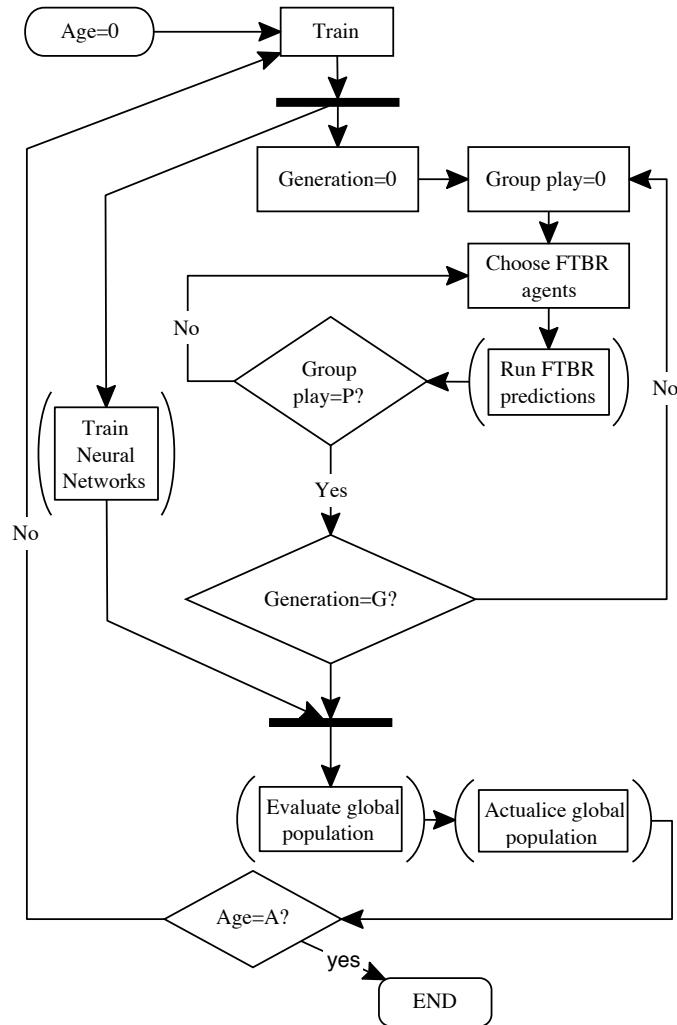


Figura 6.2: Diagrama de flujo del entrenamiento genético.

A nivel de era todas las especies de predictores trabajan juntas y son evaluadas colectivamente. La longitud de una generación de predictores neuronales coincide con la longitud de una era, así que las redes neuronales ejecutan apenas una vez el algoritmo *backpropagation* a lo largo de una era, mientras que en el mismo período transcurren varias generaciones de predictores RDST.

El desempeño de los predictores RDST a través de una generación se mide en función de sus resultados conforme participan en diferentes equipos. Un equipo se conforma mediante un subconjunto de predictores tomados al azar de la población entera del grupo de modelado. A cada equipo se le asigna la predicción de un subconjunto valores objetivos. Todas las

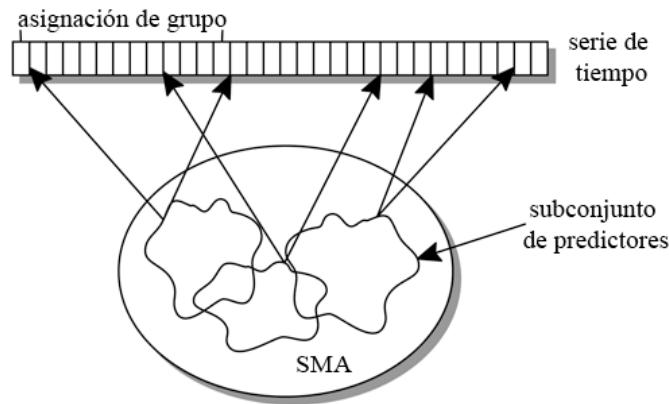


Figura 6.3: Un equipo es un subconjunto de predictores a los que se ha asignado una tareas específica.

opiniones de los agentes son tomadas en consideración para calcular un valor de predicción colectiva. El error de cada agente (ecuación 6.4) se acumula a lo largo de los diversos equipos en los cuales participa. Al final de cada generación, los agentes predictores con mayor error acumulado son identificados y substituidos por agentes nuevos. Los agentes nuevos se generan por medio de operaciones genéticas de cruce y la mutación.

