UNIVERSIDAD DE GRANADA
Departamento de Ciencias de la Computación
e Inteligencia Artificial

MODELLING TIME SERIES THROUGH FUZZY RULE-BASED MODELS: A STATISTICAL APPROACH

TESIS DOCTORAL
presentada para la obtención del
GRADO DE DOCTOR
por

José Luis Aznarte Mellado

Granada, octubre de 2008

Editor: Editorial de la Universidad de Granada
Autor: José Luis Aznarte Mellado
D.L.: Gr. 2592-2008
ISBN: 978-84-691-7892-8

Modelling time series through Fuzzy Rule-based Models: A statistical approach

Esta memoria, titulada **MODELLING TIME SERIES THROUGH FUZZY RULE-BASED MODELS: A STATISTICAL APPROACH** (es decir **MODELADO DE SERIES TEMPORALES MEDIANTE SISTE-MAS BASADOS EN REGLAS DIFUSAS: UN ENFOQUE ESTADÍSTICO**) es presentada por D. José Luis Aznarte Mellado con el fin de optar al grado de Doctor y ha sido realizada en el Departamento de Ciencias de la Computación e Inteligencia Artificial de la Universidad de Granada bajo la dirección del Doctor D. José Manuel Benítez Sánchez.

Lo cual, para que conste, firmaron ambos en Granada,
siendo de 2008 el día 11 del mes de septiembre.

El Director

Fdo. J.M. Benítez

El Doctorando

Fdo. J.L. Aznarte M.

Agradecimientos

Hay dos personas que van a descansar incluso más que yo al ver este trabajo terminado, y es para ellas desde la primera a la última página: son mis padres.

Por otro lado, como es obvio, en el desarrollo del contenido de esta memoria ha tenido mucho que decir mi director el Prof. Dr. D. José Manuel Benítez, y el entorno de trabajo que la vio nacer: el Departamento de Ciencias de la Computación e Inteligencia Artificial de la Universidad de Granada. Igualmente es preciso reconocer la comprensión del Prof. Dr. D. Javier Almendros y del resto del personal del Instituto Andaluz de Geofísica y Prevención de Desastres Sísmicos.

Además, ha sido fundamental el apoyo y la ayuda del Prof. Dr. D. Marcelo Medeiros, del Departamento de Economía de la *Pontifícia Universidade Católica* de Río de Janeiro. Junto a la hospitalidad de esta institución brasileña, la del *Econometrics Institute* de la Universidad Erasmus, de Rotterdam, personificada en el Prof. Dr. D. Dick van Dijk contribuyó lo suyo al desarrollo de las ideas contenidas en esta tesis. Éste último, junto al Prof. Dr. D. Philip Hans Franses, merecen un agradecimiento especial por su colaboración en los trámites necesarios para la obtención de la Mención Europea. También es de agradecer el desinterés y la amabilidad del grupo de investigación en Aerobiología de la Universidad de Granada a la hora de ceder sus datos para este estudio y lo mismo debe ser dicho de Gregorio I. Sáinz Palmero, de la E.T.S. Ingenieros Industriales de la Universidad de Valladolid y de la Fundación CARTIF.

Finalmente, las atentísimas correcciones y comentarios del Prof. Dr. D. Igor Zwir y la Profa. Dra. Dña. Cristina Rubio contribuyeron también a mejorar el resultado final.

Así como en lo suyo y por su parte contribuyeron el Dr. D. Diego Nieto y el Dr. D. Pedro Álvarez, que representan aquí a otras muchas personas amigas, insustituibles todas en tantas horas de las de trabajo y de las otras, aquellas más nocturnas, más vivas y menos sometidas. Aquellas que acaso sean las únicas perdurables.

Índice / Contents

List of Figures

List of Tables

1. Introduction

This first chapter offers a general description of the problem, addressing time series analysis from a historical perspective and the need for communicating two separated research fields which work on the same problem. The motivation and objectives of this work are also stated in this chapter. Finally there is a brief bibliographic review and the structure of this document is explained.

1.1. Presentation

Time series analysis is a prominent area within mathematical statistics, data analysis, stochastic finance and econometrics. During the last years, it has been a prolific field of study in terms of research and applications, and many advances have been published. Nevertheless, there is not a widely accepted methodology and much research is still to be done.

Time series analysis has three goals: *forecasting*, *modelling*, and *characterisation*. The aim of forecasting (also called predicting) is to accurately predict the short-term evolution of the system; the goal of modelling is to find a description that accurately captures features of the long-term behaviour of the system. These are not necessarily identical: finding governing equations with proper long-term properties may not be the most reliable way to determine parameters for good short-term forecasts, and a model that is useful for short-term forecasts may have incorrect long-term properties. The third goal, system characterisation, attempts with little or no a priori knowledge to determine fundamental properties, such as the number of degrees of freedom of a system or the amount of randomness. This overlaps with forecasting but can differ: the complexity of a model useful for forecasting may not be related to the actual complexity of the system.

Applications of time series analysis and forecasting methods are found in a

wide variety of areas of human knowledge: signal processing, industrial process control, econometrics, meteorology, physics, biology, medicine, oceanography, seismology, astronomy, psychology... Moreover, in most of these disciplines it is crucial to obtain reliable predictions of future values of different features under study, as this allows for the possibility of optimally controlling future conditions of the global process in which they are subsumed.

Although the need for forecasting is as old as the application areas mentioned above, time series analysis as a discipline is established in the past century. Before the 1920s, forecasting was done by simply extrapolating the series through a global fit in the time domain. The beginning of "modern" time series prediction might be set at 1927 when Yule invented the autoregressive technique in order to predict the annual number of sunspots. His model predicted the next value as a weighted sum of previous observations of the series. In order to obtain interesting behaviour from such a linear system, outside intervention in the form of external shocks must be assumed. For the half-century following Yule, the reigning paradigm remained that of linear models driven by noise.

However, the realisation that apparently complicated time series can be generated by very simple equations pointed to the need for a more general theoretical framework for time series analysis and prediction.

Two crucial developments occurred around 1980; both were enabled by the general availability of powerful computers that permitted much longer time series to be recorded, more complex algorithms to be applied to them, and the data and the results of these algorithms to be interactively visualised. The first development, state-space reconstruction by time-delay embedding, drew on ideas from differential topology and dynamical systems to provide a technique for recognising when a time series has been generated by deterministic governing equations and, if so, for understanding the geometrical structure underlying the observed behaviour. The second development was the emergence of the field of machine learning, typified by neural networks, that can adaptively explore a large space of potential models. With the shift in artificial intelligence towards data-driven methods, the field was ready to apply itself to time series, and time series, now recorded with orders of magnitude more data points than were available previously, were ready to be analysed with machine-learning techniques requiring relatively large data sets.

1.2. Motivation

Model building is one of the basic steps in time series analysis. These models have historically been linear and of a regression type. This approach is not always realistic, and recently non-linear models have increasingly been used. The model has to be estimated using the given data, either parametrically or non-parametrically. Prior to the estimation phase there is an identification phase, where it is determined what kind of model should be used and how long memory this model should possess. Finally there is a diagnostic checking phase to check whether the model works satisfactorily.

The statistical approach relies heavily in the formal application of these three steps, *identification*, *estimation* and *evaluation*. It develops tests guaranteeing that the resulting model will have good statistical properties, and gives formal proofs of them. This yields a sound mathematical foundation to time series forecasts, and allows for a priori knowledge about the capabilities of a given method.

Notwithstanding, the formal proof of the properties of a method usually requires setting some restrictions which can sometimes be unrealistic. These formal requirements include hard to prove conditions such as compactness of the parameter space. To obtain acceptable results in real world problems, these requirements are sometimes ignored or tempered.

Soft Computing, on the contrary, is driven by a more pragmatic approach which tries to obtain feasible solutions. Mainly due to the complexity of the models, usually no formal proofs are derived and hence no restrictions are stated. Having fewer (or no) a priori requirements makes Soft Computing methods more widely applicable and easier to use. Notwithstanding, statistical inference should be applied to them more often.

Obtaining methods which conjugate the advantages of both fields (statistical time series analysis and Soft Computing) is the main motivation of this work. A collaborative perspective —instead of the confrontation which has historically run the relationships between the two areas— could derive important benefits to time series forecasting, and this is the core reason for this work.

1.3. Objectives

The main objectives of this work are:

- To search for formal relationships amongst models from the Statistical Time Series Analysis area and the Soft Computing area.
- To exploit the existing relationships aiming towards a fruitful exchange of knowledge between both areas.

As a secondary objective, the application of contributions to the resolution of real world time series problems will be also considered.

1.4. Previous works

"Classical" bibliography on time series study conforms a considerably long list, and new reference volumes are still being published every day. Notwithstanding, we can cite [8, 70, 4] as volumes amongst the most referenced regarding the linear approach to time series. With a more modern perspective, Tong's book [86] is one of the most influential texts about nonlinear modelling of time series.

Not many researchers from the statistical perspective have approached Soft Computing nor its relationships with statistical methods. There is an exception for this statement: neural networks have become very popular also in the statistical field, being widely used for time series analysis. In [103] there is a thorough review of the state of the art (published in a statistical journal and from a statistical point of view), which include many comparison works between models. Another interesting review is [72].

Fewer are the works in this area devoted to the other Soft Computing components. Concretely, fuzzy logic or neuro-fuzzy models are mentioned in [21, 75, 27, 57].

In the Soft Computing field, researchers have approached Time Series Analysis or Forecasting problem mainly in two different fashions. On the one hand, some researchers saw in Time Series a huge collection of big data sets, that could be used to test new or existing models. This approach is fair, but ignores the special features that distinguish a time series from other sources of data, disregarding at the same time all the scientific knowledge gathered through years

for this specific problem. Examples that illustrate this situation are the tests performed with models like ANFIS [39, 40, 1], EFuNN [43, 44] or ANNBFIS [56]. More examples can be found in [31, 46, 67, 60, 55].

On the other hand, there are papers that present Soft Computing-based models tailored to model and forecast specific time series. Electric load forecasting is one of the most faced problems [42, 45, 82, 18], but there is a great number of other examples: financial forecasting [51, 50, 88], biological forecasting [29, 68] etc. These researchers try to model or predict real world cases using generic models and adapting them to some observable features of the data, but still do not make full use of the tools provided by the classical time series analysis (as statistical inference, for instance).

1.5. Structure of the document

This document is structured as follows. Chapter 2 describes some of the Soft Computing techniques applied to time series analysis. In particular, it covers the widely known Artificial Neural Network (ANN) model and the Fuzzy Rule Based Models (FRBM).

Chapter 3 reviews the classical tools for time series analysis, as Box Jenkins methodology and Holt Winters smoothing. As well, a brief description of the threshold based autoregressive family of methods and some other recent developments is given.

In Chapter 4 we identify and study some of the relationships existing amongst some of the models described. In particular, the relationship between the AR models and fuzzy rules is stated, and this gives place to some other important links amongst models which are explored.

Chapter 5 unfolds the application of the results contained in Chapter 4. The statistical properties of the FRBM are established, we introduce a linearity test against the model and some diagnostic tests, and a hybrid incremental modelling cycle is presented.

These results are put to practice in Chapter 6, where a Montecarlo experiment is run together with the application of the theoretical results to real world series.

Finally, Chapter 7 is devoted to summarise the main contributions of this work and to state some future research lines that will be explored.

2. Fuzzy Rule-based Models

This chapter describes succinctly the main concepts related to the Fuzzy Rule-based Models used in this work. After reviewing some ideas about system identification, we will expose the concept of fuzzy rule and the systems based on it, together with some general considerations about design of fuzzy rule based models.

2.1. System Identification

The goal of Science is to reach a complete knowledge of the systems that surround us, so that we can understand them and predict its behaviour. Not surprisingly, though, reality is usually too complex and the scientist has to approach it through simplified descriptions. There is a clear need to consider essential aspects of the systems under study and disregard other, less important details. These simplified descriptions are referred to as *models*.

There are many different classes of models, but the steady advance of knowledge is only possible if the models are well founded in a suitable formal development and a rigorous perspective. Hence, one important objective of Science is stated as the obtaining of mathematical models that reflect the existent relationships between the different variables that play a role in the behaviour of a system.

The task of establishing a mathematical model of a system is called *system identification*. It is currently carried out in two steps: modelling and estimation. We first define the structure of the model that we are associating with the system. This decision is fundamental and depends, essentially, on the complexity of the system under study and the expected use for the model, which on turn implies many other aspects as the approximation degree, the tractability etc. Once the model is fixed, it is necessary to adjust its parameters so that the system description is as suitable as possible.

This identification process is usually divided into five elements:

a) a set of data samples of the behaviour of the system, and information coming from experts on it,

b) a set of candidate models,

c) some criteria to evaluate different models,

d) a way of validating the chosen model against observed data which are independent of those used in a), and

e) the evaluation of the model.

Identification methods start from a sample set of the behaviour of the system to be modelled. The obtaining of this data set is fundamental to guarantee that the final model is useful. First we must establish which are the variables affecting the system, and which are the system's outputs for a given set of inputs. It is important to perform a good selection, as we must avoid introducing useless or redundant information which can only result in noise, as well as not considering crucial variables whose effect can be determinant. The collected data must cover all the domain for each variable, or otherwise the model will not be robust against situations not considered in the sample set.

The selection of the model to be used is also fundamental in the identification process. It basically depends on the complexity of the system under study and the expected use for the model, considering aspects as theoretic tractability, desired degree of approximation, implementation difficulty and efficiency, robustness against noisy inputs etc. In principle, the set of available models is infinite. We will always aim for the simpler and more efficient one which still satisfies the other restrictions.

When there is not much information about the system (apart from the sample set of behaviour), it can be seen as a black-box. For identification of such systems, it is sometimes preferred to use a family of models satisfying the "Universal Approximation" property, which guarantees the existence of at least one member of the family which approximates the behaviour of the system with arbitrary precision. Knowing some properties of the behaviour of the system is a great help and sometimes is absolutely essential to orient the identification

process. As an example, many systems of practical interest have a continuous mathematical behaviour. Not considering this fact makes the identification problem practically unsolvable.

Once the model structure is fixed, it will be necessary to adapt it to the system under study. The model depends on a parameter set whose values control its behaviour. The aim is, hence, to estimate the value of those parameters so that the resulting model is the "best". This requires establishing a criterion which allows to measure the quality of the adjustment of the model to the system, and to optimise this criterion. Usually, it is a measure of the error and it is minimised, as in the common case where the criterion is the sum of squared differences between the dependent variables that the model generates and the real outputs observed in the system. This criterion is used because its easy mathematical tractability, not because it is the best criterion or the most interesting one.

Notwithstanding, it is not convenient to limit the criteria to the error measure, and other criteria can be defined. For example, in the case of a dynamical system, additional error measures involving temporal evolution can be used. In many cases, the model is a functional relationship between variables, and adjustment consists in determining the optimal values for a set of numeric parameters. Then, the estimation is performed through a numeric optimisation method, generally iterative.

When the model structure and its associated parameters have been estimated, it is necessary to validate the model. Generally, validation is carried out by trying the model on data which have not been used in its estimation process. Validation is necessary, because the model can adjust very well to all the data used during its design process if it has enough freedom degrees, and, in spite of that, have a very bad response to previously unseen situations. This phenomenon is known as *over training* and must be avoided. Directly related to this problem, it is important to considerate that no model can be (and no model should be) a perfect description of the system as in that case the complexity of the model would be equivalent to that of the system and hence the model would not be useful. The objective is to create a model which is a sufficient description of some particular aspects of the system, according to some criterion.

The models resulting from an identification problem can be useless for many reasons: maybe the available data were not informative enough, or even irrelevant, maybe they are too noisy or they do not cover the system's whole input

space; it can be the case that the numeric optimisation methods used were not powerful enough so as to find an optimal set of values according to the evaluation criterion; this criterion can be inadequate or even the mere structure of the model can be inappropriate for the description of the system. In any of these cases, it is necessary to identify the cause for deficiencies and proceed to its resolution. This leads to iterative identification methodologies, where, starting from an initial model built according to hypotheses about the system, there are successive steps which modify the model trying to correct deficiencies of previous versions of it.

2.2. Fuzzy Rule based Models

The classical model for representation and processing of knowledge is Aristotelian Logic. Extended into Propositional Logic and Predicate Logic, it covers many aspects of human knowledge, but it fails against many other inherent characteristics, as, for example, uncertainty, vagueness, incomplete information etc. Many variants of logic have been proposed in order to tackle the resolution of these problems, but none obtained entirely satisfactory results.

One logical construct used in this context is the *production rule*. These rules are defined as

> IF a set of conditions is satisfied, THEN it is possible to infer a set of conclusions.

The first part of this rule is known as *antecedent*, condition or just left part. The second is known as *consequent*, conclusion or right part. There are many advantages of using this knowledge representation model:

1. It is a natural way to code knowledge. It is widely applicable, covering many domains whose knowledge is expressed in terms of cause-effect relations.

2. It is a mixed representation model, as it has a declarative part as well as a procedural one. The mix is such that the union of advantages is superior to the union of disadvantages of both approaches.

3. There exists a very old model to represent the rule: the logic implication. Even though the meaning of the rule does not correspond exactly to the material implication of classical logic.

All these advantages have produced an enormous spread of systems based on these rules, called *production systems*. The vast majority of expert systems developed and, over all, those exploited in practice are production systems, which makes them the main paradigm of knowledge based system. The application of these systems has been an enormous success in many fields. Notwithstanding, they are not free of disadvantages, mainly related with the precise way of dealing with a knowledge which, usually, comes from an imprecise source (human experts).

The development of knowledge-based models implies several stages, of which *Knowledge Acquisition* represents the main bottle-neck. Its objective is the extraction of rules and heuristics used by the expert to solve the problems he or she faces, and represent them in an adequate form to process them using computers. Traditionally it is carried out using manual techniques, consistent in several interviews between the knowledge engineer and the expert. But these procedures do not always yield the desired results, owing to different causes, amongst them the difficulty of experts to verbalise their knowledge.

To improve the Knowledge Acquisition process, its automation has been studied. Many researchers have devoted to this task obtaining diverse results. One of the main ways to tackle this problem is called Machine Learning. The reason is simple: maybe the expert finds it difficult to verbalise and transmit its knowledge, but he or she can always apply them to solve problems in his or her area. In this situation, the knowledge engineer can observe and take notes of the expert's work. Furthermore, it is frequent to have wide recordings of the expert's acts. If we can obtain a system which learns the behaviour of the expert, we can get a knowledge representation which is understandable by a computer.

Machine Learning is comprised of diverse techniques and algorithms, many of which have been applied to this task of *Knowledge Acquisition*. Two of the most popular techniques in the last years are Genetic Algorithms and Artificial Neural Networks.

Most of the Machine Learning techniques start from a set of examples and they return the knowledge encoded in production rules, as this is a common representation in knowledge-based models. Methods based on these techniques

are called *rule extraction methods* (REM).

When the reasoning model implemented by these REM is based in classical logic, it is crisp, it does not admit any kind of vagueness. On the other hand, humans are perfectly capable of obtaining conclusions out of imprecise statements or facts. This type of reasoning is more qualitative than quantitative and drives far from the framework of classical logic. It is the base for the human capabilities to understand natural language, interpret manuscripts, develop tasks requiring training or mental abilities, and, definitely, take rational decisions in complex or uncertain situations. This type of reasoning is known as *Approximate Reasoning*.

The principles of Approximate Reasoning are clearly opposed to the quantitative and precise tradition of Science. This prevented scientists to devote enough attention to it up to the 1970 decade. Formalisation of Approximate Reasoning through similar rules as those used by classical logic has captured a great deal of attention in the last years, together with a growing interest in every attempt to reach better descriptions of the complex processes underlying reasoning and decision-making.

Fundamental concepts to formalise Approximated Reasoning are found in Fuzzy Set Theory, and more precisely in Fuzzy Logic, making L.A. Zadeh a pioneer in such formalisation.

2.2.1. Fuzzy Rules

L.A. Zadeh, in [98], introduced a new perspective for Science when proposed that we should not try to escape to uncertainty or vagueness, but to search for means to represent it which allowed its use and control. Zadeh realised that the classical concept of set did not properly represent many of the sets that we use normally. Take, as an example, the young people set. A person aged 18 is clearly classified as young. Someone aged 40 is also clearly not young. A person aged 27 is young, but less young than someone aged 25, and younger than someone in its 30s. But, where is the limit between being young and not being young? We cannot establish a clear limit point, but the property “being young” is gradual. Youth cannot be defined in a precise way, it suffers from uncertainty because it is vague. The transition between fulfilling the property and not fulfilling it is not crisp, but smooth, continuous. This gradual way of fulfilling a property cannot

be represented via a classical set in which given an element it either belongs to the set or not.