La operación de cruce (para los agentes RDST) se realiza según el procedimiento siguiente (véase la figura 6.4):

1. Se seleccionan dos padres al azar, de la población global en el grupo de modelado donde pertenecen los agentes que serán substituidos.
2. En cada modelo se elige, aleatoriamente, un punto de cruce y se intercambian los segmentos correspondientes. Si existen medidas repetidas para la misma variable en alguno de los nuevos modelos, éstas son eliminadas. Estos nuevos modelos son utilizados para generar agentes nuevos.
3. Si hay por lo menos dos agentes que serán eliminados, éstos son substituidos por los nuevos agentes. Si solamente hay un agente que será substituido, su substituto se elige aleatoriamente de entre los nuevos agentes.
4. El proceso se repite hasta que se han substituido todos los predictores más débiles en la generación actual

Una fracción predefinida de todos los agentes RDST nuevos experimenta un proceso de mutación, la cual consiste en realizar cambios al azar en la estructura de su modelo.

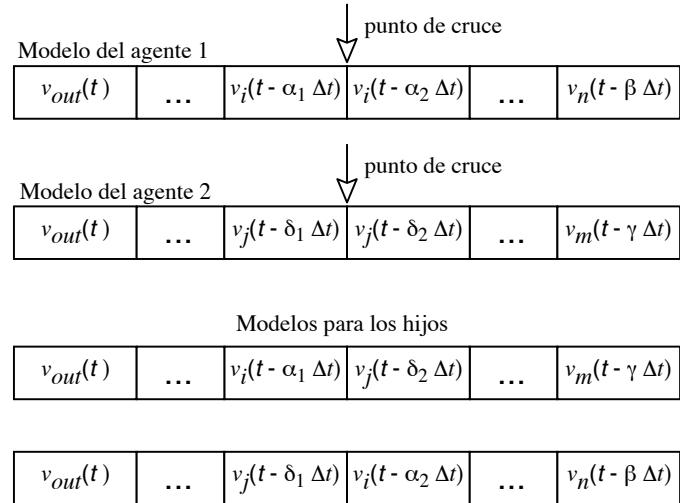


Figura 6.4: Operación de cruce para predictores RDST.

Al final de cada era, la población completa del grupo de modelado, incluyendo todas las especies de predictores, es evaluada en una tarea colectiva de predicción. Los agentes RDST con bajo desempeño se substituyen como en el proceso de evolución generacional antes presentado. A lo largo de una era, todas las redes neuronales (RNA y RNAD) son sometidos a un proceso de entrenamiento, al final del cual, y dadas las restricciones establecidas se espera que muestren un buen desempeño. Sin embargo, aún falta verificar que sean capaces de colaborar con los demás predictores que, en el caso de los experimentos reportados en este trabajo son predictores RDST. Los agentes RNA y RNAD con peor desempeño son reemplazados por nuevos predictores cuyo modelo es generado como en la población inicial.

Se escogen para reemplazo un número N_r de predictores, definido como

$$N_r = \text{inferior} \left(\frac{N_R}{2} \right) \quad (6.9)$$

siendo $\text{inferior}(x)$ una función que regresa el mayor entero menor que x y N_R el número de redes neuronales (RNA o RNAD) en el grupo de modelado.

De esta manera, los predictores complejos RNA y RNAD reciben la oportunidad de adaptarse ontogenéticamente, pero después se da oportunidad al proceso de adaptación filogenética para buscar mejores individuos.

6.2.3. Reentrenamiento en línea

Debido al comportamiento colectivo de un SMA para modelado híbrido inteligente y a la redundancia de los modelos parciales usados por los agentes, es posible eliminar cierta fracción de predictores sin tener una reducción sensible en el desempeño de SMA. Esto permite mantener en funcionamiento el algoritmo genético en tiempo de ejecución con un costo computacional mínimo: El reentrenamiento en línea de un SMA, consiste en substituir una proporción dada de los agentes tomados de entre aquellos que han acumulado un error mayor que un valor dado de tolerancia, a lo largo de la fase de operación.

6.3. Predictores especializados

La aproximación evolutiva utilizada para configurar el SMA da como resultado un sistema de agentes expertos especializados en la predicción de una serie de tiempo. Los agentes que han sobrevivido el proceso de entrenamiento cuentan con modelos diferentes, complementarios e incluso contradictorios para el mismo segmento de un problema dado. El entrenamiento evolutivo presenta también otra ventaja: se realiza de tal manera que cada agente debe, durante cada uno de los pasos de aprendizaje en los cuales participa, predecir un subconjunto de datos tomados aleatoriamente. De esta manera, al final del entrenamiento, los agentes presentan modelos que representan el mismo fenómeno pero derivados a partir de experiencias diferentes. Mediante este proceso, el algoritmo genético ayuda a mantener una diversidad basada no sólo en las técnicas de modelado específicos, sino también en los datos usados en su entrenamiento.

Bibliografía

[Abbadeni, 1996] Abbadeni, N. (1996). A new similarity matching measure application to texture-based image retrieval. In *ASIS 1996 Annual Conference Proceedings*.

[Banzhaf et al., 1998] Banzhaf, W., Nordin, P., Keller, R., and Francone, F. (1998). *Genetic Programming. An Introduction: on the automatic evolution of computer programs and its applications*. Morgan Kaufmann Publishers, Inc.

[Beer, 1996] Beer, R. (1996). A dynamical systems perspective on agent-environment interaction. In Agre, P. E. and Rosenchein, S. J., editors, *Computational Theories of Interaction and Agency*. The MIT Press.

[Bongard and Pfeifer, 2003] Bongard, J. and Pfeifer, R. (2003). Evolving complete agents using artificial ontogeny. In *Morpho-functional Machines: The New Species (Designing Embodied Intelligence)*, pages 237–258. Springer-Verlag, Berlin.

[Box et al., 1994] Box, G., Jenkins, G., and Reinsel, G. (1994). *Time Series Analysis: Forecasting and Control, 3rd ed.* Prentice Hall.

[Bradshaw, 1997] Bradshaw, J. (1997). Introduction to software agents. In Bradshaw, J., editor, *Software Agents*. The AAAI Press/The MIT Press.

[Cantú-Paz and Goldberg, 2003] Cantú-Paz, E. and Goldberg, D. E. (2003). Are multiple runs of genetic algorithms better than one? In Cantú-Paz, E. e. a., editor, *Genetic and Evolutionary Computation Conference – GECCO-2003*, pages 801–812, Berlin. Springer Verlag.

[Cantú-Paz and Kamath, 2003] Cantú-Paz, E. and Kamath, C. (2003). Evolving neural networks to identify bent-double galaxies in the first survey. *Neural Networks*, 16(3-4):507–517.