Traditional methods dealing with information, based on classical logic, act on precise data. They are not valid for reasoning processes involving vague, uncertain or imprecise concepts. The automatic treatment of this information requires another type of techniques. Zadeh contribution is a set of such techniques whose algebraic formulation was stated in the *Fuzzy Set Theory*.

The main concept of this theory is that of *fuzzy set*, which includes sets whose borders are not defined precisely. Formally, a fuzzy set[1] A of a reference set or domain U is defined as a set whose indicator function takes values in $[0,1]$ instead of in $\{0,1\}$, as is the case in classical sets. In this way, we say an element belongs to a set to a certain extent of membership, whose values constitute a continuous range. The indicator or characteristic function, usually called *membership function*, is represented as μ:

$$\mu_A : U \to [0,1].$$

Given an element $x \in U$, its membership degree to the fuzzy subset A is $\mu_A(x)$. It is immediate to observe that an ordinary set (a crisp set) is an special, extreme case of a fuzzy set.

Starting from this definition of set, Zadeh extended other concepts related with it, as the union, intersection and complement operations, the Cartesian product, relations etc., up to establishing a full theory about these sets. This extension is such that when the involved sets are crisp, then the operations also reduce to the classical ones. As well, Zadeh established a logic associated to this set theory, the Fuzzy Logic, which offers an alternative to classical logic to deal with knowledge affected by uncertainty and vagueness. There are many texts devoted to a detailed exposition of the Fuzzy Set Theory and its main applications [23, 48].

Nowadays computers are not very effective in dealing and reproducing human behaviours and reasoning styles. Zadeh interprets this fact as a sign of what he called “Incompatibility Principle”, which states that precision and complexity are incompatible properties. This way, conventional techniques inspired in the precise manipulation of numerical data are intrinsically insufficient to model

[1] The proper term is *fuzzy subset*, but the simpler *fuzzy set* is predominant in the literature.

human knowledge and complex decision making processes. Fuzzy Set Theory, according to Zadeh, can model this type of information.

People used to work with sets are able to capture and understand the functional or graphical descriptions of sets. This is not usually the case with humans used, in general, to express and exchange information in a linguistic manner. This reason and again the "Incompatibility Principle" lead to thinking of representing uncertainty or imprecise information through linguistic labels or terms, which would make decision-makers work easier.

This observations were the starting point from which Zadeh introduced the concept of linguistic variable [100, 101, 102]. In an informal way, we can define it as a variable over a discourse domain which takes values in a set of linguistic labels. Those labels have a semantic interpretation defined through a fuzzy set over the discourse domain. Formally:

A *linguistic variable* is a quintuple $(x, T(x), U, G, M)$, where: x is the name of the variable; $T(x)$ is the set of linguistic terms of x; U is the universe of discourse; G is the grammar generating terms in $T(x)$; and M is a semantic rule associating a meaning, $M(t)$, to each $t \in T(x)$, where $M(t)$ is a fuzzy subset of U.

In principle, the number of elements in $T(x)$ is arbitrary, but if it increases a lot, it will make the semantic distinguishably of some of them difficult, as a consequence of the approximate, vague or imprecise nature of the information coded by those terms about x.

In classical logic terms, when we want to represent that an object x satisfies property P, that is, x belongs to the set of objects satisfying P, we write "x is P." If the property is fuzzy, then the set P is fuzzy and the proposition "x is P" is called *fuzzy proposition*.

In order to build more complex statements, in classical logic we use the so called logic connectives: negation, conjunction, disjunction and implication. These operators act by associating a truth value to the complex proposition from the truth values of the simpler propositions that form it. To connect fuzzy propositions, we need to extend the definition of these operators. This has been a wide field of research in fuzzy logic, yielding several different definitions for each operator.

Logic negation has been extended through negation functions, amongst which the most common is $n(x) = 1 - \mu(x)$.

For conjunction there are conmutative, associative and monotone functions,

non decreasing on each argument, called t-norms. The most used functions are minimum and product. The disjunction is modelled through a family of functions known as t-conorms. For each t-norm there exists a dual t-conorm. In this case, the most used are the maximum and the bounded sum, which are duals of the minimum and the product, respectively.

If we replace the propositions in a classical production rule by fuzzy propositions, we get a *fuzzy rule*. In a general version a fuzzy rule takes the form:

$$\begin{aligned}&\text{IF } f_1(x_1,\ldots,x_n) \text{ IS } A^1 \ \Cap_1 \ \cdots \ \Cap_{k-1} \ f_k(x_1,\ldots,x_n) \text{ IS } A^k \\ &\qquad\qquad\qquad\qquad \text{THEN} \\ &\qquad g_1(y_1,\ldots,y_m) \text{ IS } B^1 \ \Cup_1 \ \cdots \ \Cup_{j-1} \ g_j(y_1,\ldots,y_m) \text{ IS } B^j \end{aligned} \quad (2.1)$$

with f_i, g_l functions, $\Cap_i, \Cup_l$ logic connectives and A^i, B^l, linguistic labels or fuzzy sets.

But this rule is too complex for being used in an effective manner. Usually, the f_i are the projections and the A^i y B^j are fixed fuzzy sets, taking the form:

$$\begin{aligned}&\text{IF } x_1 \text{ IS } A_1 \ \Cap_1 \ x_2 \text{ IS } A_2 \ \Cap_2 \ \ldots \ \Cap_{n-1} \ x_n \text{ IS } A_n \\ &\qquad\qquad\qquad\qquad \text{THEN } y_1 \text{ IS } B_1 \ \Cup_1 \ \ldots \ \Cup_{m-1} \ y_m \text{ IS } B_m \end{aligned} \quad (2.2)$$

with $\Cap_i$ and $\Cup_j$ conjunctive or disjunctive operators. In particular, in the antecedent, $\Cap_i$ are usually conjunctions.

Amongst the most important properties of fuzzy rules we have:

- Uncertainty representation. Fuzzy rules allow to gather vague concepts. This is why they are closer to the human way to deal with information than the classical production rules.

- Compact information representation model. With just one fuzzy rule we can express all the information contained in a set of classical rules.

- Local character. The information described in a single fuzzy rule usually affects only a local zone of the complete domain of the problem. Its interactions with other descriptive elements of the problem is restricted to those which are in its neighbourhood. This eases construction and interpretation of fuzzy rules.

2.2.2. Fuzzy Inference

The Compositional Rule of Inference was introduced by Zadeh in 1973 [99] as a tool to translate the classical logic "modus ponens" into Fuzzy Logic. It has been later formalised and generalised as an inference method, so the original formulation is now a particular case.

Modus ponens, the basic deduction rule in Predicate Calculus, is the best known inference method and has been widely applied in the Artificial Intelligence field. In short it can be described as:

Assuming that implication "If P then Q" is true, and that P occurs (i.e. P is true), then we conclude that the fact or proposition Q is also true.

$$\begin{array}{lll} P & \rightarrow & Q \\ P & & \\ \hline & & Q \end{array}$$

In many cases, P y Q contain knowledge about variables. The simpler case is that in which P and Q are statements about two variables, that is, P is the proposition "x is A" and Q corresponds to "y is B", where x and y are variables taking values in universes U and V, not necessarily different, while A and B are still properties about the values of x and y. Now, from the rule "If x is A then y is B", and from "x is A", we can infer the fact "y is B".

From an point of view Approximate Reasoning, we are interested in the situation in which we want to infer when the available information is imprecise, incomplete or not totally trustable, that is, when we deal with fuzzy predicates. Fuzzy logic offers an appropriate context to deal with uncertainty, because in contrast with traditional logic systems, its main objective is inference from imprecise knowledge. For this case, we have the Generalised Modus Ponens, established as:

$$\begin{array}{l} \text{IF } x \text{ IS } A \text{ THEN } y \text{ IS } B \\ x \text{ IS } A' \\ \hline \end{array}$$

where, again, x and y are variables on U and V, but now A, B y A' are fuzzy sets (linguistic properties) on the respective universes of discourse, which can also be considered as fuzzy facts or flexible restrictions related to these variables.

As in classical modus ponens the conclusion states that y is B, being B an ordinary set of V, it is reasonable to admit that in the fuzzy case the conclusion

will be defined by a fuzzy set over the universe of discourse of y, so it will have the form: "y is B'." Hence, Generalised Modus Ponens is:

$$\frac{\begin{array}{l}\text{IF } x \text{ IS } A \text{ THEN } y \text{ IS } B \\ x \text{ IS } A'\end{array}}{y \text{ IS } B'}$$

Now, the main problem is how to obtain the new fuzzy set B'. The Compositional Rule of Inference can be described and justified as follows.

A rule introduces a fuzzy relation R that binds the values of the universes of the variables linked by the rule, that is, a fuzzy set in the cartesian product of the universes of discourse $U \times V$, such as

$$\mu_R(x,y) = F(\mu_A(x), \mu_B(y)),$$

where μ_A and μ_B note the respective membership functions of the fuzzy sets A and B.

The fuzzy set B' must be inducted by A' over y through R. Hence, we can write $B' = A' \circ R$ and the question is how to build F and $\circ$ to obtain B'. To solve these problems there have been several proposals in the literature. All of them are based on the Extension Principle, which in this context is:

$$\mu_{B'}(y) = \max_x(\mu_{A'}(x) * \mu_R(x,y)),$$

being $*$ an associative and monotone operation, non decreasing on each argument (a t-norm). The effective way to realise the inferences is based on the election of F and the t-norm $*$, consequently leading to what could be called different ways of reasoning.

2.2.3. Fuzzy Rule-Based Models

A model using fuzzy rules is called *Fuzzy Rule-Based Model* (FRBM). These models constitute an extension of the models based in rules. Its main field of application is fuzzy modelling, that is, they are used to describe unknown or complex systems. In control field they have proven to be very appropriate, where they are usually called *fuzzy controllers*. Its application to control problems with hard or impossible mathematical solutions has been a milestone for the acceptation and fast expansion of the techniques based in Fuzzy Set Theory.

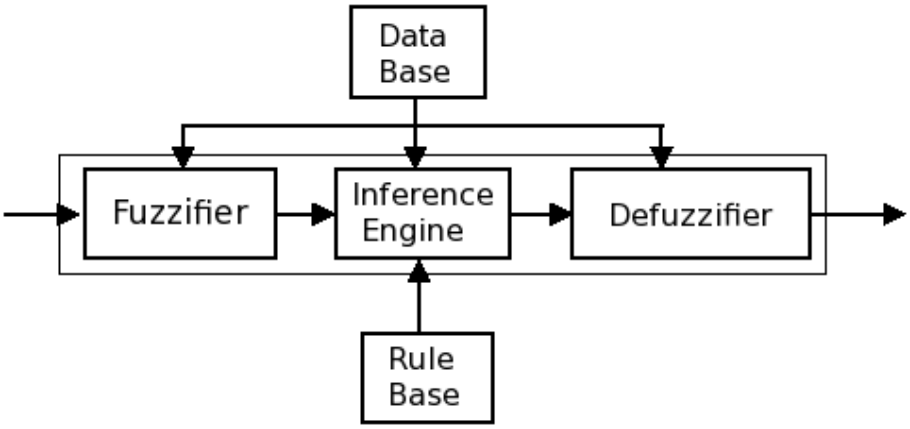

Figure 2.1.: Structure of an FRBM

FRBM are applications of spaces of dimension n to spaces of dimension m. They are formed by four main components, as shown in Fig. 2.1: *fuzzifier*, *knowledge base*, *inference engine* and *defuzzifier*.

The fuzzifier transforms the real input values into fuzzy values, usually in univalued fuzzy sets or singletons.

The knowledge base includes the rule base and the data base. In the first are included all the rules capturing the knowledge of the system. The definitions of the membership functions for the labels of the rules are kept in the data base.

The inference engine computes the fuzzy output from the fuzzy inputs by applying a fuzzy implication function. It also aggregates the outputs of the different applicable rules producing a single fuzzy output value.

Finally, the defuzzifier condenses in a single real number the inferred fuzzy output.

In short, the fuzzy inference process is as follows: Given an input, the *firing strength* of each rule is obtained. This firing strength comes from the matching degree of each component of the input with the corresponding proposition in the fuzzy rule's antecedent. When the inputs are real values and the fuzzification process transforms them into singleton fuzzy sets, this matching degree is just the membership degree of each real input value to the fuzzy set defined by each rule proposition. Then these degrees are aggregated according to the connective

operators linking the propositions. Usually, these are conjunctive operators, so the firing strength of the rule will be the value of the function modelling the conjunction (minimum, product, ...) applied on all the coupling degrees. The fuzzy output of the rule will be its consequent, in which the fuzzy sets will have a maximum membership degree equal to the firing strength of the rule. Last, in the case of systems with real output, the fuzzy output is transformed into real values by applying the correspondent defuzzification method.

FRBM have some interesting properties, amongst which its "Universal Approximation" capability stands out. In [13, 14, 49] several wide classes of fuzzy controllers are proven to be universal approximators. This result entails FRBM as right tools for approximating other systems even when the data associated to them are not fuzzy.

We can distinguish two main applications of FRBM:

1. *Linguistic description* of systems where there are uncertainty or vagueness in the inputs and/or outputs. This also includes the cases in which precision is not important. What is fundamental is to obtain a description of the inner relationships amongst inputs and outputs in an unknown system in terms of words and expressions of the natural language.

2. *System Approximation*. Exploiting its universal approximation property, FRBM are an alternative to classical mathematical models in unknown system approximation. Its strengths are clear when dealing with complex systems, where its simplicity and ease of use made them preferable to other methods.

Mamdani's FRBM

In 1975, E.H. Mamdani [61] shown the first practical application of an FRBM. It was a simplified FRBM devoted to a control task. This model had a fuzzifier, a rule base, a data base and a defuzzifier.

The rules used by this model were of the form:

$$R_i: \text{IF } x_1 \text{ IS } A_1^i \text{ AND } x_2 \text{ IS } A_2^i \text{ AND } \ldots \text{AND } x_n \text{ IS } A_n^i \\ \text{THEN } y_1 \text{ IS } B_1^i \text{ AND } y_2 \text{ IS } B_2^i \text{ AND } \ldots \text{ AND } y_m \text{ IS } B_m^i. \quad (2.3)$$

Their simplicity and ease of interpretation granted these type of rules the category of standard, being the most used rules amongst others.

The inference process uses the minimum as conjunction operator and as well as implication function. Hence, given an input $\mathbf{a} = (a_1, a_2, \ldots, a_n)$, the firing strength of the rule is:

$$\gamma_i = \min(A_1^i(a_1), A_2^i(a_2), \ldots, A_n^i(a_n)) \tag{2.4}$$

The fuzzy outputs of the rules are $B_1^i, B_2^i, \ldots, B_n^i$, cut at the level γ_i. The aggregation operator is modelled as a disjunction, usually the *maximum* operator. As defuzzification interface is common to use the centre of gravity or the maximum average.

As stated above, this is the older and simpler FRBM, which contributed to its success. Moreover, its simplicity makes it really easy and inexpensive to be implemented in hardware, and this favoured it expansion to many applications, ranging from small home appliances to big engineering devices as container cranes or automatic train drivers.

Takagi-Sugeno-Kang's FRBM

In 1985, Takagi, Sugeno and Kang [81, 78, 77] proposed a different FRBM which was much more effective in approximation tasks. Its rules, usually known as TSK rules, were of the type:

$$R_i: \text{IF } x_1 \text{ IS } A_1^i \text{ AND } x_2 \text{ IS } A_2^i \text{ AND } \ldots \text{ AND } x_n \text{ IS } A_n^i \quad \text{THEN } y = p_i(x_1, x_2, \ldots, x_n) \tag{2.5}$$

being $p_i(x_1, x_2, \ldots, x_n)$ a linear function. That is, the output adopts a purely functional shape, which, except for the case base of it being a constant, it has no simple linguistic interpretation.

The firing degree of the rules is obtained in a way similar to the one used in Mamdani's FRBM. The only difference is that the conjunction is modelled through product, instead of minimum.

$$\gamma_i = \prod \left(A_1^i(a_1), A_2^i(a_2), \ldots, A_n^i(a_n) \right) \tag{2.6}$$

The final output of the model is given by:

$$y = \frac{\sum_{i=1}^{r} \gamma_i p_i(a_1, a_2, \ldots, a_n)}{\sum_{i=1}^{r} \gamma_i} \tag{2.7}$$

The main drawback of TSK FRBM is that they usually are less interpretable than Mamdani's, but at the same time they are considered to be more precise. In time series analysis, this type of models are usually preferred, and then they use rules of the form:

$$\text{IF } y_{t-p} \text{ IS } A_1 \text{ AND } y_{t-p+1} \text{ IS } A_2 \text{ AND } \ldots \text{ AND } y_{t-1} \text{ IS } A_p$$
$$\text{THEN } y_t = b_1 y_{t-p} + b_2 y_{t-p+1} + \ldots + b_p y_{t-1} + b_{p+1} \tag{2.8}$$

In this rule, all the variables y_{t-h} are lagged values of the time series, $\{y_t\}$. This means that, in order to train the FRBM, we need to build input-output vectors of the form $(y_{t-p}, y_{t-p+1}, \ldots, y_{t-1}; X_t)$, which implies that, if we had N samples from the series, we would be able to use $N - p + 1$ training data.

Additive Fuzzy Models

Additive Fuzzy Models (AFM), proposed by Kosko [49] are characterised by a different way of performing inference.

In general, an FRBM with rules

$$R_i : \text{ IF } x = A_i \text{ THEN } y = B_i,$$

or

$$R_i : \text{ IF } x = A_i \text{ THEN } y = p(x),$$

fires the rules on a given input and each rule produces an output B_i' or $p(x)$. The final output is obtained applying an aggregation operator which, usually, is modelled via the maximum. In AFM, this final aggregation is modelled through a weighted sum:

$$B = \sum_{i=1}^{r} w_i B_i', \tag{2.9}$$

where w_i are weights associated with the rules, not necessarily equal to the firing strength γ_i. The election of these weights determines the type of inference.

For TSK rules, (2.5), the output takes the form:

$$y = \sum_{i=1}^{r} w_i p_i(x_1, x_2, \ldots, x_n). \tag{2.10}$$

2.3. Design of Fuzzy Rule based Models

In the design of an FRBM it is necessary to specify every element of which it is composed [52, 53]:

1. Linguistic variables.
2. Fuzzy rule base.
3. Inference process.
4. Fuzzifier and defuzzifier.

There are several aspects to take into consideration in this design. Amongst the main ones we have:

- Final use of the FRBM: descriptive or approximation. The final use of the FRBM plays a role in the design, as if we are interested in interpretability over precision, we will have to carefully consider some restrictions on the linguistic labels. Those restrictions are not important if the objective of the model is to approximate a system the best way possible.
- Knowledge acquisition. As knowledge based models, the main bottleneck in the design of an FRBM is the acquisition of the available knowledge. The representation used is the fuzzy rule, so knowledge acquisition methods are also known as rule extraction methods. Building the rule base poses the main challenge when building FRBM
- Design criteria. As any other design process, there must be a series of criteria whose optimisation drive the design process towards the desired goals. Amongst the common criteria we have: quality of approximation, number of rules, complexity of rules, interpretability of rules etc.

Some concrete aspects of the design of the above mentioned elements are briefly exposed below.

2.3.1. Linguistic Variables

In order to fully define a linguistic variable it is necessary to fix every component of the quintuple. The name and the domain of discourse are given by the problem to be faced.

The set of terms must be such that all the domain is covered. When aiming at building models that are manageable by a person, the cardinality of the set of linguistic terms must be limited. Zadeh introduced the term of *granularity* to establish a level of distinction amongst different quantifications of uncertainty, vagueness or approximation contained in the linguistic variables, so that one can properly represent the users' discrimination capacity. The idea is that the number of terms must be determined by the possibility of distinguishing between two different linguistic values. It is obvious that this possibility depends on the characteristics of the problem as well as on the ability of the expert and the final user. The set of labels associated to a high number of linguistic variables is understood as a common structure constituted by two basic terms with opposed semantics and several gradings. Human beings have been proved to manage a maximum granularity of 13 terms.

With respect to the grammar, in practice, the set of values taken by the variable is always the set of terms, so the grammar does not play an important role.

The key issue in designing linguistic labels is establishing its semantics, that is, deciding which fuzzy sets are represented by them. There is no standard criterion to follow in this definition, but most researchers consider the following aspects:

a) Coverage. For each point of the domain, there must be at least one label representing it with a membership degree greater than or equal to $\frac{1}{2}$.

b) Overlapping. Two contiguous labels must cut each other at a height equal to $\frac{1}{2}$.

c) Tractability. Related to the manageability of the labels by automatic computing devices. We expand on this below.

In principle, fuzzy sets can have any membership function. Notwithstanding, the labels will only be manageable if there is a specifically designed computer to deal with fuzzy information. The computers normally available are digital, based in binary logic, and they do not process this information efficiently. That

is the reason why membership functions used in practice have a shape included within a limited group, in which a function is fully defined by a small set of parameters. The most common functions are:

1. Triangular. The function has the shape of a triangle and is described by a tern (a,b,c). Its analitic expression is:

$$\mu_{\Lambda_{(a,b,c)}}(x) = \begin{cases} 0, & x \leq a \text{ or } x > c \\ \frac{x-a}{b-a}, & a < x \leq b \\ \frac{x-c}{b-c}, & b < x \leq c \end{cases} \tag{2.11}$$

2. Trapezoid. The aspect of this function is that of a trapezoid, which is defined by a tuple (a,b,c,d). The membership function is:

$$\mu_{\Pi_{(a,b,c,d)}}(x) = \begin{cases} 0, & x \leq a \text{ or } x > d \\ \frac{x-a}{b-a}, & a < x \leq b \\ 1, & b < x \leq c \\ \frac{x-d}{c-d}, & c < x \leq d \end{cases} \tag{2.12}$$

3. Gaussian. Bell shaped and characterised by two parameters, mean c and deviation σ:

$$\mu_G(x) = \exp\left(-\frac{(x-c)^2}{\sigma^2}\right) \tag{2.13}$$

4. Generalised bell. Used by Jang in the original formulation of his model ANFIS. It is also a bell, but it contains three parameters, a,b,c:

$$\mu_B(x) = \frac{1}{1+\left[\left(\frac{x-c_i}{a_i}\right)^2\right]^{b_i}}, \tag{2.14}$$

5. Logistic. Also called *sigmoid*, it comes from the Artificial Neural Network field, and it is defined by two parameters, γ and c:

$$\mu_S(x) = \frac{1}{1+\exp\left(-\gamma(x-c)\right)} \tag{2.15}$$

Label definition is also determined by the final objective of the FRBM to be designed. If it is a descriptive model, interpretability is most important, and hence the labels must have a comprehensible aspect and their definitions must be the same in every rule where the variable appears. In these models, the definitions of the labels are included in the data base, as they are usually given by the experts.

If the model is of approximating intention, the intelligibility of the membership functions is not so important and the fuzzy values associated to each variable can be different on each rule. The variables loose the linguistic character and adopt a purely fuzzy nature. In these models each rule stores the complete definition of its variables.

In most real-world applications, where computational efficiency is usually very important, triangular or trapezoidal functions are used.

2.3.2. Rule Base

Obtaining the rules that store the knowledge in an FRBM is the fundamental problem in the design process. When an expert is available, he or she can deliver the rules, but the knowledge acquisition process is a hard one, and sometimes does not lead to satisfactory results.

Such is the reason for the proposal of Machine Learning techniques as opposed to interview-based ones. The idea is to use these processes to capture or learn the knowledge from a set of examples describing the behaviour of the system. Later, when the model is sufficiently trained, this knowledge is translated into fuzzy rules. This approach has given as a result a collection of procedures to extract fuzzy rules from data, based on highly diverse algorithms or automatic learning models: clustering techniques, classification trees, evolutive algorithms, logic and neural networks.

2.3.3. Inference Process

In an FRBM evaluation and inference process, the definition of the logic operators plays a fundamental role. As indicated, there are different versions for each of them: conjunction, disjunction, negation and implication. Particular choices of the operators result in different “ways of reasoning”, so, in each case, we can adapt the inference process to the final use of the FRBM. There are no

restrictions in the combination of operators, but only the fact that the t-norm and the t-conorm are dual. There exist many empirical studies of the effectivity of diverse operator combinations, in particular those which try to establish the effect of the use of distinct t-norms or implication functions. The results of these studies are not conclusive. Formal argumentation on this choice is not a very common area of study, and the most common choices are the aforementioned Mamdani and TSK models.

2.3.4. Fuzzification and Defuzzification

When the system under study has real input and/or outputs, the FRBM must possess interfaces to perform the corresponding transformations between real and fuzzy data. As stated above, these interfaces are called fuzzifier and defuzzifier, respectively.

Concerning the fuzzifier, the most common option is to convert the real numbers into fuzzy singletons. Given a value $x_0 \in \mathbb{R}$, its associated fuzzy set has the following membership function:

$$\mu_A(x) = \begin{cases} 1, & x = x_0 \\ 0, & x \neq x_0 \end{cases}.$$

As well, sometimes triangular fuzzy numbers centred on the real value and with a small width δ are used:

$$\mu_A(x) = \Lambda(x_0 - \delta, x_0, x_0 + \delta).$$

The defuzzifier plays the opposite role: to transform a fuzzy number into a single real value. The most commonly used criteria are [52, 53]:

1. Maximum height. It returns the point where the fuzzy set reaches its maximum membership value.

2. Maximum average. The output is the arithmetic mean of all the points of the fuzzy set which reach the highest membership degree.

3. Centre of gravity. The returned value is the weighted mean of all the points of the support of the set, where the weight assigned to each point is its membership degree to the set.

It is important to remark that it is usual to defuzzify after the whole inference process. But there are other models in which defuzzification takes place on each consequent before the aggregation of all of them. The aggregation is performed in this case on already defuzzified values. This option is more efficient but has a drawback in quality.

2.4. Hybrid Neuro-Fuzzy Models

As an example of some of the developments that took place in the field of the design of FRBM, this section covers one of the most successful paradigms in practice.

The resurgence of interest in the field of artificial neural networks brought a new approach to fuzzy literature. The backpropagation learning algorithm, which until its application to neural networks was not very popular, is actually a universal learning paradigm for any "smooth" parametrised model, including fuzzy inference systems. As a result, fuzzy systems not only can take linguistic information (linguistic rules) from human experts, but they can adapt themselves as well, using numeric data (input/output data pairs) to achieve a better performance.

Neuro-fuzzy systems arose from the lack of standard methods to transform human knowledge or experience into an information base in a fuzzy inference system. To reach that objective, effective methods to adjust membership functions, as well as to minimise the output error measure, or maximise its performance index, were needed.

2.4.1. Adaptive Neuro-Fuzzy Inference System (ANFIS)

With the aforementioned ideas in mind, ANFIS (Adaptive Neuro-Fuzzy Inference System) architecture was developed by R. Jang [39]. It provides a method to build up a set of TSK fuzzy rules, with appropriated membership functions to generate the optimum input/output data pairs. This is achieved using a hybrid learning algorithm, based in the common adaptive network optimisation methods (steepest descent and least squares estimator).

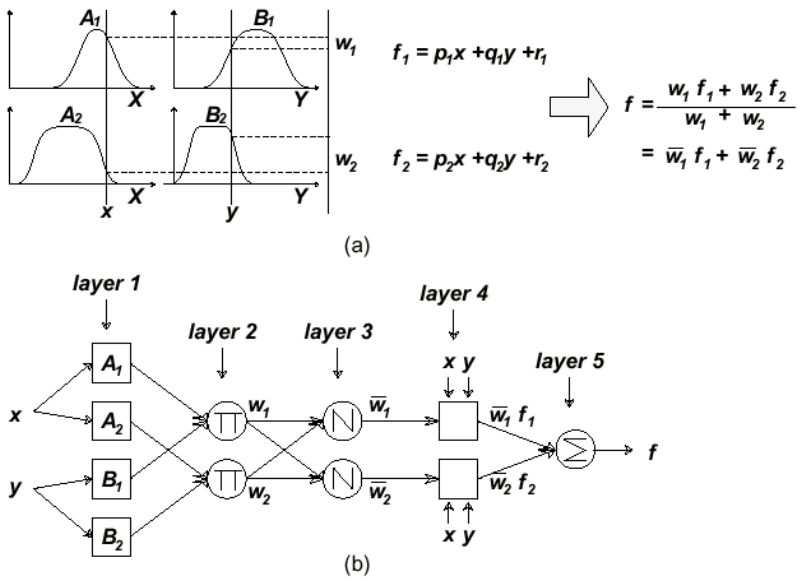

Figure 2.2.: (a) A two-input TSK fuzzy model with two rules; (b) equivalent ANFIS architecture.

ANFIS Architecture

For simplicity, we assume the fuzzy inference model under consideration has two inputs x and y and one output z. Suppose that the rule base contains two fuzzy TSK rules:

$$\text{Rule} \quad 1: \quad \text{If } x \text{ is } A_1 \text{ and } y \text{ is } B_1 \text{ then } f_1 = p_1 x + q_1 y + r_1$$
$$\text{Rule} \quad 2: \quad \text{If } x \text{ is } A_2 \text{ and } y \text{ is } B_2 \text{ then } f_2 = p_2 x + q_2 y + r_2$$

Figure 2.2(a) illustrates the reasoning mechanism for this model; the corresponding equivalent ANFIS architecture is shown in figure 2.2(b), where nodes of the same layer have similar functions, as described below (we will note the output of the ith node in layer l as $O_{l,i}$).

Layer 1 Every node i in this layer is a square node with node function given by

$$O_i^1 = \mu_{A_i}(x), \tag{2.16}$$

where x is the input to node i, and A_i is the linguistic label (*small*, *large* etc.) associated with this node function. In other words, O_i^l is the membership function of A_i and it specifies the degree to which the given x satisfies the quantifier A_i. In this model, $\mu_{A_i}(x)$ is chosen to be bell-shaped with maximum equal to 1 and minimum equal to 0, such as the generalised bell function, Equation (2.14), or the Gaussian function, Equation (2.13), page 23. In fact, any continuous and piecewise differentiable functions, such as commonly used trapezoidal or triangular-shaped membership functions, are also qualified candidates for node functions in this layer. Parameters in this layer are referred to as *premise parameters*.

Layer 2 Every node in this layer is a circle node labelled Π which multiplies (t-norm) the incoming signals and sends the product out:

$$\omega_i = \mu_{A_i}(x) \times \mu_{B_i}(y), \; i = 1,2. \tag{2.17}$$

Each node output represents the firing strength of a rule. (In fact, other t-norm operators that perform generalised AND can be used as the node function in this layer).

Layer 3 Every node in this layer is a circle node labelled N. The ith node calculates the ratio of the ith rule's firing strength to the sum of all rules' firing strengths:

$$\overline{\omega}_i = \frac{\omega_i}{\omega_1 + \omega_2}, \; i = 1,2. \tag{2.18}$$

For convenience, outputs of this layer will be called *normalised firing strengths*.

Layer 4 Every node i in this layer is a square node with a node function

$$O_i^4 = \overline{\omega}_i f_i = \overline{\omega}_i (p_i x + q_i y + r_i), \tag{2.19}$$

where $\overline{\omega}_i$ is the output of layer 3, and $\{p_i, q_i, r_i\}$ is the parameter set of the consequent of rule i. Parameters in this layer will be referred to as *consequent parameters*.

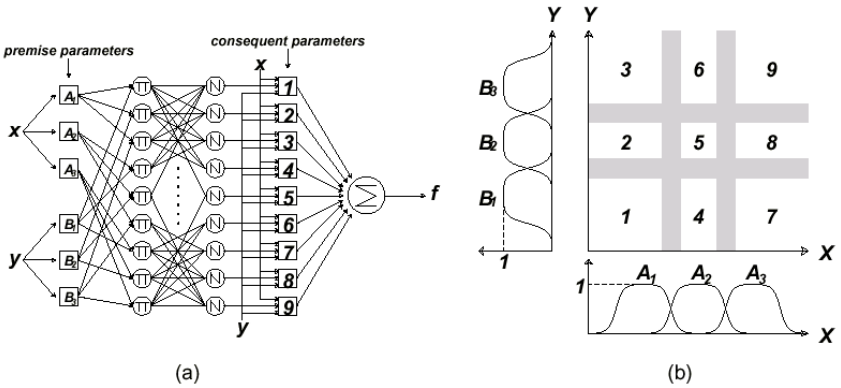

Figure 2.3.: (a) 2-input ANFIS with 9 rules; (b) corresponding fuzzy subspaces.

Layer 5 The single node in this layer is a circle node labelled Σ that computes the overall output as the summation of all incoming signals, i.e.,

$$O_1^5 = \sum_i \overline{\omega}_i f_i = \frac{\sum_i \omega_i f_i}{\sum_i \omega_i} \tag{2.20}$$

Figure 2.3 shows a 2-input ANFIS with 9 rules. Three membership functions are associated with each input, so the input space is partitioned into 9 fuzzy subspaces, each of which is governed by a fuzzy if-then rules. The premise part of a rule defines a fuzzy subspace, while the consequent part specifies the output within this fuzzy subspace.

Hybrid Learning Algorithm

Because ANFIS uses only differentiable functions, it is easy to apply standard learning procedures from neural network theory. For ANFIS a mixture of back-propagation (gradient descent) and least squares estimator (LSE) is used. Back-propagation is used to learn the antecedent parameters, i.e. the membership functions, and LSE is used to determine the coefficients of the linear combina-

tions in the rules' consequents[2]. A step in the learning procedure has two parts. In the first part the input patterns are propagated, and the optimal consequent parameters are estimated by an iterative least mean squares procedure, while the antecedent parameters are assumed to be fixed for the current cycle through the training set. In the second part, the patterns are propagated again, and in this epoch backpropagation is used to modify the antecedent parameters, while the consequent parameters remain fixed. This procedure is then iterated.

Let us consider an ANFIS model with n input units, k rule units and a single output unit that is to be trained with a learning problem of P patterns. The computation of the output of an ANFIS model is:

$$y = \frac{\sum_{i=1}^{k} \omega_i y_i}{\sum_{i=1}^{k} \omega_i} = \sum_{i=1}^{k} \overline{\omega}_i y_i = \sum_{i=1}^{k} \overline{\omega}_i \left(\alpha_0^{(i)} + \alpha_1^{(i)} x_1 + ... + \alpha_n^{(i)} x_n \right), \tag{2.21}$$

where

$$\omega_r = \prod_{i=1}^{n} \mu_{j_r}^{(i)}(x_i) \tag{2.22}$$

is the degree of fulfilment of rule R_r and

$$\overline{\omega}_i = \frac{\omega_r}{\sum_{i=1}^{k} \omega_i} \tag{2.23}$$

is the normalised degree of fulfilment. This expression for y is linear in the consequent parameters α_i^r and therefore these parameters can be estimated by LSE.

Let $\mathbf{N}$ be a matrix that contains one row for each pattern of the training set. Each row contains k repetitions of $\left(1, \overline{\omega}_1^{(i)}, ..., \overline{\omega}_k^{(i)}\right)$, where $\overline{\omega}_j^{(i)}$ is the normalised degree of fulfilment of rule j after the ith pattern has been propagated. In addition let $\mathbf{T}$ be the (column) vector of the target output values from the training set and let

$$\mathbf{A} = \left(\alpha_0^{(1)}, ..., \alpha_n^{(1)}, ..., \alpha_0^{(k)}, ..., \alpha_n^{(k)} \right)^T \tag{2.24}$$

be the (column) vector of all the consequent parameters of all the rules.

The consequent parameters are determined by the following matrix equation:

$$\mathbf{NA} = \mathbf{T}. \tag{2.25}$$

[2]As we will see in Chapter 5, this is called Concentrated Maximum Likelihood in the statistic literature.

With k rule units we have $M = k \cdot (n+1)$ consequent parameters; M is the size of $\mathbf{A}$. The dimension of $\mathbf{N}$ is $P \times M$ and the size of $\mathbf{T}$ is P. Because we usually have more training patterns than parameters, i.e. P is greater than M, the problem is over-determined, and generally there is no exact solution. To overcome this problem a least squares estimate $\mathbf{A}^*$ of $\mathbf{A}$ is determined that minimises the squared error $\|\mathbf{NA} - \mathbf{T}\|^2$. This can be done by writing

$$\mathbf{A}^* = (\mathbf{N}^T\mathbf{N})^{-1}\mathbf{NT}, \tag{2.26}$$

where $\mathbf{N}^T$ is the transpose of $\mathbf{N}$ and $(\mathbf{N}^T\mathbf{N})^{-1}\mathbf{N}$ is the pseudo-inverse of $\mathbf{N}$ if $\mathbf{N}^T\mathbf{N}$ is non-singular. However, to compute a solution is computationally extremely expensive due to matrix inversion, and it becomes impossible if $\mathbf{N}^T\mathbf{N}$ is singular. There is an iterative LSE procedure which is more efficient and is used to train ANFIS.

Let $\mathbf{n}_i^T$ be the ith row vector of matrix $\mathbf{N}$ and let $\mathbf{t}_i^T$ be the ith element of vector $\mathbf{T}$. Then a solution for $\mathbf{A}$ can be computed iteratively by evaluating

$$\mathbf{A}_{i+1} = \mathbf{A} + \Sigma_{i+1} \cdot \mathbf{n}_{i+1} \cdot (\mathbf{t}_{i+1}^T - \mathbf{n}_{i+1}^T \cdot \mathbf{A}_i) \tag{2.27}$$

$$\Sigma_{i+1} = \Sigma_i \cdot \frac{S_i \cdot \mathbf{n}_{i+1} \cdot \mathbf{n}_{i+1}^T \cdot \Sigma_i}{1 + \mathbf{n}_{i+1}^T \cdot \Sigma_i \cdot \mathbf{n}_{i+1}}, \qquad (i = 0, 1, ..., P-1) \tag{2.28}$$

where Σ is called the covariance matrix and the least squares estimate $\mathbf{A}^*$ is equal to $\mathbf{A}_P$. The initial conditions for the procedure are $\mathbf{A}_0 = 0$ and $\Sigma_0 = \gamma \mathbf{I}_M$ where γ is a large positive number and $\mathbf{I}_M$ is the identity matrix of dimension $M \times M$. If the ANFIS network has more than one output, for example l, then $\mathbf{T}$ is a $P \times l$ matrix, and $\mathbf{t}_i^T$ is its ith row vector. In this case $\mathbf{A}$ becomes an $M \times l$ matrix.

The modifications for the antecedent parameters are determined by backpropagation. Let p be a parameter of the fuzzy set $\mu_{j_r}^{(i)}$ from the antecedent of some rule R_r. We consider the change in p for a single rule R_r after a pattern has been propagated, where y^* is the target output value. The error measure E is the usual sum of squared differences between target and actual output. By

iterative application of the chain rule we obtain

$$\begin{aligned}
\Delta p &= -\sigma \frac{\partial E}{\partial p} \\
&= -\sigma \frac{\partial E}{\partial y} \frac{\partial y}{\partial \overline{\omega_r}} \frac{\partial \overline{\omega_r}}{\partial \omega_r} \frac{\partial \omega_r}{\partial \mu_{j_r}} \frac{\partial \mu_{j_r}^{(i)}}{\partial p} \\
&= \sigma \cdot (y^* - y) \cdot y_r \cdot \frac{\overline{\omega_r} \cdot (1 - \overline{\omega_r})}{\omega_r} \cdot \frac{\omega_r}{\mu_{j_r}} \frac{\partial \mu_{j_r}}{\partial p} \\
&= \frac{\sigma}{\mu_{j_r}} \cdot y_r \cdot (y^* - y) \cdot \overline{\omega_r} \cdot (1 - \overline{\omega_r}) \cdot \frac{\partial \mu_{j_r}^{(i)}}{\partial p},
\end{aligned} \tag{2.29}$$

where σ is a learning rate. For the last factor of the equation we obtain the following for the three parameters of a fuzzy set $\mu_{j_r}^{(i)}$:

$$\begin{aligned}
\frac{\partial \mu_{j_r}^{(i)}}{\partial a} &= \frac{2b}{a} \cdot \mu_{j_r}^{(i)}(x_i)^2 \cdot \left(\frac{x_i - c}{a}\right)^{2b} \\
\frac{\partial \mu_{j_r}^{(i)}}{\partial b} &= -2 \cdot \mu_{j_r}^{(i)}(x_i)^2 \cdot \log\left(\frac{x_i - c}{a}\right) \cdot \left(\frac{x_i - c}{a}\right)^{2b} \\
\frac{\partial \mu_{j_r}^{(i)}}{\partial b} &= \frac{2 \cdot b \cdot \mu_{j_r}^{(i)}(x_i)^2}{x_i - c} \cdot \left(\frac{x_i - c}{a}\right)^{2b}
\end{aligned} \tag{2.30}$$

The learning procedure suggested by Jang [39] consists of the following steps:

(i) Propagate all patterns from the training set and determine the consequent parameters by iterative LSE (eq. 2.27). The antecedent parameters remain fixed.