- [Cellier, 1991] Cellier, F. (1991). General system problem solving paradigm for qualitative modeling. In Fishwick, P. A. and Luker, P. A., editors, *Qualitative Simulation Modeling and Analysis*, pages 51–71. Springer-Verlag, New York.
- [Chappelier and Grumbach, 1994] Chappelier, J. and Grumbach, A. (1994). Time in neural networks. *SIGART Bulletin*, 5(3).
- [Şirin and Güvenir, 1993] Şirin, I. and Güvenir, H. (1993). Prediction of stock market index changes. *Adaptive Intelligent Systems*, pages 149–159.
- [de Albornoz, 1996] de Albornoz, A. (1996). *Inductive Reasoning and Reconstruction Analysis: Two Complementary Tools for Qualitative Fault Monitoring of Large-Scale Systems*. PhD thesis, Universitat Politècnica de Catalunya.
- [De Falco et al., 1998] De Falco, I., Cioppa, A., Iazzetta, A., Natale, P., and Tarantino, E. (1998). Optimizing neural networks for time series prediction. In *Third World Conference on Soft Computing (WSC3)*.
- [Demezeau and Müller, 1990] Demezeau, Y. and Müller, J. (1990). Descentralized artificial intelligence. In Demezeau, Y. and Müller, J.-P., editors, *Proceedings of the First European Workshop on Modelling Autonomous Agents in a Multi-Agent World*. Elsevier Science Publishers.
- [Dorffner, 1996] Dorffner, G. (1996). Neural networks for time series processing. *Neural Network World*, 6(4).
- [Duch and Jankowski, 1994] Duch, W. and Jankowski, N. (1994). Complex systems, information theory and neural networks. Technical report, Nicholas Copernicus University, Poland.
- [Duin, 2002] Duin, R. (2002). The combining classifier: To train or not to train? In R. Kas-turi, D. Laurendeau, C. S., editor, *Proceedings 16th International Conference on Pattern Recognition ICPR16*, volume II, pages 765–771, Los Alamitos. IEEE Computer Society Press.
- [Durfee, 1996] Durfee, E. (1996). Planning in distributed artificial intelligence. In O’Hare, G. and Jennings, N., editors, *Foundations of Distributed Artificial Intelligence*. John Wiley & Sons.

- [Elman, 1990] Elman, J. (1990). Finding structure in time. *Cognitive Science*, 14(2):179–211.
- [Fisher et al., 1988] Fisher, D., Grinstein, G., and Khurana, A. (1988). Theory of random magnets. *Physics Today*.
- [Floreano and Mondada, 1996] Floreano, D. and Mondada, F. (1996). Evolution of plastic neurocontrollers for situated agents. In Maes, P., Mataric, M., Meyer, J., Pollack, J., Roitblat, H., and Wilson, S., editors, *From Animals to Animats IV: Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior*, Cambridge, MA. MIT Press-Bradford Books.
- [Fogel, 1995] Fogel, D. (1995). *Evolutionary Computation*. IEEE Press.
- [Fogel, 1999] Fogel, D. (1999). An overview of evolutionary programming. In Davis, L., K De jong, M. V., and Whitley, L., editors, *Evolutionary Algorithms*. Springer Verlag, New York.
- [Fu, 1994] Fu, L. (1994). *Neural Networks in Computer Intelligence*. McGraw-Hill.
- [Giles et al., 1997] Giles, L., Lawrence, S., and Tsoi, C. (1997). Rule inference for financial prediction using recurrent neural networks. In *Proceedings of IEEE/IAFE Conference on Computational Intelligence for Financial Engineering*.
- [Gorodetski et al., 2002] Gorodetski, V., Karsayev, O., and Samoilov, V. (2002). Multi-agent data fusion systems: Design and implementation issues. In *Proceedings of the 10th International Conference on Telecommunication Systems - Modeling and Analysis*, Monterey, CA, pages 762–774.
- [Gorodetski et al., 2003] Gorodetski, V., Karsayev, O., and Samoilov, V. (2003). Distributed learning of information fusion: A multiagent approach. In *Proceedings of the Sixth International Conference on Information Fusion*, pages 318–325.
- [Gower, 1971] Gower, J. (1971). A general coefficient of similarity and some of its properties. *Biometrics Journal*, (27):857–874.
- [Gutknecht and Ferber, 1997] Gutknecht, O. and Ferber, J. (1997). Madkit: Organizing heterogeneity with groups in a platform for multiple multi-agent systems. Technical Report R.R.LIRMM 97188, Laboratoire D’Informatique, de Robotique et de Microélectronique de Montpellier. CNRS – Université Montpellier II.