(ii) Propagate all patterns again and update the antecedent parameters by backpropagation (eqs. 2.29 and 2.30). The consequent parameters remain fixed.

(iii) If the error was reduced in four consecutive steps then increase the learning rate by 10%.

(iv) Stop if the error is small enough, otherwise continue with step (i).

2.4.2. Hybrid Neuro-Fuzzy Inference System (HyFIS)

The Mamdani fuzzy inference model [61] was proposed as the first attempt to control a steam engine and boiler combination by a set of linguistic control rules obtained from experienced human operators. Fig. 2.4, in page 33, shows the diagram of this type of fuzzy reasoning. Kim and Kasabov [46] developed a neuro-fuzzy version of this model called HyFIS.

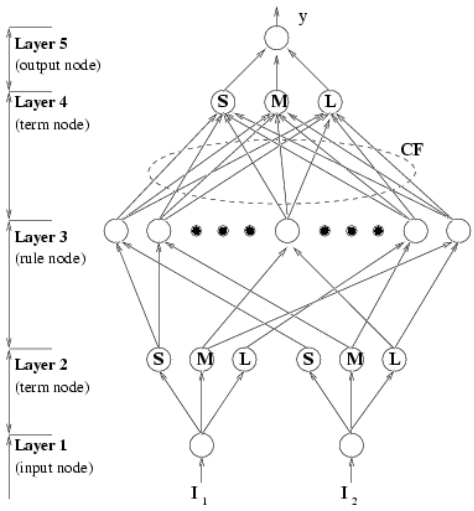

Figure 2.4.: The structure of HyFIS.

The architecture of HyFIS

HyFIS is a multilayer neural network-based fuzzy model. Its topology is shown in Fig. 2.4 and the system has a total of five layers. There are some similarities between this model and the previously seen ANFIS, being the differences caused by the fact that ANFIS is a TSK fuzzy inference model, as stated in page 29, while HyFIS is a Mamdani inference model.

In HyFIS, the input and output nodes represent the input states and output control/decision signals, respectively. In the hidden layers, we find nodes representing membership functions associated with the system variables and nodes representing rules. This provides an easy interpretation of the functioning of the network, and this is one of the main advantages of this model.

Nodes in layer 1 are input nodes that directly transmit input signals to the next layer. Nodes in layers 2 and 4 are *term nodes* that act as membership functions to express the system's fuzzy linguistic variables. For example, the fuzzy sets defined for the input–output variables in Fig. 2.4 are large (L), medium (M)

and small (S). The number of labels is arbitrary, so if more granularity is required, any —odd— number of labels can be used. Each node in layer 3 is a *rule node* and represents one fuzzy rule. The connection weights between different layers are usually set to 1, except for the connection between layers 3 and 4, in which the weights represent confidence factors (CFs) for each rule.

Kim and Kasabov used Gaussian membership functions, with parameters are the mean value c and the variance σ, as in Equation (2.13), page 23.

The meaning and functions of the nodes in HyFIS are described below. Following the notation in [46], we use indexes i, j, k and l for nodes in layers 2, 3, 4, and 5, respectively. The output from the nth node of layer (m) is denoted by $y_n^{(m)}$.

Layer 1 (*Input Layer*) Nodes in this layer are input nodes that represent input linguistic variables as crisp values. The nodes in this layer only transmit input values to the next layer, the membership function layer. Each of the nodes in layer 1 is connected to only those nodes of layer 2 which represent the linguistic values of the linguistic variable represented by it.

Layer 2 (*Antecedent Layer*) Nodes in this layer represent the membership functions corresponding to the labels of each linguistic variable. As seen before those membership functions are implemented using Gaussian functions. If no expert knowledge is available, initially the membership functions are spaced equally over the domain space. The output function of the nodes in this layer is, hence,

$$y_i^{(2)} = \mu_G(x). \tag{2.31}$$

Parameters in this layer are referred to as *antecedent parameters*.

Layer 3 (*Rule Layer*) Each node in layer 3 represents an antecedent of a fuzzy rule, i.e. a t-norm combination of input labels. Thus, all the nodes in this layer form a fuzzy rule base, and the output function is expressed as:

$$y_j^{(3)} = \min_{i \in I_j} y_i^{(2)} \tag{2.32}$$

where I_j is the set of indexes of the nodes in layer 2 that are connected to node j in layer 3, and $y_i^{(2)}$ is defined in eq. 2.31.

Layer 4 (*Consequent Layer*) A node in layer 4 represents a possible consequent of a fuzzy rule, and performs the t-conorm operation to integrate the field rules leading to the same output linguistic variables. The connection pattern between layer 3 and 4 depends upon the rule base obtained in the structure learning algorithm, as we shall see below. Each node in this layer represents a fuzzy label from the fuzzy quantisation space of an output variable. The activation of the node represents the degree to which this membership function is supported by some of the fuzzy rules together. The connection weights w_{kj} of the links connecting nodes k in layer 4 to nodes j in layer 3 represent conceptually the certainty factors of the corresponding fuzzy rules when inferring fuzzy output values. The initial connection weights of the links between layers 3 and 4 are randomly selected in the interval $[-1,1]$. The output of the nodes in this layer are expressed as:

$$y_k^{(4)} = \max_{j \in I_k} \left(y_j^{(3)} \omega_{kj}^2 \right) \tag{2.33}$$

where I_k is the set of indexes of the nodes in layer 3 that are connected to the node k in layer 4. Actually these connecting links function as a connectionist inference engine, which avoids the rule-matching process. Each of the rules is activated to a certain degree represented by the squared weight values.

Layer 5 (*Output Layer*) This represents the output variables of the model. These nodes and the links attached to them act as a defuzzifier. A node in this layer computes a crisp output signal. An approximation to the centre of gravity method was used, and was computed as:

$$y_l^{(5)} = \frac{\sum_{k \in I_l} y_k^{(4)} \sigma_{lk} c_{lk}}{\sum_{k \in I_l} y_k^{(4)} \sigma_{lk}} \tag{2.34}$$

where I_l is the set of indexes of the nodes in layer 4 which are connected to the node l in layer 5 and c_{lk} and σ_{lk} are the parameters of the membership function of the output linguistic value represented by k in layer 4.

Hybrid Learning Algorithm for HyFIS

The original HyFIS model used a two-phase hybrid learning scheme. In phase one, also known as *rule determination phase* the set of rules to be used by the system is defined using fuzzy techniques. In phase two, a supervised learning scheme based on a gradient descent learning is used to optimally adjust the membership functions for desired outputs. To begin the learning procedure, a training data set and the desired granularity of the fuzzy partitions (i.e. the size of the term set of each input or output linguistic variable) must be defined.

Algorithm 1 Wang and Mendel rule extraction (Cf. [89])

for each variable **do** {*STEP 1:* Partition variable's domains into fuzzy regions}
  Find the definition domain.
  Add 0.5% to each interval limit.
  Divide the domain in r fuzzy regions.
  Obtain the parameters (c,σ) for each label.
end for
for all $d \in$ Training data set **do** {*STEP 2:* Generate intermediate rules from training data}
  for all value in d **do**
    Find the label which fits it the best.
    Compute the CF as the product of the membership degrees.
  end for
end for
for all intermediate rule **do** {*STEP 3:* Refine the intermediate rule set}
  if the combination of labels is new **then**
    Insert the rule in the final rule set.
  else if its CF is higher **then**
    Update the CF of the corresponding rule in the final rule set.
  else
    Reject this rule.
  end if
end for

Structure learning phase A simple algorithm proposed by Wang and Mendel

[89] was used to generate fuzzy rules from the numerical input/output training data. This algorithm is composed by three main steps:

1. Divide the input and output spaces in fuzzy regions, and initialise a membership function for each of these regions.
2. Generate fuzzy rules from the available training data pairs. There will be as many rules as data included in the training set. Then, the membership function with higher fitness for each variable in each datum are determined.
3. A certainty factor (CF) is assigned to each rule. This factor is normally computed as the product of the membership degrees of each variable in a rule. Now, if two or more rules cause a conflict (because they have the same antecedent but different consequent) the rule with higher CF is inserted in the rule base and the rest are ignored.

The detailed pseudo-code corresponding to the Wang and Mendel algorithm is shown in Algorithm 1.

Parameter learning phase After the fuzzy rules are found, the whole network structure is established, and the network enters the second learning phase to optimally adjust the parameters of the membership functions. As stated above, backpropagation is used to perform this adjustment.

Let y_l be the desired output of the network for an input vector $X = (x_1, x_2, \cdots, x_p)$. In [46] the learning algorithm for HyFIS is derived using a gradient descent scheme to minimise the error function

$$E = \frac{1}{2} \sum_{X} \sum_{l=1}^{q} (d_l - y_l^{(5)})^2 \tag{2.35}$$

where q is the number of nodes in layer 5.

As an example, let p be a parameter of a fuzzy label belonging to the consequent layer. We consider the change in p after a pattern has been propagated, and hence the general learning rule used by gradient descent learning is:

$$p(t+1) = p(t) + \Delta p = p(t) - \eta \Big(\frac{\partial E}{\partial p} \Big) \tag{2.36}$$

where $\eta > 0$ is the learning rate, and the chain rule is described as follows:

$$\frac{\partial E}{\partial p} = \frac{\partial E}{\partial y_k^{(4)}} \frac{\partial y_k^{(4)}}{\partial p} = \frac{\partial E}{\partial y_l^{(5)}} \frac{\partial y_l^{(5)}}{\partial y_k^{(4)}} \frac{\partial y_k^{(4)}}{\partial p}. \tag{2.37}$$

To calculate the learning rule for each parameter, we shall describe the computations of $\partial E/\partial p$ layer by layer, starting from the output nodes. This calculation is specified in [46], and the resulting update formulae are:

Consequent Layer's parameters:

$$\begin{aligned} c_k(t+1) &= c_k(t) + \eta \delta^{(5)} \frac{\sigma_k y_k^{(4)}}{\sum_{\forall m} \sigma_m y_m^{(4)}} \\ \sigma_k(t+1) &= \sigma_k(t) + \eta \delta^{(5)} \frac{y_k [c_k (\sum_{\forall m} y_m^{(4)} \sigma_m - \sum_{\forall m} y_m^{(4)} \sigma_m c_m)]}{(\sum_{\forall m} y_m^{(4)} \sigma_m)^2} \end{aligned} \tag{2.38}$$

where $\delta^{(5)}$ is the error signal that comes from the output layer.

Antecedent Layer's parameters:

$$\begin{aligned} c_i(t+1) &= c_i(t) + \eta \beta_i y_i^{(2)} \frac{2(x-c_i)}{\sigma_i^2} \sum_{j \in O_i^{(2)}} \delta_j^{(3)} \\ \sigma_i(t+1) &= \sigma_i(t) + \eta \beta_i y_i^{(2)} \frac{2(x-c_i)^2}{\sigma_i^3} \sum_{j \in O_i^{(2)}} \delta_j^{(3)} \end{aligned} \tag{2.39}$$

where $\sum_{j \in O_i^{(2)}} \delta_j^{(3)}$ represents the error signals coming from the previous layer —but only from those nodes to which the unit that we are updating is connected—, and

$$\beta_i = \begin{cases} 1 & \text{if } i = \underset{i \in I_j}{\arg\min}(y_i^{(2)}) \\ 0 & \text{in any other case} \end{cases} \tag{2.40}$$

being i the index of the unit that produced the minimum output in the forward pass, i.e. we only consider the error signals of those units that used the output of the unit that we are updating.

3. Statistical Models for Time Series Analysis

This chapter reviews the classical tools for Time Series Analysis together with the latest statistical developments. Section 3.1 describes the most commonly used approach to Time Series: the Box-Jenkins classical methodology. Section 3.2.2 discusses another widely used classical method: the Holt-Winters exponential smoothing procedure. Finally, section 3.3 gives an overview of the more recent Threshold autoregressive models, including the last developments in this area.

3.1. Box-Jenkins Methodology

The most popular class of linear time series models consists of autoregressive moving average (ARMA) models [7], including purely autoregressive models (AR) and purely moving-average (MA) models as special cases. ARMA(p, q) models are frequently used to model linear dynamic structures, to depict linear relationships among lagged variables, and to serve as vehicles for linear forecasting.

In the original work by Box and Jenkins, a methodology is suggested for the application of these models, consisting basically in three phases:

Model identification The first step in developing a Box-Jenkins model is to determine if the series is stationary and if there is any significant seasonality that needs to be modelled. Box and Jenkins recommend the differencing approach to achieve stationarity and to remove seasonality.

Once stationarity and seasonality have been addressed, the next step is to identify the order (i.e., the p and q) of the autoregressive and moving average terms. Several empirical rules exist, and it is also possible to

explore many combinations of ARMA processes using specific software (as GNU's R) and choose the one which minimises the cost function[1].

Model estimation The main approaches to fitting Box-Jenkins models are non-linear least squares and maximum likelihood estimation. The latter is generally the preferred technique. The likelihood equations for the full Box-Jenkins model are complicated and are not included here.

Model validation Model diagnostics for Box-Jenkins models centre its attention in the error term ε_t. It is assumed to follow the assumptions for a stationary univariate process. The residuals should be white noise (or independent when their distributions are normal) drawings from a fixed distribution with a constant mean and variance. If the Box-Jenkins model is a good model for the data, the residuals should satisfy these assumptions.

If these assumptions are not satisfied, we need to fit a more appropriate model. That is, we go back to the model identification step and try to develop a better model. Hopefully the analysis of the residuals can provide some clues as to a more appropriate model.

In order to apply these models, the series must satisfy two requirements: (a) the series is stationary or can be transformed in such through a simple transformation (like differentiating) and (b) the series follows a linear model. Both hypotheses are a mathematical idealisation which can be hard for some cases.

3.1.1. AR Model

An *autoregressive model* of order $p \geq 1$ is defined as

$$X_t = b_1 X_{t-1} + \ldots + b_p X_{t-p} + \varepsilon_t \tag{3.1}$$

where $\{\varepsilon_t\} \sim N(0, \sigma^2)$, usually known as *white noise*. For this model we write $\{X_t\} \sim \text{AR}(p)$, and the time series $\{X_t\}$ generated from this model is called the AR(p) process.

Model (3.1) represents the current state X_t through its immediate p past values $X_{t-1}, \ldots, X_{t-p}$ in a linear regression form. It explicitly specifies the relationship between the current value and its past values. The model is easy

[1]This possibility of doing an automatic search through the order's permutations was prohibitive in computer time in the past, but nowadays is not an extremely costly process.

to implement and therefore is arguably the most popular time series model in practice.

3.1.2. MA Model

A *moving average process* with order $q \geq 1$ is defined as

$$X_t = \varepsilon_t + a_1\varepsilon_{t-1} + \ldots + a_q\varepsilon_{t-q}, \tag{3.2}$$

where $\{\varepsilon_t\} \sim N(0,\sigma^2)$, a white noise. We note this model as MA(q), and it expresses a time series as a moving average of a white noise process. The correlation between X_t and X_{t-h} is due to the fact that they may depend on the same ε_{t-j}'s.

3.1.3. ARMA Model

The AR and MA classes can be further enlarged to model more complicated dynamics of time series. Combining AR and MA forms together yields the popular autoregressive moving average (ARMA) model defined as

$$X_t = b_1X_{t-1} + \ldots + b_pX_{t-p} + \varepsilon_t + a_1\varepsilon_{t-1} + \ldots + a_q\varepsilon_{t-q}, \tag{3.3}$$

where $\{\varepsilon_t\} \sim N(0,\sigma^2)$, $p,q \geq 0$ are integers and (p,q) is called the order of the model.

ARMA models are one of the most frequently used families of parametric models in time series analysis. This is due to their flexibility in approximating many stationary processes.

3.2. Smoothing and decomposition methods

3.2.1. Seasonal-trend decomposition based on loess smoothing

STL [17] is a filtering procedure for decomposing a time series into trend, seasonal, and remainder components. STL has a simple design that consists of a sequence of applications of the loess smoother; the simplicity allows analysis of the properties of the procedure and allows fast computation, even for very long

time series and large amounts of trend and seasonal smoothing. Other features of STL are specification of amounts of seasonal and trend smoothing that range, in a nearly continuous way, from a very small amount of smoothing to a very large amount; robust estimates of the trend and seasonal components that are not distorted by aberrant behaviour in the data; specification of the period of the seasonal component to any integer multiple of the time sampling interval greater than one; and the ability to decompose time series with missing values.

3.2.2. Holt-Winters smoothing

The development of time series models begun with a modelling strategy called Time Series Decomposition [94]. This approach is based in describing the behaviour of the time series through its non-observable components: trend, seasonality, cycle and random perturbation. It is not possible to observe the trend of a time series, but it can be argued that it is linear, nonlinear or exponential. In the same way, it may be clear that in some specific periods of time, the values of the time series are higher or lower with a certain degree of regularity.

Exponential Smoothing techniques provide a way for predicting future time series values by weighting the influence of past observations. They are also sometimes called self-adaptive methods because once the parameters are estimated, the forecasts can be updated at each new observation. Holt-Winters method tries to express the time series as an additive or multiplicative combination of its components:

$$X_t = \mu_t + T_t k + S_t + \varepsilon_{t-q} \quad \text{or} \quad X_t = (\mu_t + T_t k) S_t + \varepsilon_{t-q} \tag{3.4}$$

where μ_t is the exponentially weighted average of the past values of the series,

$$\mu_t = \alpha_1 \frac{X_t}{S_{t-l}} + (1-\alpha_1)(\mu_{t-1} + T_{t-1}), \tag{3.5}$$

T_t is the trend component, computed as

$$T_t = \alpha_2(\mu_t - \mu_{t-1}) + (1-\alpha_2)T_{t-1}, \tag{3.6}$$

and S_t is the seasonal component, with cycle l, computed as

$$S_t = \alpha_3 \frac{X_t}{\mu_t} + (1-\alpha_3)S_{t-l}. \tag{3.7}$$

By analysing equation (3.5), we can observe how it calculates the value of the constant component in time t taking in first place the information contained in the value of X_t corrected with the seasonality, and then adds up the information given by the values of the immediately previous instant as the sum of the trend estimation and the constant term previously calculated.

Equation (3.6) estimates the value of the trend in time t by combining the difference of the estimation of the average in t and in $t-1$ with the value of the trend in the previous instant.

Finally, equation (3.7) estimates the seasonal factor taking into account on the one hand an estimation of the seasonal effect in instant t calculated as the ratio between the value of the original series in time t and an estimation of the average in t. On the other hand, it sums to this ratio the contribution of the seasonal factor previously calculated.

Parameters α_1, α_2 and α_3 define the contribution of each component to the overall estimation, and must be fixed by the practitioner. Automatic selection of those parameters is implemented in some software packages as GNU's R.

Because this is a recursive method, initial values must be fixed before using it. Some generic rules have been proposed and, as before, automatic selection is available in GNU's R software amongst others.

3.3. Nonlinear models

Given the limitations of the basic models seen above, in the last years much research has been devoted to nonlinear models. Different studies show that nonlinear and non-stationary models are more flexible in capturing the characteristics of data and that, in some cases, are better in terms of estimation and forecasting. These advances do not rule out linear models at all, because these models are a first approach which can be of great help to further estimate some of the parameters.

Modelling of any real-world problem by using nonlinear models must start by evaluating if the behaviour of the series follows a linear or nonlinear pattern, and in the latter case, analyse the type of nonlinearity ruling it. There are several tests which allow for testing linearity against some fixed nonlinear alternatives, but they go beyond the scope of this document. Tong [86] proposed a classification of these nonlinear alternatives:

- Threshold models
- Amplitude-dependent exponential autoregressive models (EXPAR)
- Fractional autoregressive models (FAR)
- Product autoregressive models (PAR)
- Random coefficient autoregressive models (RCA)
- Newer exponential autoregressive models (NEAR)
- Discrete state-space autoregressive models
- Bilinear models (BL)
- Nonlinear moving average models
- Autoregressive models with conditional heteroscedasticity
- Second generation models
- Doubly stochastic models
- State dependent models (SDM)

In this document we will be focusing just on Threshold models, and we will cover the following types:

- Piecewise linear models or Threshold autoregressive models (TAR)
- Smooth transition (or threshold) autoregressive models (STAR), which include
  - Logistic smooth transition autoregressive models (LSTAR)
  - Exponential smooth transition autoregressive models (ESTAR)
  - Normal smooth transition autoregressive models (NSTAR)
- Autoregressive neural network (AR-NN)
- Linear local global neural network (L^2GNN)
- Neuro-coefficient smooth transition autoregressive model (NCSTAR)

The last three are closely related to the Artificial Neural Network paradigm, as we shall see.