BIBLIOGRAFÍA

- [Harik et al., 1999] Harik, G., Cantú-Paz, E., Goldberg, D., and Miller, B. (1999). The gambler's ruin problem, genetic algorithms, and the sizing of populations. *Evolutionary Computation*, 7(3):231–253.
- [Harter, 2001] Harter, D. (2001). Ontogenetic development of skills, strategies and goals for autonomously behaving systems. In *Proceedings of the 5th World Multi-Conference on Systemics, Cybernetics and Informatics (SCI 2001)*, pages 178–181, Orlando, FL.
- [Hellström and Hellström, 1998] Hellström, T. and Hellström, K. (1998). Predicting the stock market. Technical Report IMa-TOM-1997-07, Dept. of Mathematics and Physics, Mälardalen University, Sweden.
- [Herz, 1996] Herz, A. (1996). Global analysis of recurrent neural networks. In Domany, E., van Hemmen, J., and Schulten, K., editors, *Models of Neural Networks III, Springer Series Physics of Neural Networks*. Springer-Verlag.
- [Iba, 1999] Iba, H. (1999). Evolving multiple agentes by genetic programming. In Lee Spector, W. B. L., O'Reilly, U.-M., and Angeline, P. J., editors, *Advances in Genetic Programming, Vol. III*. The MIT Press.
- [Jennings, 1996] Jennings, N. (1996). Coordination techniques for distributed artificial intelligence. In O'Hare, G. and Jennings, N., editors, *Foundations of Distributed Artificial Intelligence*. John Wiley & Sons.
- [Jennings et al., 1998] Jennings, N., Sycara, K., and Wooldridge, M. (1998). A roadmap of agent research and development. *Int. Journal of Autonomous Agents and Multi-Agent Systems*, 1(1):7–38.
- [Jennings and Wooldridge, 1995a] Jennings, N. R. and Wooldridge, M. J. (1995a). Applying agent technology. *Applied Artificial Intelligence: An International Journal*, 9(4).
- [Jennings and Wooldridge, 1995b] Jennings, N. R. and Wooldridge, M. J. (1995b). Intelligent agents: Theory and practice. *The Knowledge Engineering Review*, 10(2).
- [Kantor, 1959] Kantor, J. R. (1959). *Interbehavioral Psychology: A Sample of Scientific Science Construction*. Principia.
- [Karnik and Mendel, 1999] Karnik, N. and Mendel, J. M. (1999). Applications of type-2 fuzzy logic systems to forecasting of time-series. *Information Sciences*, 120:89–111.

- [Kim and Lee, 1999] Kim, I. and Lee, S.-R. (1999). A fuzzy time series prediction method based on consecutive values. In *1999 IEEE International Fuzzy Systems Conference Proceedings*, pages 703–707, Seoul, Korea.
- [Kolarik and Rudorfer, 1994] Kolarik, T. and Rudorfer, G. (1994). Time series forecasting using neural networks. In *Proceedings of the international conference on APL : the language and its applications*, pages 86–94, Antwerp, Belgium. ACM.
- [Kosanović et al., 1996] Kosanović, B., Chaparro, L., and Sclabassi, R. (1996). Signal analysis in fuzzy information space. *Fuzzy Sets and Systems*, 77(1):49–62.
- [Koza, 1998] Koza, J. (1998). Genetic programming. In Williams, J. G. and Kent, A., editors, *Encyclopedia of Computer Science and Technology*. M.I.T. Press.
- [Kutsurelis, 1998] Kutsurelis, J. (1998). Forecasting financial markets using neural networks: An analysis of methods and accuracy. Master’s thesis, Naval Postgraduate School, Monterey, California, U.S.A.
- [Last et al., 2001] Last, M., Klein, Y., and Kandel, A. (2001). Knowledge discovery in time series databases. *IEEE Transactions on Systems, Man and Cybernetics-Part B: Cybernetics*.
- [Lawrence et al., 1996] Lawrence, S., Chung-Tsoi, A., and Lee-Giles, C. (1996). Noisy time series prediction using symbolic representation and recurrent neural network grammatical inference. Technical Report UMIACS-TR-9627, Institute for Advanced Computer Studies, University of Maryland, U.S.A.
- [Lee-Giles et al., 2001] Lee-Giles, C., Lawrence, S., and Chung-Tsoi, A. (2001). Noisy time series prediction using a recurrent neural network and grammatical inference. *Machine Learning*, 44(1/2):161–183.
- [Magdon-Ismail et al., 1998] Magdon-Ismail, M., Nicholson, A., and Abu-Mostafa, Y. (1998). Estimating model limitations in financial markets. *Proceedings of the IEEE*, 86(11).
- [Makridakis et al., 1983] Makridakis, S., Wheelwright, S., and McGee, V. (1983). *Forecasting: methods and applications. 2a Ed.* John Wiley & Sons, Inc.
- [Mandischer, 1993] Mandischer, M. (1993). Representation and evolution of neural networks. In Albrecht, R. F., Reeves, C. R., and Steele, N. C., editors, *Proceedings of the International*

- Conference on Artificial Neural Nets and Genetic Algorithms*, pages 643–649, Wien and New York. Springer.
- [Merani, 1976] Merani, A. (1976). *Diccionario de Psicología*. Ed. Grijalbo.
- [Müller, 1996] Müller, H. (1996). Negotiation principles. In O'Hare, G. and Jennings, N., editors, *Foundations of Distributed Artificial Intelligence*, pages 211–229. John Wiley & Sons.
- [Moody, 1995] Moody, J. (1995). Economic forecasting: Challenges and neural network solutions. In *International Symposium on Artificial Neural Networks, Taiwan*.
- [Morrison, 1991] Morrison, F. (1991). *The Art of Modeling Dynamic Systems. Forecasting for Chaos, Randomness, & Determinism*. Multiscience Press, Inc.
- [Moulin and Chaib-Draa, 1996] Moulin, B. and Chaib-Draa, B. (1996). An overview of distributed artificial intelligence. In O'Hare, G. and Jennings, N., editors, *Foundations of Distributed Artificial Intelligence*, pages 3–55. John Wiley & Sons.
- [Mukherjee et al., 1997] Mukherjee, S., Osuna, E., and Girosi, F. (1997). Nonlinear prediction of chaotic time series using support vector machines. In Principe, J., Giles, L., Morgan, N., and Wilson, E., editors, *(IEEE) Workshop on Neural Networks for Signal Processing (VII)*, page 511. IEEE Press.
- [Ndumu and Nwana, 1997] Ndumu, D. and Nwana, H. (1997). Research and development challenges for agent-based systems. *IEE/BCS Software Engineering Journal: special issue on Agents Technology*, 144(1).
- [Nolfi and Floreano, 1999] Nolfi, S. and Floreano, D. (1999). Learning and evolution. *Autonomous Robots*, 7(1).
- [Nolfi and Parisi, 1997] Nolfi, S. and Parisi, D. (1997). Learning to adapt to changing environments in evolving neural networks. *Adaptive Behavior*, 5(1):75–98.
- [Nwana et al., 1996] Nwana, H., Lee, L., and Jennings, N. (1996). Coordination in software agent systems. *BT Technology Journal*, 14:4.
- [Ortega et al., 2001] Ortega, J., Koppel, M., and Argamon, S. (2001). Arbitrating among competing classifiers using learned referees. *Knowledge and Information Systems*, 3(4).