Notation

For the sake of clarity, and in order to keep a unified view of Threshold models, we will use a homogeneous notation for all the models described here.

Henceforth we will note:

- y_t is the value at time t of a time series $\{y_t\}$.
- $\tilde{\mathbf{x}}_t \in \mathbb{R}^p$ is a $p \times 1$ vector of lagged values of y_t and/or some exogenous variables.
- $\mathbf{x}_t \in \mathbb{R}^{p+1}$ is defined as $\mathbf{x}_t = [1, \tilde{\mathbf{x}}_t']'$, referring to its first element as an *intercept*.
- The general nonlinear model is expressed as

$$y_t = \Psi(\mathbf{x}_t; \boldsymbol{\psi}) + \varepsilon_t, \tag{3.8}$$

  where

  - $\Psi(\mathbf{x}_t; \boldsymbol{\psi})$ is a nonlinear function of the variables $\mathbf{x}_t$ with parameter vector $\boldsymbol{\psi}$.
  - $\{\varepsilon_t\}$ is a sequence of independently normally distributed random variables with zero mean and variance σ^2.

- The logistic function used in the models when defined over the domain $\mathbb{R}^p$ is usually expressed as

$$f(\boldsymbol{\omega}\mathbf{x}_t) = (1 + \exp(-\boldsymbol{\omega}\mathbf{x}_t))^{-1}, \tag{3.9}$$

  where the norm of $\boldsymbol{\omega}$, called γ or *slope parameter* controls the speed of change between models, and β is comparable to the threshold that marks the regime switch. Hence the logistic function is hence sometimes rewritten as

$$f(\gamma(\boldsymbol{\varphi}\mathbf{x}_t - \beta)) = (1 + \exp\left(\gamma(\boldsymbol{\varphi}\mathbf{x}_t - \beta)\right))^{-1}, \tag{3.10}$$

  and in its one-dimensional flavour

$$f(\gamma(y_{t-d} - c)) = (1 + \exp(\gamma(y_{t-d} - c)))^{-1}, \tag{3.11}$$

  where y_{t-d} is usually known as the *transition* or *threshold variable*, and d is called *delay parameter*.

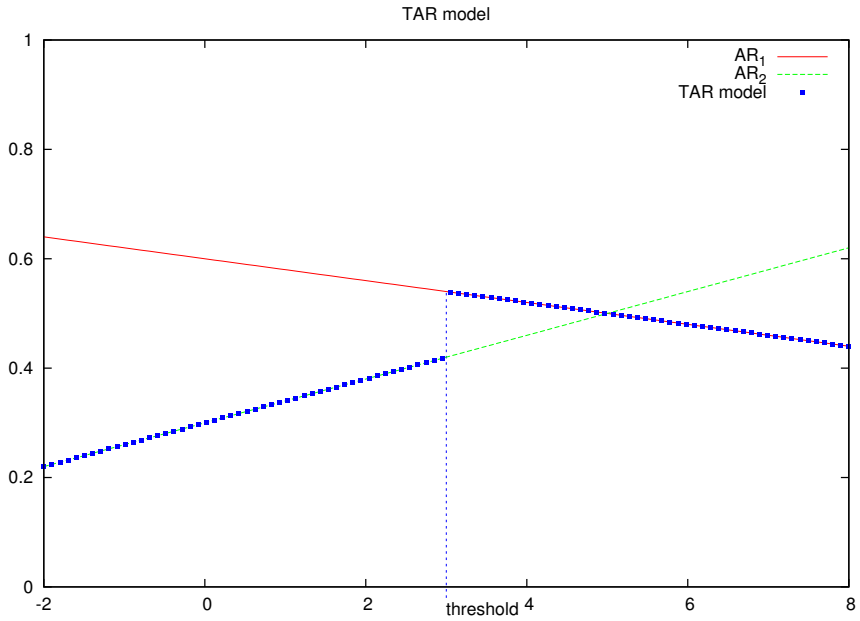

Figure 3.1.: An example of TAR model

3.3.1. Threshold autoregressive model (TAR)

In an attempt to solve the limitations of the linear approach, Tong [85] presented a piecewise linear approximation which consisted in partitioning a state-space into several subspaces.

A *threshold autoregressive* (TAR) model with k ($k \geq 2$) regimes is defined as

$$y_t = \sum_{i=1}^{k} \boldsymbol{\omega}_i \mathbf{x}_t I(s_t \in A_i) + \varepsilon_t = \sum_{i=1}^{k} \{\omega_{i,0} + \omega_{i,1} y_{t-1} + \omega_{i,p_i} y_{t-p_i} + \varepsilon_t\} I(s_t \in A_i) + \varepsilon_t, \quad (3.12)$$

where s_t is the threshold variable, I is an indicator (or *step*) function, $p_1, \ldots, p_k$

are some unknown positive integers, $\boldsymbol{\omega}_i$ are unknown parameters, and $\{A_i\}$ forms a partition of $(-\infty,\infty)$ with $\cup_{i=1}^{k} A_i = (-\infty,\infty)$ and $A_i \cap A_j = \emptyset, \forall i \neq j$.

When the threshold variable is one of the lagged values of y_t, i.e. $s_t = y_{t-d}$, the model is known as *self-exciting* —yielding the acronym SETAR.

In this model, we fit on each subset A_i a linear autoregressive form. The partition is dictated by the threshold variable y_{t-d}. It is often the case that $A_i = (r_{i-1}, r_i]$, with $-\infty = r_0 < r_1 < \ldots < r_k = \infty$, where the r_i's are called thresholds.

3.3.2. Smooth transition autoregressive model (STAR)

A key feature of TAR models is the discontinuous nature of the AR relationship as the threshold is passed. If one believes that nature is generally continuous, one might choose an alternative model called *smooth threshold autoregressive* or *smooth transition autoregressive* (STAR) proposed by Teräsvirta [83]. In STAR models there is a smooth continuous transition from one linear AR to another, rather than a sudden jump.

In STAR models and variants (cf. [87]), we change the indicator function $I(\cdot)$ in (3.12) from a *step* function that takes the value zero below the threshold and one above it, to a smooth function with sigmoid characteristics, as is (3.11). Hence, the STAR model with k regimes ($k > 2$) is defined as

$$y_t = \sum_{i=1}^{k} \boldsymbol{\omega}_i \mathbf{x}_t \mathrm{F}_i(s_t;\gamma_i,c_i) + \varepsilon_t, \tag{3.13}$$

The transition function, $\mathrm{F}(s_t;\gamma,c)$, is a continuous function that is bounded between 0 to 1. The most popular choices for $\mathrm{F}(s_t;\gamma,c)$ are discussed below. In the original STAR model, the transition variable s_t is assumed to be a lagged endogenous variable, that is $s_t = y_{t-d}$ for certain integer $d > 0$. In that case the model is usually called self exciting TAR (SETAR). We do not make this assumption here. Thus, the transition variable can also be an exogenous variable ($s_t = z_t$), or a (possibly nonlinear) function of lagged endogenous variables: $s_t = h(\tilde{\mathbf{x}}_t;\boldsymbol{\alpha})$ for some function h which depends on the $(p \times 1)$ parameter vector $\boldsymbol{\alpha}$. Finally, the transition variable can be a linear time trend ($s_t = t$) which gives rise to a model with smoothly changing parameters.

The regime that occurs at time t is determined by the observable variable s_t and the associated value of $\mathrm{F}(s_t;\gamma,c)$. Different choices for the transition func-

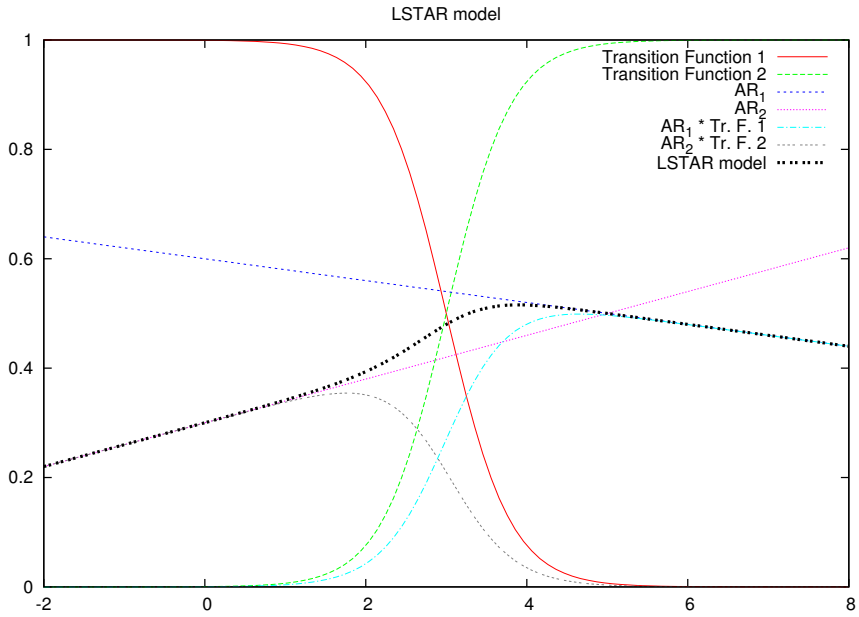

Figure 3.2.: An example of 2 regime STAR model using logistic transition function.

tion give rise to different types of regime-switching behaviour. A popular choice for $\mathrm{F}(s_t;\gamma,c)$ is the first-order logistic function, equation (3.11), and the resultant model is called the logistic STAR (LSTAR).

In the LSTAR model, we define the transition function $\mathrm{F}(s_t;\gamma,c)$ of expression (3.13) as

$$\mathrm{F}_i(s_t;\gamma_i,c_i) = \begin{cases} 1-f(s_t;\gamma_i,c_i) & \text{if } i=1 \\ f(s_t;\gamma_i,c_i)-f(s_t;\gamma_{i+1},c_{i+1}) & \text{if } 1<i<k \\ f(s_t;\gamma_i,c_i) & \text{if } i=k \end{cases} \tag{3.14}$$

where $f(s_t;\gamma_i,c_i)$ is defined as in (3.11). The LSTAR model can be (and usually

is) consequently rewritten as

$$y_t = \boldsymbol{\omega}_1 \mathbf{x}_t + \sum_{i=2}^{k} \boldsymbol{\omega}_i \mathbf{x}_t \mathrm{f}(s_t; \gamma_i, c_i) + \varepsilon_t. \tag{3.15}$$

Each of the parameters c_i in (3.15) can be interpreted as the threshold between two regimes, in the sense that the logistic function changes monotonically from 0 to 1 as s_t increases and $\mathrm{F}(c_i; \gamma_i, c_i) = 0.5$. The parameter γ_i determines the smoothness of the transition from one regimen to another. As γ_i becomes very large, the logistic function approaches the indicator function $I(\cdot)$ and hence the change of $\mathrm{F}(s_t; \gamma_i, c_i)$ from 0 to 1 becomes instantaneous at $s_t = c$. Consequently, the LSTAR nests threshold autoregressive (TAR) models as a special case. When $\gamma \to 0$ the LSTAR model reduces to a linear AR model.

In the LSTAR model, the regime switches are associated with small and large values of the transition variable s_t relative to c. In certain applications it may be more appropriate to specify the transition function such that the regimes are associated with small and large absolute values of s_t (again relative to c). This can be achieved by using, for example, the exponential function, in which case the model may be named ESTAR. Other frequently used function is the normal distribution, which yields the acronym NSTAR.

3.3.3. Autoregressive neural network model (AR-NN)

After the success of Artificial Neural Networks in so many fields including Time Series Analysis, Medeiros *et al.* [84] considered them as statistical nonlinear models and applied statistical inference to the problem of the model's specification. They devised a "bottom-up" strategy which allowed for proper statistical inference, as well as an in-sample evaluation of the estimated model.

The autoregressive single hidden layer neural network model is defined as

$$y_t = \boldsymbol{\omega}_1 \mathbf{x}_t + \sum_{i=2}^{k} \nu_i f(\boldsymbol{\omega}_i \mathbf{x}_t) + \varepsilon_t \tag{3.16}$$

where ν_i are known as "connection strengths" as in the neural network literature. Furthermore, the function $f(\cdot)$ is called a "hidden unit" or "squashing function", and is assumed to be logistic in this section. Although in the Soft Computing field it is frequent to take $\boldsymbol{\omega}_1 = 0$, the AR-NN includes this "linear unit" because of the definition of the transition function, as in equation (3.14).

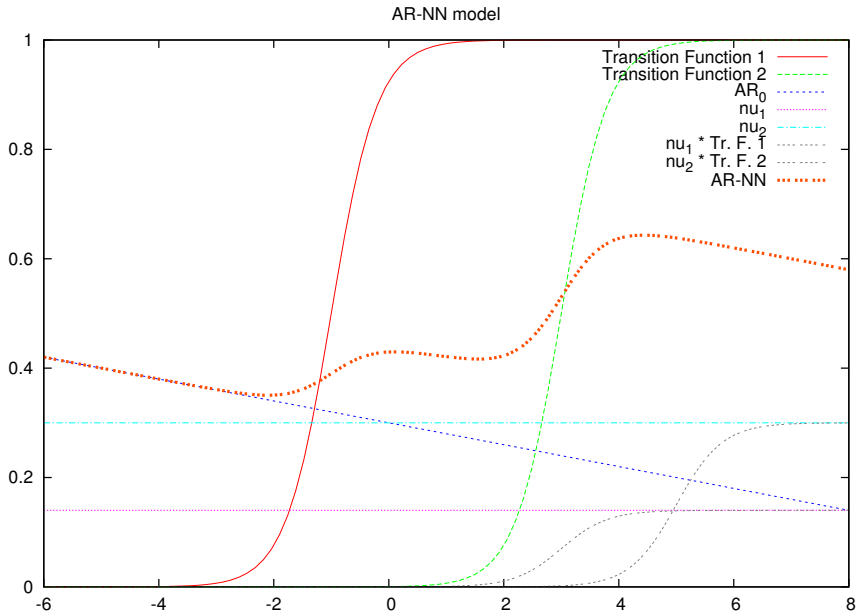

Figure 3.3.: An example of an AR-NN with 2 hidden units.

The geometric interpretation of this model considers that the AR-NN divides the p-dimensional Euclidean space with hyperplanes (defined by $\boldsymbol{\omega}_i \mathbf{x}_t$) resulting in several polyhedral regions. It computes the output as the sum of the contribution of each hyper-region modulated by a smoothing function $f(\cdot)$.

Following [84], an AR-NN can be either interpreted as a semi-parametric approximation to any Borel-measurable function or as an extension of the LSTAR model where the transition variable can be a linear combination of stochastic variables. A statistical problem of this model, though, is that it is, in principle, neither globally nor locally identified[2].

Three characteristics of the model imply non-identifiability. The first one is

[2]This is a mostly irrelevant concept for Soft Computing researchers so far, but in our opinion it might be used in the future as we shall see in chapter 4.

the exchangeability property of the AR-NN model. The value in the likelihood function of the model remains unchanged if we permute the hidden units. This results in $h!$ different models that are indistinguishable from each other and in $h!$ equal local maxima of the log-likelihood function. The second characteristic is that, for the squashing function, $f(x) = 1 - f(-x)$. This yields two observationally equivalent parametrisations for each hidden unit. Finally the presence of irrelevant hidden units is a problem. If model (3.16) has hidden units such that $\nu_i = 0$ for at least one i, the parameters $\boldsymbol{\omega}_i$ remain unidentified. Conversely, if $\boldsymbol{\omega}_i = 0$ then ν_i can take any value without the likelihood function being affected.

The approach devised by [84] overcomes these limitations, being thus able to build models with good statistical properties.

3.3.4. Linear Local Global Neural Network (L^2GNN)

Another statistical approach to artificial neural networks is the Local Global Neural Network (LGNN) model. The central idea of LGNN is to express the input-output mapping by a piecewise structure. The network output is constituted by a combination of several pairs, each of those composed by an approximation function and by an activation-level function. The activation-level functions are equivalent to the squashing function or hidden units mentioned above, and define the role of an associated approximation function for each subset of the domain. Partial superposition of activation-level functions is allowed. In this way, the problem of approximation functions is approached by the specialisation of neurons in each of the sectors of the domain. In other words, the neurons are formed by pairs of activation-level and approximation functions that emulate the generator function in different parts of the domain.

The LGNN is thus defined as

$$y_t = \sum_{i=1}^{k} \mathrm{L}(\mathbf{z}_t; \boldsymbol{\psi}_{L_i}) \mathrm{B}(\mathbf{z}_t; \boldsymbol{\psi}_{B_i}) + \varepsilon_t \tag{3.17}$$

where $\mathbf{z}_t$ is a vector of lagged values of y_t and/or some exogenous variables and the functions $\mathrm{L}(\mathbf{z}_t; \boldsymbol{\psi}_{L_i}) \colon \mathbb{R}^p \to \mathbb{R}$ and $\mathrm{B}(\mathbf{z}_t; \boldsymbol{\psi}_{B_i}) \colon \mathbb{R}^p \to \mathbb{R}$ are the approximation and activation-level functions respectively.

In the original formulation, $\mathrm{B}(\mathbf{z}_t;\boldsymbol{\psi}_{B_i})$ is defined as

$$\mathrm{B}(\mathbf{z}_t;\boldsymbol{\psi}_{B_i}) = -\left[\frac{1}{1+\exp\left(\gamma(\boldsymbol{\varphi}\mathbf{z}_t-\beta^{(1)})\right)} - \frac{1}{1+\exp\left(\gamma(\boldsymbol{\varphi}\mathbf{z}_t-\beta^{(2)})\right)}\right] \tag{3.18}$$

where $\boldsymbol{\psi}_{B_i} = \left(\gamma, \boldsymbol{\varphi}, \beta^{(1)}, \beta^{(2)}\right)$.

This model is closely related with the *mixture-of-experts* approach [38] and offers a great flexibility in the functional form of the approximation function $\mathrm{L}(\mathbf{z}_t;\boldsymbol{\psi}_{L_i})$. This flexibility has not been fully explored so far, but there have been attempts to combine in the same model linear approximators with nonlinear ones [26], for example.

A special case of the LGNN model is the Linear-Local Global Neural Network (L^2GNN) [76]. In this case, the approximation functions are linear, that is, $\mathrm{L}(\mathbf{z}_t;\boldsymbol{\psi}_{L_i}) = \boldsymbol{\omega}_i\mathbf{z}_t$, with $\boldsymbol{\omega}_i = (\omega_{i1},\omega_{i2},\ldots,\omega_{ip})' \in \mathbb{R}^p$. Hence, the L^2GNN is defined as

$$y_t = \sum_{i=1}^{k} \boldsymbol{\omega}_i\mathbf{z}_t\mathrm{B}(\mathbf{z}_t;\boldsymbol{\psi}_{B_i}) + \varepsilon_t \tag{3.19}$$

and the stochastic process consists of a mixture of linear processes.

It is worth noting that, as the previous AR-NN model, this model is neither locally nor globally identified, although in this case, the model has a generic property, asymptotic stationarity, which is proved useful and is closely related to the type of activation-level functions that it uses.

3.3.5. Neuro-Coefficient Smooth Transition AutoRegression (NCSTAR)

One of the last developments in threshold-based models is the Neuro-Coefficient STAR [64]. This model is a generalisation of some of the previously described models and can handle multiple regimes and multiple transition variables. It can be seen as a linear model whose parameters change through time and are determined dynamically by a single hidden layer feedforward neural network.