- [Ozawa and Niimura, 1999] Ozawa, K. and Niimura, T. (1999). Fuzzy time-series model of electric power consumption. In *Proceedings of the 1999 IEEE Canadian Conference on electrical and Computer Engineering*, pages 1195–1198, Alberta, Canada.
- [Ozawa et al., 1997] Ozawa, K., Watanabe, T., and Kanke, M. (1997). Fuzzy auto-regressive model and its applications. In Jain, L., editor, *1997 First International Conference on Knowledge-Based Intelligent Electronic Systems*, pages 112–117, Adelaide, Australia.
- [Papageorgiou, 1997] Papageorgiou, C. (1997). High frequency time series analysis and prediction using markov models. In *Proceedings of the Conference on Computational Intelligence for Financial Engineering*, pages 182–185.
- [Piaget, 1983] Piaget, J. (1983). *La psicología de la inteligencia*. Ed. Crítica, España.
- [Plaza and Ontañón, 2003] Plaza, E. and Ontañón (2003). Cooperative multiagent learning. In *Adaptive Agents and Multi-Agent Systems, Lecture Notes on Artificial Intelligence 2636*, pages 1–17. Springer-Verlag.
- [Ridley, 2003] Ridley, M. (2003). *Evolution, 3rd edition*. Blackwell Publishers.
- [Romero Cortés and Sheremetov, 2001] Romero Cortés, J. and Sheremetov, L. (2001). Model of cooperation in multi agent systems with fuzzy coalitions. In *Lecture Notes In Computer Science archive. Revised Papers from the Second International Workshop of Central and Eastern Europe on Multi-Agent Systems: From Theory to Practice in Multi-Agent Systems*, pages 263–272. Springer-Verlag.
- [Rumelhart et al., 1986] Rumelhart, D., Hinton, G., and Williams, R. (1986). Learning representations by back-propagation errors. *Nature*, 323(9):533–536.
- [Soto and Núñez, 2003] Soto, R. and Núñez, G. (2003). Dynamical fuzzy sets for time series forecasting. In *IASTED International Conference on Modelling and Simulation (MS 2003)*.
- [Stone and Veloso, 2000] Stone, P. and Veloso, M. (2000). Multiagent systems: A survey from a machine learning perspective. *Autonomous Robots*, 8(3):345–383.
- [Tsai and Wu, 1999] Tsai, C. and Wu, S. (1999). Da study for second-order modeling of fuzzy time series. In *Proceedings of 1999 IEEE International Fuzzy Systems Conference*, pages 719–725.

BIBLIOGRAFÍA

- [Tudhope and Taylor, 1996] Tudhope, D. and Taylor, C. (1996). A unified similarity coefficient for navigating through multi-dimensional information. In *Texture 2003. The 3rd international workshop on texture analysis and synthesis*, pages 1–6, Nice, France.
- [Velazquez, 2002] Velazquez, A. (2002). *Localización, recuperación e identificación de la capa de caracteres, contenida en los planos cartográficos*. PhD thesis, Instituto Politécnico Nacional, México.
- [Weigend et al., 1991] Weigend, A., Huberman, B., and Rumelhart, D. (1991). Predicting sunspots and exchange rates with connectionist networks. In Eubank, S. and Casdagli, M., editors, *Proceedings of the 1990 NATO Workshop on Nonlinear Modeling and Forecasting*. Addison-Wesley.
- [Yager and Filev, 1994] Yager, R. and Filev, D. (1994). *Essentials of Fuzzy Modeling and Control*. John Wiley & Sons.
- [Yang and Zou, 2004] Yang, Y. and Zou, H. (2004). Combining time series models for forecasting. *International Journal of Forecasting*, 20:69–84.
- [Yao and Poh, 1995] Yao, J. and Poh, H. (1995). Equity forecasting: A case study on the klse index. In *Proceedings of the 3rd International Conference On Neural Networks in the Capital Markets NNCM'95*.
- [Yao et al., 1996] Yao, J., Poh, H., and Jasic, T. (1996). Foreign exchange rates forecasting with neural networks. In *International Conference on Neural Information Processing*, pages 754–759.
- [Yao and Tan, 2000] Yao, J. and Tan, C. (2000). A case study on using neural networks to perform technical forecasting of forex. *Neurocomputing*, 34(1-4):79–98.
- [Zhang, 1999] Zhang, B.-T. y Cho, D.-Y. (1999). Co-evolutionary fitness switching: Learning collective behaviors using genetic programming. In Spector, L., Langdon, W. B., O'Reilly, U.-M., and Angeline, P. J., editors, *Advances in Genetic Programming, Vol. III*, pages 425–445. MIT Press.