Consider a linear model with time-varying coefficients expressed as

$$y_t = \boldsymbol{\phi}_t'\mathbf{x}_t + \varepsilon_t, \tag{3.20}$$

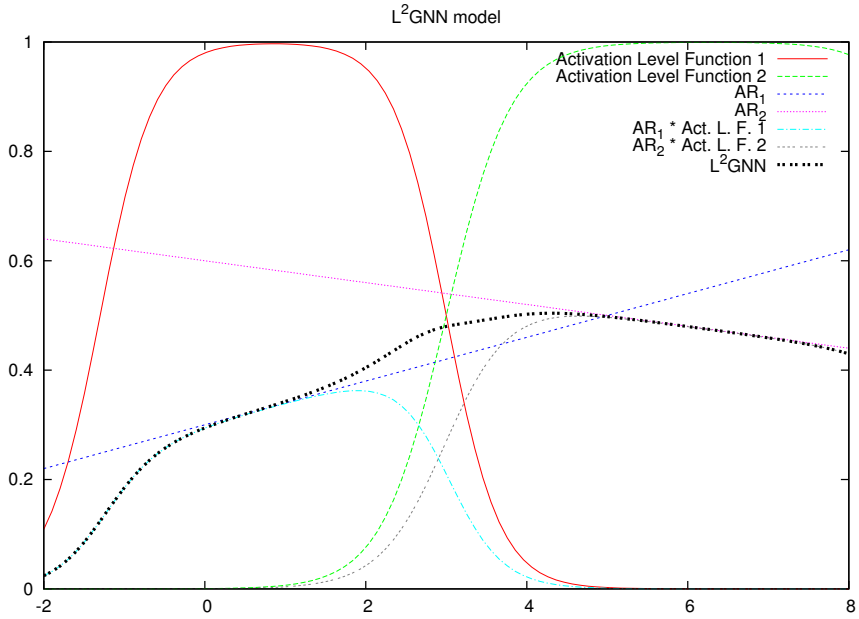

Figure 3.4.: An example of an L^2GNN with 2 hidden units.

where $\boldsymbol{\phi}_t = (\phi_t^{(0)}, \phi_t^{(1)}, \ldots, \phi_t^{(p)})' \in \mathbb{R}^{p+1}$ is a vector containing the coefficients of the model. The time evolution of the coefficients $\phi_t^{(j)}$ of (3.20) is given by the output of a single hidden layer neural network with k hidden units

$$\phi_t^{(j)} = \sum_{i=1}^{k} \nu_{ji} f(\boldsymbol{\omega}_i \mathbf{z}_t) - \nu_{j0}, \quad j = 0, \cdots, p, \tag{3.21}$$

where ν_{ji} and ν_{j0} are real coefficients.

Substituting the p realisations of (3.21) in (3.20) we obtain the general form of the NCSTAR model:

$$y_t = \boldsymbol{\nu}_1 \mathbf{x}_t + \sum_{i=2}^{k} \boldsymbol{\nu}_i \mathbf{x}_t f(\boldsymbol{\omega}_i \mathbf{z}_t) + \varepsilon_t, \tag{3.22}$$

where $\mathbf{z}_t$ is a $q \times 1$ vector of transition variables and $\omega_i = [\omega_{1i}, \ldots, \omega_{qi}]'$ are real parameters. The norm of ω_i, called γ_i, is known as the *slope parameter*. In the limit, when the slope parameter approaches infinity, the logistic function becomes a step function. Again, f is defined as in (3.9), but if the alternative smoothing function f, as defined in (3.14), is used, the model can be rewritten as

$$y_t = \sum_{i=1}^{k} \boldsymbol{\nu}_i \mathbf{x}_t f_i(\boldsymbol{\omega}_i \mathbf{z}_t) + \varepsilon_t. \tag{3.23}$$

As happened with previous models, this model is neither locally nor globally identified, and again this is due to the three characteristics of neural networks that cause non-identifiability (see section 3.3.3). A discussion about these problems and their solutions can be found in [64].

The choice of the elements of $\mathbf{z}_t$, which determines the dynamics of the process allows a number of special cases. An important one is when $\mathbf{z}_t = y_{t-d}$. In this case, model (3.22) becomes a LSTAR model with k regimes, expressed as in (3.15). It should be noticed as well that this model also nests the SETAR model. When $\gamma_i \to \infty \ \forall i$, the LSTAR model becomes a SETAR model with k regimes.

Another interesting case is when $\boldsymbol{\nu}_i' = (\nu_{0i}, 0, \ldots, 0)$. Then the model becomes an AR-NN model with k hidden units, as seen in section 3.3.3. Finally, this model is related to the Functional Coefficient Autoregressive (FAR) model [16], to the Single-Index Coefficient Regression model [96], and to Fuzzy Additive Systems, as we shall see in chapter 4.

4. Relations amongst models

This chapter contains the main original contribution that we have developed. It explores the relationships existing between two families of time series models: statistical threshold based models seen in Chapter 3 and fuzzy rule based models covered by Chapter 2. Section 4.1 states the close ties between TSK fuzzy rules and AR models, and the rest of the chapter analyses the consequences of this result.

4.1. The AR model and TSK fuzzy rules

Fuzzy rules are one of the main elements of fuzzy systems. When applied to Time Series, as seen in equation (2.8), page 20, fuzzy rules can describe the relationship between the lagged variables in some parts of the state-space. A close look into this equation suggests the following

Proposition 4.1.1. *When used for Time Series modelling, a TSK fuzzy rule can be seen as a local AR model, applied on the state-space subset defined by the rule antecedent.*

Proof. Let us compare the expression of a fuzzy rule applied on Time Series problems, equation (2.8), page 20, and the expression for an autoregressive model, equation (3.1), page 40. Ignoring the white noise process, the consequent of the rule is exactly an AR model. The antecedent of the rule, as we know, defines the necessary conditions for the rule to be applied, that is, the state-space subset where the rule is applicable. □

This connection between the two models opens the possibility of an exchange of knowledge from one field to another, enabling us to apply what we know about AR models to fuzzy rules and vice versa. From the point of view of the Box-Jenkins models, this kind of fuzzy rules represents a local AR model which is

applied only when some conditions hold. These conditions are given by the terms in the rule antecedent, and are expressed as the fuzzy membership degree of the lagged variables to some fuzzy sets describing parts of the state-space domain. This scheme is closely related to the structure of the Threshold Autoregressive family of models, and in section 4.4.2 we will cover the consequences of this fact.

Regarding fuzzy rules, its relationship with autoregressive models may allow us to use the knowledge gathered through years about identification and estimation of those models to develop new, more appropriate methods to fit the consequent of fuzzy rules. This issue is addressed in section 4.4.1.

An example

Let us consider the following AR process:

$$y_t = 2.1 + 0.01y_{t-1} - 0.1y_{t-2} + \varepsilon_t, \tag{4.1}$$

which can be seen as a definition of the relationship between the "output" variable y_t and the "input" variables, y_{t-1} and y_{t-2}. This relationship can be displayed graphically as shown in Fig. 4.1 (a).

We might build a fuzzy rule whose consequent is exactly the aforementioned AR model:

$$\text{IF } y_{t-2} \text{ IS } A_1 \text{ AND } y_{t-1} \text{ IS } A_2 \quad \text{THEN } y_t = 2.1 + 0.01y_{t-1} - 0.1y_{t-2} + \varepsilon_t, \tag{4.2}$$

where (ignoring multiplicative constants)

$$A_i(x) = \exp\left(\frac{-(x-\mu_i)^2}{2\sigma_i^2}\right) \quad i = 1,2 \tag{4.3}$$

are the membership functions of the fuzzy variables of the rule's antecedent. A graphical representation of this fuzzy rule is shown in Fig. 6.1 (b), in which $\mu_1 = \mu_2 = 2.5$ and $\sigma_1 = \sigma_2 = 2.0$.

From the graphical representation shown in Fig. 4.1, it is fairly clear that the application of the fuzzy rule amounts to the application of the AR(2) model in the state-space subset defined by the membership of the "input" variables to the membership functions defined for its antecedent. It must also be noted that this state-space subset is a fuzzy subset, and hence its borders are not crisp.

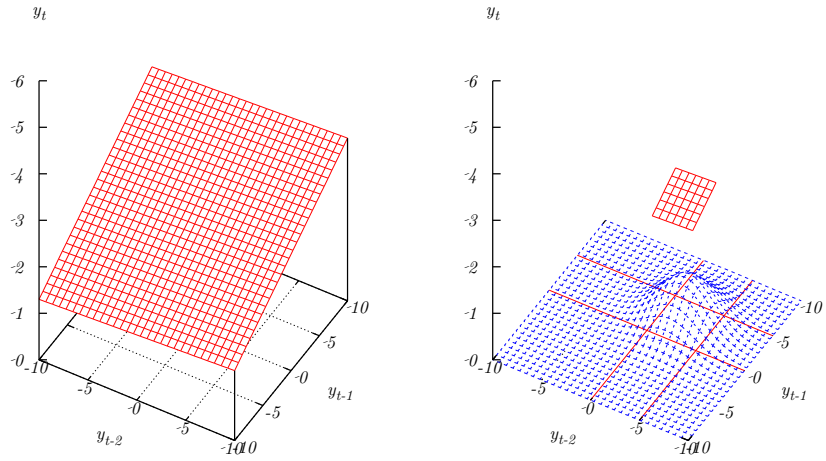

Figure 4.1.: (a) Plane defined by the AR(2) model (4.1). (b) Graphical representation of the fuzzy rule which makes use of the AR(2) model.

4.2. STAR model and fuzzy inference systems

With the previous result in mind, we are able to go further in the exploration of the relationships between threshold models and fuzzy logic-based models. On the one hand, we have seen that AR models are good linear models applicable to prediction problems. As well, we know that a TAR model is basically a set of local AR models, and that it allows for some nonlinearity in its computations. On the other hand, we have seen how a fuzzy rule relates to an AR model, in Proposition 4.1.1. Knowing that fuzzy inference systems contain sets of fuzzy rules, we may be interested in considering the relationship existing between

threshold models and fuzzy inference systems.

It is rather clear that there is some parallelism between the two aforementioned families of models. At a high level, models from both sides are composed of a set of elements (AR – fuzzy rules) which happen to be closely related, as stated above. On a lower level, both families of models rely on building a hyper-surface on the state-space which tries to model the relationship between the lagged variables of a time series. Moreover, both define this hyper-surface as the composition of hyper-planes which apply only in certain parts of the state-space. This can be seen clearly in figure 4.2, which shows the graphical representation of the fuzzy inference system or the STAR model.

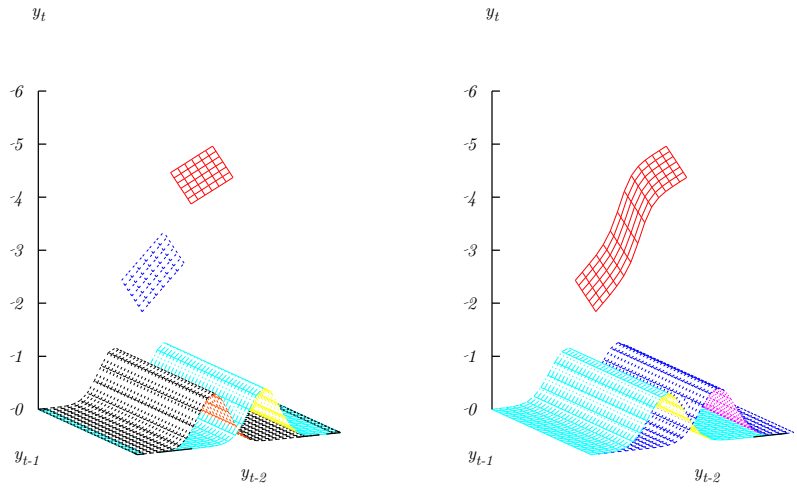

Figure 4.2.: (a) Two local AR models (or two fuzzy rules) (b) The STAR model (or the fuzzy inference system) derived from the two AR (or rules) shown in (a).

Indeed, we can prove the following

Proposition 4.2.1. *The STAR model is functionally equivalent to an Additive TSK FRBM with only one term in the rule antecedents*

Proof. We must recall the expression of a STAR model, equation (3.13), page 47, and the expression of the inference mechanism of a TSK AFM, equation (2.10), page 21. By looking at both expressions, we know that for both models to compute the same thing it is necessary that

1. $p(\mathbf{y}) = b_0 + b_1 y_{t-1} + \ldots + b_p y_{t-p} + \varepsilon_t$. Linear combination of the inputs are used in both cases.

2. $G(y_{t-d}) = w$, which means that w must be computed as the product of a single membership function, that is, $G(y_{t-d}) = w = \mu_A(y_{t-d})$.

   This condition holds independently of the smoothing function used by the STAR model, either logistic, exponential, Gaussian or any other, because FRBM can use any of these functions. This also suggests the study of STAR models which would use other families of functions.

Consequently, the type of FIS which satisfies the equivalence relationship uses the following type of fuzzy rules:

$$\text{IF } y_{t-d} \text{ IS } A_1 \text{ THEN } y_t = b_1 y_{t-p} + b_2 y_{t-p+1} + \ldots + b_p y_{t-1} + b_{p+1}, \quad (4.4)$$

which are just a special case of the regular TSK rule applied to time series. □

4.3. Advanced threshold models and fuzzy inference systems

As described in section 3.3, recent developments of the threshold autoregressive family of models include the AR-NN, the LGNN and the NCSTAR models. We will now explore the consequences of Proposition 4.1.1 regarding those models.

4.3.1. Autoregressive neural network (AR-NN)

Recalling equation (3.16), it is clear that the AR-NN is composed of an AR linear term and a neural network:

$$y_t = \underbrace{\boldsymbol{\omega}_1 \mathbf{x}_t}_{AR} + \underbrace{\sum_{i=2}^{k} \lambda_i \mathrm{f}(\boldsymbol{\omega}_i \mathbf{x}_t)}_{NN} + \varepsilon_t$$

The neural network term is a regular multilayered perceptron, and, as such, is interpretable as a fuzzy additive system, in the way shown in [6]. This work states as well that, by using the interactive-or operator, it is possible to view artificial neural networks as Mamdani-type fuzzy inference systems.

Furthermore, under the FRBM paradigm, the AR term of the AR-NN can be considered as a *generic* rule, that is, a rule which applies on the whole domain of the problem. Such generic rules, which fire unconditionally, produce a default answer which is added to the values of the fired rules on those areas covered by them. This type of rules has been used previously by researchers and practitioners to encode knowledge which is domain-wide applicable.

Thus, we can prove the following

Proposition 4.3.1. *The Autoregressive Neural Network (AR-NN) model is functionally equivalent to a TSK FRBM with a default rule.*

Proof. Using the result in [6], which states that a neural network is functionally equivalent to an FRBS, and considering the AR term as a rule of type

$$\text{IF true THEN } y_t = \boldsymbol{\omega}_1 \mathbf{x}_t, \tag{4.5}$$

the proof is trivial. □

Viewing the AR-NN as a combination of an AR model and a fuzzy inference system allows for linguistic interpretation of the system, and, amongst other things, this let us include *a priori* expert knowledge into the model. Other advantages of this equivalence relationship will be addressed in sections 4.4.1 and 4.4.2.

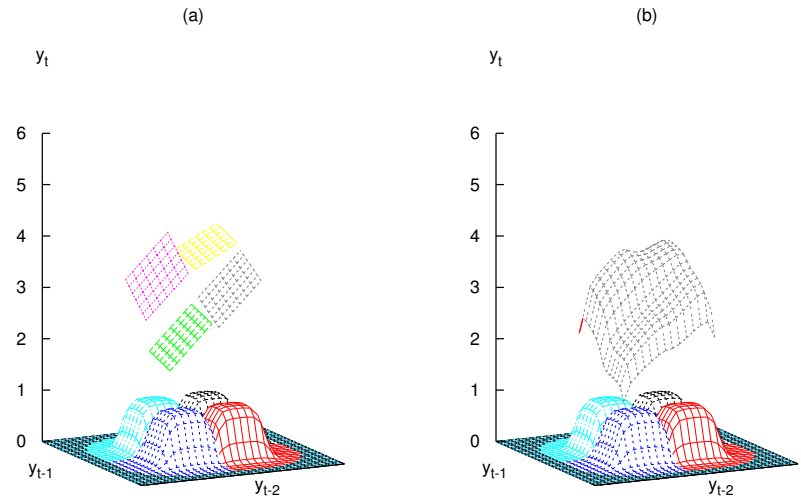

Figure 4.3.: (a) Four local AR models (or fuzzy rules) (b) The L^2GNN model (or the fuzzy inference system) derived from them.

4.3.2. Local global neural network

The more general approach of LGNN models, closely related to mixtures of experts model, satisfies the following

Theorem 4.3.2. *Local Global Neural Networks are a generalisation of Additive TSK FRBM.*

Proof. It is straightforward by looking at the expression of TSK rules (2.8), on page 20, and the expression for the LGNN (3.17), page 51. Since $L(z_t;\phi_{l_i})$ can take any form, it can also be a linear function of the inputs, which is exactly a TSK rule. As the aggregation rule for LGNN is additive, we can conclude that

the LGNN model is a generalisation of Additive TSK FRBM. □

For the same reason that the original formulation of the LGNN only pays attention to linear approximation functions (linear LGNN models), this type of fuzzy rules have not been used too much in the Soft Computing literature. Furthermore, it is generally preferred to keep the consequents linear and to encode all the nonlinearity in the antecedents.

If linear consequent were used (i.e. in the L^2GNN model), though, the relationship with Additive TSK FRBM is immediate:

Proposition 4.3.3. *Linear Local Global Neural Network (L^2GNN) models are functionally equivalent to Additive TSK FRBM.*

4.3.3. Neuro-coefficient smooth transition autoregressive models

This kind of models introduces time varying coefficients to combine AR models. Their mathematical formulation, however, is quite similar to the L^2GNN model. The only difference between the L^2GNN model and the NCSTAR is the form of their activation level functions. As we have seen, the L^2GNN uses a special type of bell-shaped function B, equation (3.18), page 52, which has $p+3$ degrees of freedom (given by the parameter vector $\boldsymbol{\psi}_B = [\gamma, \boldsymbol{\varphi}, \beta^{(1)}, \beta^{(2)}]$). In a complete L^2GNN model there are thus $(p+3) \times k$ parameters for the activation level functions (there are as many activation functions as hidden neurons).

On the other hand, the NCSTAR model uses the differences of exponential functions, F, as defined in equations (3.14), on page 48. As before, there are k functions: as many as hidden neurons. Hence, as the degrees of freedom of these functions are $p+2$ (given by the parameter vector $[\gamma, \boldsymbol{\varphi}, \beta]$), there are $(p+2) \times k$ parameters for the activation level functions.

There are k less parameters in the NCSTAR model, which is due to the fact that each pair of hidden units (and hence each pair of activation level functions) share one of their parameters.

Aside from this difference in the number of degrees of freedom of each model, both use a bell-shaped function which can be used in the definition of fuzzy variables. Hence the NCSTAR can be expressed also as an Additive TSK FRBM.

Moreover, both functions are derivable, so gradient-descent based learning is directly applicable to them.

When it comes to study links of these models to FRBM, the answer is similar to those obtained for the previous statistical models. It can be expressed in terms of the following

Proposition 4.3.4. *Neuro-Coefficient Smooth Transition Autoregressive (NC-STAR) models are functionally equivalent to Additive TSK FRBM.*

Proof. By looking at equations (2.10) (Additive TSK FRBM) and (3.22) (NC-STAR model), the proof is trivial using the proofs of theorems 4.2.1 and 4.3.2. □

Finally, the following theorem condenses the results drawn above:

Theorem 4.3.5. *The TSK FRBM is a generalization of the threshold models TAR, STAR, AR-NN, L^2GNN and NCSTAR.*

Proof. Trivial in the light of propositions 4.1.1, 4.2.1, 4.3.1, 4.3.3 and 4.3.4. □

4.4. Consequences and implications

Theorems 4.2.1, 4.3.1, 4.3.2 and 4.3.4 entail important implications that may affect the way threshold models and FRBM are understood and used. Since each of these threshold models can be expressed as a fuzzy rule based model, all the properties and tools of this Soft Computing approach are directly applicable to it. The opposite is also true: tools and properties of threshold models are valid for fuzzy inference systems. Some examples follow.

4.4.1. Soft Computing implications

One of the major criticism of the Soft Computing models has historically been the lack of mathematical proofs for their statistical properties. This situation is starting to change nowadays. In [62] a coherent modelling strategy which relies on statistical inference is presented to build artificial neural networks for Time Series. After our results a similar strategy can be applied to fuzzy systems.

The aforementioned modelling strategy, for example, uses a "bottom-up" strategy to build the model, consisting of the three stages usually applied in statistical modelling: *specification*, *estimation* and *evaluation*. The three stages rely on

well-known statistical procedures: in the specification stage, a variable selection is performed based on linearising the model and applying statistical techniques to choose the variables. Estimation of the parameters, i.e. the number of hidden units, is done by maximum likelihood. This procedure makes it possible to obtain an idea of the uncertainty in the parameter estimates through (asymptotic) standard deviation estimates. This is not possible using the common variable selection algorithms. Finally, evaluation of the model is performed through two in-sample misspecification tests: the first one tests for the instability of the parameters and the second one tests the assumption of no serial correlation in the errors.

Another example, is given by the main property of the L^2GNN model: it is asymptotically stationary under mild conditions (see Theorem 1 of [76]). After our Proposition 4.3.3, the same can be said about fuzzy additive systems.

For the NCSTAR model, an equivalent three-stage procedure is given (the evaluation stage is explained in [63]). This is again directly applicable to fuzzy additive systems based on Theorem 4.3.4. In this case, specification is performed through a sequence of Lagrange Multiplier tests, which are also used to evaluate the model's parameter constancy, serial independence and constant error variance.

The effective application of these modelling strategies to fuzzy systems may help overcome the traditional distrust affecting some scientific areas with respect to Soft Computing.

4.4.2. Statistical implications

The expression of a threshold model as a set of fuzzy rules has an immediate advantage: the model may be interpretable in terms of human language. Two consequences of this fact allow for an improved use of threshold models:

1. There exists the possibility of extracting linguistic knowledge from a tuned model, in order to contrast it with the knowledge of a human expert. The advantages of this are clear: the human expert could learn from the model and improve her or his knowledge of the problem.

2. There exists the possibility of incorporating linguistic knowledge to the models. This allows for a human expert to *teach* the model about specific

parts of the problem which could be hard to capture for the building procedure. As well, it is possible to give initial values to the modelling algorithm based on the expert's knowledge instead of using other criteria.

Another consequence of this contribution is that the building strategy for threshold models could use the Soft Computing advances in automatic model specification and estimation. For example, clustering techniques could be used to fix the number and boundaries of the local regimes of threshold models. As well, a myriad of Soft Computing optimisation techniques are at hand to be applied in fine-tuning the threshold models parameters.

5. Statistical approach to Fuzzy Rule-based time series modelling

This chapter deals with one of the consequences of the theorems derived in Chapter 4. We study the statistical properties of Fuzzy Rule-based Models: stationarity, identifiability. We also derive linearity tests against FRBM, as well as an incremental specification procedure and diagnostic checking tools.

5.1. Motivation

As stated before, a fundamental objection argued by scientists with a classical statistical background against Soft Computing models in general and neural networks and FRBM in particular was the lack of a sound theory behind them. Not being able to prove *a priori* if such models had good statistical properties (their much praised 'black-box' condition) prevented them to be accepted by wide parts of the scientific community despite its good performance in practical situations. Fuzzy-related researchers' and practitioners' attitude towards this has usually been to work from an engineering point of view and to further extend the practical applications of the models and methods in hope that their empirical benefits were at some point good enough as to finally convince the scientific community.

The results presented in Chapter 4 have an immediate impact on this question, as they permit the derivation of a statistical approach to a family of Soft Computing models, namely the FRBM family, considering them as nonlinear time series models.

This includes *a priori* proofs of their statistical properties, such as stationarity or identifiability, which will throw some light on their inner behaviour. Also, the use of log-likelihood based estimation methods allow us to guarantee existence, convergence, consistence and asymptotic normality of the estimators, properties that are a must for a statistical model to be accepted. As well, linearity tests grant the ability to decide, based on the data, if a series can be modelled with a single linear autoregressive model or if a FRBM seems appropriate instead. These tests will also be used to iteratively decide if a model has enough rules to capture the dynamics of the data or if, on the contrary, more complexity should be introduced in the model in the form of new rules. This decision is crucial in developing a *bottom-up* building strategy which results in more parsimonious models, with a rule base whose complexity is adapted to the complexity of the data. Finally, diagnostic checks over the residuals will bring the possibility of determining whether a model is effectively capturing the inner properties of a dataset or not.

In order to derive this statistical approach, a word on notation must be said. In the standard FRBM framework, the residuals are considered as an information source about the 'goodness of fit' of the model. They are looked at once the model is built, as they are the basis for computing the so-called *error measures*: mean squared error, mean average error and so on.

In the statistical field, on the other hand, the time series formed by the residuals, $\{\varepsilon_t\}$, is a fundamental piece of the modelling process, and as such it is always included in the definition of the models. Hence, in this Chapter we will define the Additive TSK FRBM, Equation (2.10), in the time series framework as

$$y_t = \mathrm{G}\left(\mathbf{x}_t;\boldsymbol{\psi}\right) + \varepsilon_t = \sum_{i=1}^{r} p_i\left(\mathbf{x}_t;\boldsymbol{\psi}_{p_i}\right)\cdot\mu_i\left(\mathbf{x}_t;\boldsymbol{\psi}_{\mu_i}\right) + \varepsilon_t$$
$$= \sum_{i=1}^{r} \mathbf{b}_i\mathbf{x}_t\cdot\mu_i\left(\mathbf{x}_t;\boldsymbol{\psi}_{\mu_i}\right) + \varepsilon_t, \quad (5.1)$$

where $\boldsymbol{\psi} = (\boldsymbol{\psi}_p, \boldsymbol{\psi}_\mu)$ is the parameter vector, including the consequent (linear) parameters, $\boldsymbol{\psi}_p = (\mathbf{b}_1,...,\mathbf{b}_r)$ and the antecedent (nonlinear) parameters, $\boldsymbol{\psi}_{\mu_i}$, whose number depends on the type of membership function, μ, used. The residuals, ε_t, are hence included in the definition of the FRBM.

In this chapter, we will consider two types of membership functions: sigmoid, μ_{S}, and Gaussian, μ_{G}. The sigmoid function is the one used in [64], and although

it is not so common in the fuzzy literature, we will use it here as an immediate result derived from the equivalences stated in the previous Chapter. As we know, it is defined as

$$\mu_{\mathsf{S}}(\mathbf{x}_t;\boldsymbol{\psi})) = \frac{1}{1+\exp\left(-\gamma(\boldsymbol{\omega}\mathbf{x}_t - c)\right)}, \tag{5.2}$$

where $\boldsymbol{\psi} = (\gamma, \boldsymbol{\omega}, c)$.

On the other hand, Gaussian function will also be used because it is the most common membership function in fuzzy models. It is usually expressed as

$$\mu_{\mathsf{G}}(\mathbf{x}_t;\boldsymbol{\psi})) = \prod_i \exp\left(-\frac{(x_i - c_i)^2}{2\sigma^2}\right) \tag{5.3}$$

but we will rewrite it throughout this Chapter as

$$\mu_{\mathsf{G}}(\mathbf{x}_t;\boldsymbol{\psi})) = \prod_i \exp\left(-\gamma(x_i - c_i)^2\right), \tag{5.4}$$

where $\boldsymbol{\psi} = (\gamma, \mathbf{c})$.

5.2. Statistical properties of FRBM for time series analysis

Deriving necessary and sufficient conditions for stationarity or identifiability of nonlinear time series models is usually not easy, and this is of course the case for the FRBM. To our notice, the question of whether a FRBM is stationary or not has never been addressed, not surprisingly considering the facts stated above.

Stationarity of the model is one of the most central questions in linear time series theory. A model is stationarity if the probabilistic structure of the series that it generates is constant over time, or at least asymptotically constant (when not started in equilibrium).

Identifiability, on the other hand, understood as the uniqueness and minimality of a specification of the model, is also crucial for statistical inference, as no test can be derived if it is not guaranteed.

In order to define the conditions that ensure stationarity and identifiability, we will make use of the equivalence relationships described in Chapter 4 above, and follow [64, 76] amongst others, to translate these properties from regime switching autoregressive models to FRBM.

5.2.1. Asymptotic stationarity of the model

Stationarity of a random process is related to the mean value and variance of the observation data, both of which should be constant over time, and the covariance between the observations x_t and x_{t-d} should only depend on the distance d between the two observations and does not change over time. A time series is *weakly stationary* if $\mathrm{E}(x_t) = \mu$ and $cov(x_t, x_{t+h}) = \kappa_h$, $\forall t$, i.e., means and covariances do not depend on time t. A stronger criterion is that the whole distribution (and not only mean and covariance) of the process does not depend on time, and in this case it is called *strictly stationary*. Strong stationarity implies weak stationarity if the second moments of the series exist [54].

If $\{x_t\}$ is strictly stationary, then $\mathrm{P}(x_t \in A) = \pi(A)$, $\forall t$, and $\pi(\cdot)$ is called the *stationary distribution* of the series. Obviously the series can only be stationary from the beginning if it is started with the stationary distribution such that $x_0 \sim \pi$. If it is not started with π, e.g., because x_0 is a constant, then we call the series *asymptotically stationary* if it converges to its stationary distribution:

$$\lim_{t\to\infty} \mathrm{P}(x_t \in A) = \pi(A). \tag{5.5}$$

In order to study the asymptotic properties of the threshold autoregressive family of models the concept of *characteristic equation* was introduced. The characteristic equation of a FRBM can be defined as

$$\lambda^p - c_1\lambda^{p-1} - c_2\lambda^{p-2} - \cdots - c_p = 0 \tag{5.6}$$

with

$$c_j = \sum_{i=1}^{r} \|b_{ij}\|,\ j = 1,\ldots,d \tag{5.7}$$

being b_{ij} the jth coefficient of the ith linear model and r the number of linear models (the number of rules).

It is easy to verify that model (5.1) has a finite number of limiting linear models of the form

$$y_t = c_0^{(k)} + c_1^{(k)} y_{t-1} + \cdots + c_0^{(k)} y_{t-p} + \varepsilon_t. \tag{5.8}$$

Obviously, when all the limiting linear models of a model are asymptotically stationary (their roots are inside the unit circle), the model cannot be but asymptotically stationary itself. On the other hand, if one or more of the limiting linear

models have roots outside the unit circle, or have unit roots, we must study the model carefully, and its stationarity depends on the membership functions used.

As explained in [76], if the membership functions are "large", being "active" in half the space, then an explosive limiting regime will lead to asymptotically nonstationary model with probability strictly greater than 0. This is the case with sigmoid functions, and that is the reason why an FRBM using the sigmoid function (equation (2.15)) as membership functions is not guaranteed to be asymptotically stationary.

The problem is different, though, if Gaussian membership functions (equation (2.13)) are used, as these are "small" in the sense that they cover a small fraction of any sufficiently large sphere. This is similar to the case for the L^2GNN, which uses as membership function the difference of two sigmoid functions. The main difference is that the L^2GNN's membership functions are active in the infinite space left between two parallel hyperplanes, whilst the Gaussian functions are active only inside the limited space of a hypersphere.

Theorem 5.2.1. *The Additive TSK FRBS is asymptotically stationary if it uses Gaussian membership functions with* $\gamma \neq 0$.

Proof. Trivial after Theorem 1 of [76]. With respect to the L^2GNN, the conditions for an explosive or unit-root limiting linear model to escape to infinity are much simpler: the membership function must have value 1 ($\gamma = 0$) or it will always return close to the origin. □

5.2.2. Identifiability of the model

If we consider the use of FRBM as Statistical modelling, we can see it as a procedure to specify the probability of the observations by a family of distributions, indexed by parameters. This procedure includes the statistical inference, the simulation and the prediction. All these depend on identifiable models, so it is important to study the identifiability conditions for FRBM.

For example, we must explicitly specify the sources of uniqueness of the model in order to guarantee convergence of the mean squared error (MSE) estimator function. This issue has been deeply studied in the nonlinear statistical models framework, including the feedforward neural network [84] and some derived models [76, 64].

Here we will adapt those results for the Gaussian FRBM model, stating under which conditions identifiability is guaranteed. In order to do so, we will first discuss the concepts of minimality [79] or "nonredundancy" [36] and the concept of model reducibility.

Definition 5.2.2. An FRBM model is *minimal* (or *nonredundant*) if its input-output map cannot be obtained from another FRBM with fewer rules.

One of the sources of unidentifiability in an FRBM is the presence of irrelevant rules, that can be removed without affecting its modelling capabilities. Obviously, the minimality condition holds only for irreducible models.

In this chapter, we will consider two types of FRBM, attending to their membership functions: sigmoid and gaussian. Sigmoid function are those used in [64], and although they are not so common in the fuzzy literature, we will use them here as an immediate result derived from the equivalences stated in the previous chapter. As we know, it is defined as

$$\mu_{\mathrm{S}}(\boldsymbol{x}_t;\boldsymbol{\psi})) = \frac{1}{1+\exp\left(-\gamma(\boldsymbol{\omega}\mathbf{x}_t - c)\right)}. \tag{5.9}$$

On the other hand, Gaussian functions will be also used because they are the most common membership function in fuzzy models. They are given by

$$\mu_{\mathrm{G}}(\boldsymbol{x}_t;\boldsymbol{\psi})) = \prod_i \exp\left(-\frac{(x_i - c_i)^2}{2\sigma^2}\right). \tag{5.10}$$

Definition 5.2.3. An FRBM model is *reducible* if one of the following conditions hold:

i. Some of the consequents of the rules vanish ($\mathbf{b}_k \to 0$ for some k).

ii. Some of the membership functions vanish ($\mu_k \to 0$ for some k).

Furthermore, we can define the property of identifiability as

Definition 5.2.4. An FRBM is *identifiable* if there are no two sets of parameters such that the corresponding input-output maps are identical.

There are two properties of FRBM that cause unidentifiability:

(P. 1) The *interchangeability* of the rules. The order in which rules are considered is totally irrelevant for the computations of the model but affects the search in the parameter space (giving place to multiple local maxima for the log-likelihood function).

(P. 2) The presence of *irrelevant* rules, i.e., if there is at least one rule with zero consequent ($\mathbf{b}^k = 0$ for some k) or if the conjunction of the membership functions is zero for at least one rule ($\mu_k \to 0$ for some k).

If we ensure that the model is irreducible, then we know that the only way to change the input-output map is through property (P. 1). This can be achieved in the style of [64] by applying a "specific-to-general" model building strategy based on statistical inference through Lagrange Multiplier (LM) linearity tests.

As it was proved in [36, 79], an irreducible model is minimal. This equivalence implies that there are no means, apart from the conditions stated in Definition 5.2.3 of reducibility, to further reduce the number of rules of a FRBM without changing the functional input-output map.

The problem of interchangeability of rules (P. 1) can be prevented by establishing a unique order among them. This might be ensured by defining (and forcing) a lexicographical order, $\prec$, among the rule antecendent parts. We first establish an order among every variable's membership functions, which is induced by the ordering of their location parameters $c_{d,k}$ (independently of their width or steepness, i.e., γ_k). This ordering is usually given by their linguistic definition. Then, to compare (and sort) the rules, we apply the lexicographical order, which would result in the following Restriction:

$$\mu(\mathbf{x}_t; \boldsymbol{\psi}_{\mu_k}) \prec \mu(\mathbf{x}_t; \boldsymbol{\psi}_{\mu_{k+1}}), \quad k = 1, \ldots, K \tag{R. 1}$$

This restriction defines a complete ordering for rules, which would allow us to write $R_i \prec R_{i+1}$.

By imposing (R. 1), we prevent the interchangeability of rules. We can thus guarantee that, if irrelevant rules do not exist, the model is identifiable and minimal.

In order to formally state the sufficient conditions under which the FRBM model is globally identifiable, and following [76], we need the following assumptions.

Assumption 5.2.5. *The linear parameters* $\mathbf{b}_k$ *do not vanish for any* k*. Furthermore,* $\gamma_k > 0 \forall k$*.*

Assumption 5.2.6. *The covariate vector* $\mathbf{x}_t$ *has an invariant distribution that has a density everywhere positive in an open ball.*

Assumption 5.2.5 prevents from the effects of property (P. 2) and Assumption 5.2.6 avoids problems related to multicollinearity.

We will also make use of the following assumptions related to linear independence of the transition functions:

Lemma 5.2.7. *The family of n-dimensional sigmoid cumulative distribution functions is linearly independent.*

Proof. Trivial in light of Proposition 2 and Theorem of [97]. □

Lemma 5.2.8. *The family of n-dimensional Gaussian cumulative distribution functions is linearly independent.*

Proof. Straightforward after the linear independence of exponential functions. □

This allows us to state the following

Theorem 5.2.9. *Under restriction (R. 1) and Assumptions 5.2.5 and 5.2.6, the TSK additive FRBM with Gaussian membership functions is globally identifiable.*

Proof. Let us suppose two vector of parameters, $\boldsymbol{\psi} = [\boldsymbol{\psi}'_\mu, \boldsymbol{\psi}'_f]'$ and $\overline{\boldsymbol{\psi}} = [\overline{\boldsymbol{\psi}_\mu}', \overline{\boldsymbol{\psi}_f}']'$ such that

$$\sum_{i=1}^{K} \tilde{\mu}(\mathbf{x}_t; \boldsymbol{\psi}_{\mu_i}) f(\mathbf{x}_t; \boldsymbol{\psi}_{f_i}) = \sum_{j=1}^{K} \tilde{\mu}(\mathbf{x}_t; \overline{\boldsymbol{\psi}_{\mu_j}}) f(\mathbf{x}_t; \overline{\boldsymbol{\psi}_{f_j}}). \tag{5.11}$$

To prove global identifiability of the FRBM we need to show that, under restriction (R. 1) and the assumptions, (5.11) is satisfied if and only if $\boldsymbol{\psi}_{\mu_k} = \overline{\boldsymbol{\psi}_{\mu_k}}$ and $\boldsymbol{\psi}_{f_k} = \overline{\boldsymbol{\psi}_{f_k}}$ for $k = 1, \ldots, K$.

Assumption 5.2.5 clearly excludes the possibility of (5.11) being true when both sides of the equality are zero, so we shall study the other possibilities.

To ease the notation, we will note $\mu_i(\mathbf{x}_t) = \tilde{\mu}(\mathbf{x}_t; \boldsymbol{\psi}_{\mu_i})$, $\overline{\mu_j}(\mathbf{x}_t) = \tilde{\mu}(\mathbf{x}_t; \overline{\boldsymbol{\psi}_{\mu_j}})$, $\mathbf{b}_i\mathbf{x}_t = f(\mathbf{x}_t; \boldsymbol{\psi}_{f_i})$ and $\overline{\mathbf{b}_i}\mathbf{x}_t = f(\mathbf{x}_t; \overline{\boldsymbol{\psi}_{f_j}})$ henceforth in this proof. We can thus rewrite (5.11) as:

$$\sum_{i=1}^{K} \mu_i(\mathbf{x}_t)\mathbf{b}_i\mathbf{x}_t - \sum_{j=1}^{K} \overline{\mu_j}(\mathbf{x}_t)\overline{\mathbf{b}_j}\mathbf{x}_t = 0 \tag{5.12}$$

This equality can be true under two different situations:

i) If every $\mu_i(\mathbf{x}_t)$ is different from every $\overline{\mu_j}(\mathbf{x}_t)$, then we know that $\mathbf{b}_i\mathbf{x}_t = \overline{\mathbf{b}_j}\mathbf{x}_t = 0$, by Assumptions 5.2.7 and 5.2.8.

Obviously, this would contradict Assumption 5.2.5.

ii) There exist i_1, j_1 such that $\mu_{i_1}(\mathbf{x}_t) = \overline{\mu_{j_1}}(\mathbf{x}_t)$.

We know that $\mu_l(\mathbf{x}_t) \neq \mu_m(\mathbf{x}_t)$ for $l \neq m$ and $\overline{\mu_l}(\mathbf{x}_t) \neq \overline{\mu_m}(\mathbf{x}_t)$ for $l \neq m$. Hence, we could write (5.12) as

$$\left(\mathbf{b}_{i_1} - \overline{\mathbf{b}_{j_1}}\right)\mathbf{x}_t\mu_{i_1}(\mathbf{x}_t) + \sum_{\substack{i=1 \\ i\neq i_1}}^{K} \mathbf{b}_i\mathbf{x}_t\mu_i(\mathbf{x}_t) - \sum_{\substack{j=1 \\ j\neq j_1}}^{K} \overline{\mathbf{b}_j}\mathbf{x}_t\overline{\mu_j}(\mathbf{x}_t) = 0 \tag{5.13}$$

This equation is similar to (5.12) in that it would be true under the same two situations. Hence, following the same rationale, we could further write it as

$$\left(\mathbf{b}_{i_1} - \overline{\mathbf{b}_{j_1}}\right)\mathbf{x}_t\mu_{i_1}(\mathbf{x}_t) + \left(\mathbf{b}_{i_2} - \overline{\mathbf{b}_{j_2}}\right)\mathbf{x}_t\mu_{i_2}(\mathbf{x}_t) + \sum_{\substack{i=1 \\ i\neq i_1 \\ i\neq i_2}}^{K} \mathbf{b}_i\mathbf{x}_t\mu_i(\mathbf{x}_t) - \sum_{\substack{j=1 \\ j\neq j_1 \\ j\neq j_2}}^{K} \overline{\mathbf{a}_j}\mathbf{x}_t\overline{\mu_j}(\mathbf{x}_t) = 0 \tag{5.14}$$

Hence, we can proceed inductively (in k steps) up to

$$\left(\mathbf{b}_{i_1} - \overline{\mathbf{b}_{j_1}}\right)\mathbf{x}_t\mu_{i_1}(\mathbf{x}_t) + \ldots + \left(\mathbf{b}_{i_K} - \overline{\mathbf{b}_{j_K}}\right)\mathbf{x}_t\mu_{i_K}(\mathbf{x}_t) = 0, \tag{5.15}$$

which, as all the $\mu_{i_k}(\mathbf{x}_t)$ are distinct and hence linearly independent, forces $\mathbf{b}_{i_k} = \mathbf{b}_{j_k}$ for every k, resulting in $\boldsymbol{\psi} = \overline{\boldsymbol{\psi}}$. It also remarkable that, in 5.15, actually $i_k = j_k$ for $k = 1, \ldots, K$ because restriction (R. 1) holds, q.e.d.

There is an alternative proof as follows: [30] stated the functional equivalence between fuzzy rule-based systems and Gaussian mixtures. In particular, TSK rule-based systems were proven to be equivalent to Gaussian mixtures of *equal priors*. Using this result, and knowing that Proposition 2 in [97] guarantees the identifiability of Gaussian mixtures, restriction (R. 1) gives as a result identifiable fuzzy rule-based systems. □

Theorem 5.2.9 applies to Additive TSK FRBMs using Gaussian membership functions. The extention to other types of membership functions is straightforward as long as we can derive a result similar to Assumptions 5.2.7 and 5.2.8.

Another question worth to study is the effect of the restrictions posed by Theorem 5.2.9 concerning the input space fuzzy partitions allowed. One might think that the rule ordering restriction can somehow limit the validity of the result to just some cases of TSK FRBM.

As the reader might know, there are two main ways to partition the input space in the fuzzy subspaces which are covered by each rule. One alternative, called *grid partition* consists in setting a number of one-dimensional membership functions on each dimension and use as many rules as combinations of different membership functions there are. This results in every part of the input space covered by at least one rule. The other alternative is called *patched partition* and places multidimensional membership functions only in relevant parts of the space.

Actually, the ordering restriction allows for two rules to share at most all the one-dimensional membership functions but one. This is the usual situation when we have a grid type input space partition. Of course, it also allows for no multidimensional membership functions being shared amongst two rules, which is the case for patched type partition. Hence both main input space fuzzy partitioning schemes are covered by Theorem 5.2.9.

5.3. Linearity tests for FRBM

Since FRBM can be seen as nonlinear regression models, the standard procedures for testing parameter significance, like LM-tests, should be applicable, in principle. To perform these tests, however, the asymptotic distribution of the

model parameters must be known. This issue is dealt with in Section 5.4.2, where it is shown that the parameters of a FRBM are asymptotically normal.

In the fuzzy literature, however, no attention has been paid to hypothesis testing up to now. While it is obvious that a linear time series should be modelled with a linear model, i.e. a single (default) rule, to our knowledge there is no testing procedure to avoid the mistake of using highly complex structures to model simple problems.

Next we propose a statistical test to decide if a problem can be solved using a linear model or if we need a combination of rules to model it. We will first study the alternative of using a FRBM with sigmoid membership functions as defined in (5.2), page 68, and then we will turn to testing against a Gaussian (5.4) FRBM.

5.3.1. Logistic membership function, μ_S

Suppose that we have a FRBM composed of a single linear model which applies to the whole input space,

$$y_t = G(\mathbf{x}_t; \boldsymbol{\psi}) = \mathbf{b}_0 \mathbf{x}_t + \varepsilon_t. \tag{5.16}$$

Now we want to know if the use of an extra rule with logistic membership functions would increase the performance of the model. We would add that rule as follows:

$$y_t = G(\mathbf{x}_t; \boldsymbol{\psi}_p, \boldsymbol{\psi}_\mu) = \mathbf{b}_0 \mathbf{x}_t + \mathbf{b}_1 \mathbf{x}_t \mu_S(\mathbf{x}_t; \boldsymbol{\psi}) + \varepsilon_t. \tag{5.17}$$

where $\boldsymbol{\psi}_p = (\mathbf{b}_0, \mathbf{b}_1)$ is the so called vector of linear parameters and $\boldsymbol{\psi}_\mu = (\gamma, \boldsymbol{\omega}, c)$ contains the nonlinear parameters. In order to test for linearity, the logistic membership function is redefined as

$$\mu_S(\mathbf{x}_t; \boldsymbol{\psi}) = \frac{1}{1 + \exp(-\gamma(\boldsymbol{\omega}\mathbf{x}_t - c))} - \frac{1}{2}. \tag{5.18}$$

Substracting one half from the membership function is useful just in deriving linearity tests because it simplifies the notation, but it does not affect the generality of the argument. In the estimating phase, the models do not contain that term.

Our goal is to test for the significance of the extra rule, so an appropriate null hypothesis could be

$$H^0 : \gamma = 0, \tag{5.19}$$

being the alternative $H^1 : \gamma > 0$. Hypothesis (5.19) opens up the possibility of studying linearity in the Lagrange Multiplier (LM) testing framework. Under this null hypothesis, the contribution of the extra rule, C, is identically equal to a constant and merges with the intercept b_{00} of the default rule, that is, the rule is not necessary.

We assume that, under (5.19), the maximum likelihood estimators of the parameters of (5.18) are asymptotically normal and hence can be estimated consistently (as granted by Theorem 5.4.2, section 5.4, page 82).

As it was thoroughly discussed in section 5.2.2, model (5.18) is only identified under the alternative hypothesis, i.e., if the null is true, the parameters are not locally unique and thus the estimator does not follow an asymptotic normal distribution. This issue is known as the problem of 'hypothesis testing when a nuissance parameter is present only under the alternative', and was first studied by [19]. In this situation, the test statistic of the LM-test does not follow a known distribution and thus the standard asymptotic distribution theory for the likelihood ratio is not available.

However, we can avoid this difficulty and obtain a χ^2-statistic by following the method first suggested in [58] and then widely applied to neural network-based models by [84, 64, 76] amongst others. This method proposes the expansion of the expression of the firing strength of a fuzzy rule using logistic membership functions into a Taylor series around the null hypothesis $\gamma_1 = 0$:

$$\tilde{\mu}_{\mathsf{S},1}(\mathbf{x}_t;\boldsymbol{\psi}) = \mu_{\mathsf{S}}(\mathbf{x}_t;0,\boldsymbol{\omega},c) + \left.\frac{\partial \mu_{\mathsf{S}}}{\partial \gamma}\right|_{\gamma=0} \gamma + R(\mathbf{x}_t;\boldsymbol{\psi}) = \frac{1}{4}\gamma(\boldsymbol{\omega}\mathbf{x}_t - c) + R_1(\mathbf{x}_t;\boldsymbol{\psi}) \tag{5.20}$$

which for the expression of the contribution of the extra rule yields

$$C \approx \mathbf{b}_1\mathbf{x}_t\left[\frac{1}{4}\gamma(\boldsymbol{\omega}\tilde{\mathbf{x}}_t - c)\right] = \theta_0 + \sum_{i=1}^{q}\theta_i x_i + \sum_{i=1}^{q}\sum_{j=i}^{q}\theta_{ij}x_i x_j. \tag{5.21}$$

The first linear term and the intercept θ_0 merge with the system's default rule, while the remainder of the Taylor expansion adds up to the error term, it becoming $\varepsilon^\star = \varepsilon + \mathbf{b}_1\mathbf{x}_t R(\mathbf{x}_t;\boldsymbol{\psi})$, which means that $\varepsilon^\star = \varepsilon$ under the null. Thus the expansion results in the following model:

$$y_t = \boldsymbol{\pi}'\mathbf{x}_t + \sum_{i=1}^{q}\sum_{j=1}^{q}\theta_{ij}x_i x_j + \varepsilon_t^\star \tag{5.22}$$

Using (5.22) instead of (5.17) circumvents the identification problem, and we obtain a simple test of linearity. The null hypothesis can be defined as $\mathrm{H}^0 : \theta_{ij} = 0$. However, the parameters θ_{ij} do not depend on b_{10}. Thus, when the only nonlinear element in (5.17) is the intercept, the test has no power. To remedy this situation, [58] suggests a third-order Taylor approximation of the transition function, expressed as

$$\tilde{\mu}_{\mathrm{S},3}(\mathbf{x}_t;\boldsymbol{\psi}) = \frac{1}{4}\gamma(\boldsymbol{\omega}\mathbf{x}_t - c) + \frac{1}{48}\gamma^3(\boldsymbol{\omega}\mathbf{x}_t - c)^3 + R_3(\mathbf{x}_t;\boldsymbol{\psi}) \tag{5.23}$$

Now, the expression of the contribution of the extra rule yields

$$C \approx \theta_0 + \sum_{i=1}^{q}\theta_i x_i + \sum_{i=1}^{q}\sum_{j=i}^{q}\theta_{ij}x_i x_j + \sum_{i=1}^{q}\sum_{j=i}^{q}\sum_{k=j}^{q}\theta_{ijk}x_i x_j x_k + \sum_{i=1}^{q}\sum_{j=i}^{q}\sum_{k=j}^{q}\sum_{l=k}^{q}\theta_{ijkl}x_i x_j x_k x_l, \tag{5.24}$$

which finally gives a new expression for the model as

$$y_t = \boldsymbol{\pi}'\mathbf{x}_t + \sum_{i=1}^{q}\sum_{j=i}^{q}\theta_{ij}x_i x_j + \sum_{i=1}^{q}\sum_{j=i}^{q}\sum_{k=j}^{q}\theta_{ijk}x_i x_j x_k + \sum_{i=1}^{q}\sum_{j=i}^{q}\sum_{k=j}^{q}\sum_{l=k}^{q}\theta_{ijkl}x_i x_j x_k x_l + \varepsilon_t^{\star}. \tag{5.25}$$

The null hypothesis is now defined as

$$\mathrm{H}_0 : \theta_{ij} = 0 \wedge \theta_{ijk} = 0 \wedge \theta_{ijkl} = 0 \;\forall\; i,j,k,l \in 1,\ldots,q. \tag{5.26}$$

This null hypothesis circumvents the identification problem, and allows us to obtain a statistical test concerning the use of the nonlinear term, the extra rule. This test is based on the local approximation to the log-likelihood for observation t, which takes the form (ς is the variance of ε)

$$l_t = -\frac{1}{2}\ln(2\pi) - \frac{1}{2}\ln\varsigma^2 - \frac{1}{2\varsigma^2}\left\{ y_t - \boldsymbol{\pi}'\mathbf{x}_t - \sum_{i=1}^{q}\sum_{j=1}^{q}\theta_{ij}x_i x_j - \sum_{i=1}^{q}\sum_{j=1}^{q}\sum_{k=1}^{q}\theta_{ijk}x_i x_j x_k - \sum_{i=1}^{q}\sum_{j=i}^{q}\sum_{k=j}^{q}\sum_{l=k}^{q}\theta_{ijkl}x_i x_j x_k x_l \right\}^2. \tag{5.27}$$

In order to build the test statistic we need to make the following assumptions, as in [64]:

Assumption 5.3.1. *The* $((r+1)\times 1)$ *parameter vector defined by* $[\boldsymbol{\psi}', \varsigma^2]'$ *is an interior point of the compact parameter space* Ψ *which is a subspace of* $\mathbb{R}^r \times \mathbb{R}^+$*, the r dimensional Euclidean space.*

Assumption 5.3.2. *Under the null hypothesis, the data generating process (DGP) for the sequence of scalar real valued observations* $y_t{}_{t=1}^{T}$ *is an ergodic stochastic process, with true parameter vector* $\boldsymbol{\psi} \in \Psi$.

Assumption 5.3.3. $E|z_{t,i}|^{\delta} < \infty, \forall i \in \{1,\ldots,p\}$ *for some* $\delta > 8$.

Under H^0 and Assumptions 5.3.1, 5.3.2 and 5.3.3 we can compute the standard Lagrange Multiplier or score-type test statistic given by

$$\mathrm{LM} = \frac{1}{\hat{\sigma}^2}\sum_{t=1}^{T}\hat{\varepsilon}\hat{\boldsymbol{\tau}}_t' \times \left\{\sum_{t=1}^{T}\hat{\boldsymbol{\tau}}_t\hat{\boldsymbol{\tau}}_t' - \sum_{t=1}^{T}\hat{\boldsymbol{\tau}}_t\hat{\mathbf{h}}_t' \times \left(\sum_{t=1}^{T}\hat{\mathbf{h}}_t'\hat{\mathbf{h}}_t\right)^{-1} \times \sum_{t=1}^{T}\hat{\mathbf{h}}_t\hat{\boldsymbol{\tau}}_t'\right\} \times \sum_{t=1}^{T}\hat{\boldsymbol{\tau}}_t'\hat{\varepsilon} \quad (5.28)$$

where $\hat{\varepsilon} = y_t - \hat{\boldsymbol{\pi}}'\mathbf{x}_t$ are the residuals estimated under the null hypothesis,

$$\hat{\mathbf{h}}_t = \left.\frac{\partial G(\mathbf{x}_t;\boldsymbol{\phi},\boldsymbol{\psi})}{\partial\hat{\boldsymbol{\phi}}\partial\hat{\boldsymbol{\psi}}}\right|_{\boldsymbol{\phi}=\hat{\boldsymbol{\phi}}\wedge\boldsymbol{\psi}=\hat{\boldsymbol{\psi}}} \quad (5.29)$$

is the gradient of the model and $\hat{\boldsymbol{\tau}}_t$ contains all the nonlinear regressors in (5.25). This statistic has an asymptotic χ^2 distribution with m degrees of freedom.

This test can be carried out in stages, as shown below

1. Regress y_t on $\mathbf{x}_t$ and compute the residual sum of squares $SSR_0 = \sum_{t=1}^{T}\hat{\varepsilon}_t{}^2$
2. Regress $\hat{\varepsilon}_t$ on $\mathbf{x}_t$ and on $\hat{\boldsymbol{\tau}}_t$, the m nonlinear regressors of (5.25). Compute the residual sum of squares $SSR_1 = \sum_{t=1}^{T}\hat{\varsigma}_t{}^2$.
3. Compute the χ^2 statistic

$$\mathrm{LM}_{\chi^2}^{l} = T\frac{SSR_0 - SSR_1}{SSR_0} \quad (5.30)$$

or the F version of the test

$$\mathrm{LM}_{F}^{l} = \frac{(SSR_0 - SSR_1)}{m}\left(\frac{SSR_1}{(T-p-1-m)}\right)^{-1}. \quad (5.31)$$

If the value of the test statistic exceeds the appropriate value of the χ^2 or F distribution, the null hypothesis is rejected.

5.3.2. Gaussian membership function, μ_{G}

Next we turn our attention towards deriving a linearity test against a system using one nonlinear rule which has, as a membership function, the Gaussian function. Again, let us suppose that we have a FRBM composed of a single linear model which applies to the whole input space, as in equation (5.16), page 76.

Now we want to know if the use of an extra rule with Gaussian membership function would increase the performance of the model. We would add that rule as follows:

$$y_t = \mathrm{G}(\mathbf{x}_t;\boldsymbol{\phi},\boldsymbol{\psi}) = \mathbf{b}_0\mathbf{x}_t + \mathbf{b}_1\mathbf{x}_t\mu_{\mathrm{G}}(\mathbf{x}_t;\boldsymbol{\psi}) + \varepsilon_t. \tag{5.32}$$

If our FRBM uses Gaussian membership functions, with $\boldsymbol{\psi} = [\gamma, \mathbf{c}]$, we might rewrite it as

$$y_t = \mathbf{b}_0\mathbf{x}_t + \mathbf{b}_1\mathbf{x}_t \prod_i \exp\left(-\gamma(x_i - c_i)^2\right) + \varepsilon_t.$$

Our goal is to test for the significance of the extra rule, so in this case an appropriate null hypothesis could be

$$\mathrm{H}^0 : \gamma = 0, \tag{5.33}$$

being the alternative $\mathrm{H}^1 : \gamma > 0$. As was the case with the logistic membership function, we will study linearity in the Lagrange Multiplier (LM) testing framework. Under this null hypothesis, the contribution of rule 1, C, is identically equal to a constant and merges with the intercept b_{00} of the default rule, that is, the rule is not necessary.

Again, we must assume that, under (5.33), the maximum likelihood estimators of the parameters of (5.3.2) are asymptotically normal and hence can be estimated consistently (as granted by Theorem 5.4.2, section 5.4, page 82).

To circumvent the identifiability problem of 'hypothesis testing when a nuissance parameter is present only under the alternative', we again follow the method suggested in [58], and we face the expansion of the expression of the firing strength of a fuzzy rule using Gaussian membership functions into a Taylor series around the null hypothesis $\gamma = 0$:

$$\begin{aligned}\mu_{\mathrm{G}}(\mathbf{x}_t;\gamma,\mathbf{c}) \approx \mu_{\mathrm{G}}(\mathbf{x}_t;0,\mathbf{c}) + \left.\frac{\partial\mu_{\mathrm{G}}}{\partial\gamma}\right|_{\gamma=0}\gamma + R(\mathbf{x}_t;\gamma,\mathbf{c}) \\ = \gamma\sum(x_i - c_i)^2 + R(\mathbf{x}_t;\gamma,\mathbf{c})\end{aligned} \tag{5.34}$$

which for the expression of the contribution of the extra rule yields

$$C \approx \mathbf{b}_1\mathbf{x}_t\left[\gamma\sum(x_i-c_i)^2\right] = \sum_{i=1}^{q}\theta_i x_i + \sum_{i=1}^{q}\sum_{j=i}^{q}\theta_{ij}x_i x_j + \sum_{i=1}^{q}\sum_{j=i}^{q}\sum_{k=j}^{q}\theta_{ijk}x_i x_j x_k.$$

In this case, contrary to what happened when using the logistic membership function, the first order Taylor approximation is enough for our needs, as all the $\theta_i, \theta_{ij}, \theta_{ijk}$ depend on the intercept, b_{10}, of (5.32). The first linear term merges with the system's default rule, while the remainder of the Taylor expansion adds up to the error term, becoming $\varepsilon^\star = \varepsilon + \mathbf{b}_1\mathbf{x}_t R(\mathbf{x}_t;\gamma,\mathbf{c})$, which means that $\varepsilon^\star = \varepsilon$ under the null. Thus the expansion results in the following model:

$$y_t = \boldsymbol{\pi}'\mathbf{x}_t + \sum_{i=1}^{q}\sum_{j=1}^{q}\theta_{ij}x_i x_j + \sum_{i=1}^{q}\sum_{j=i}^{q}\sum_{k=j}^{q}\theta_{ijk}x_i x_j x_k + \varepsilon_t^\star. \tag{5.35}$$

The null hypothesis can hence be defined as

$$\mathrm{H}_0 : \theta_{ij} = 0 \wedge \theta_{ijk} = 0 \;\forall\; i,j,k \in 1,\ldots,q. \tag{5.36}$$

This null hypothesis circumvents the identification problem, and allows us to obtain a statistical test concerning the use of the extra rule. This test is based on the local approximation to the log-likelihood for observation t, which takes the form (ς is the variance of ε):

$$l_t = -\frac{1}{2}\ln(2\pi) - \frac{1}{2}\ln\varsigma^2 - \frac{1}{2\varsigma^2}\left\{y_t - \boldsymbol{\pi}'\mathbf{x}_t - \sum_{i=1}^{q}\sum_{j=1}^{q}\theta_{ij}x_i x_j - \sum_{i=1}^{q}\sum_{j=1}^{q}\sum_{k=1}^{q}\theta_{ijk}x_i x_j x_k\right\}^2. \tag{5.37}$$

As before, we must rely on Assumptions 5.3.1, 5.3.2 and 5.3.3 (page 79), which allow us to compute the standard Lagrange Multiplier or score-type test statistic given by

$$\mathrm{LM} = \frac{1}{\hat{\sigma}^2}\sum_{t=1}^{T}\hat{\varepsilon}\hat{\boldsymbol{\tau}}_t' \times \left\{\sum_{t=1}^{T}\hat{\boldsymbol{\tau}}_t\hat{\boldsymbol{\tau}}_t' - \sum_{t=1}^{T}\hat{\boldsymbol{\tau}}_t\hat{\mathbf{h}}_t' \times \left(\sum_{t=1}^{T}\hat{\mathbf{h}}_t'\hat{\mathbf{h}}_t\right)^{-1} \times \sum_{t=1}^{T}\hat{\mathbf{h}}_t\hat{\boldsymbol{\tau}}_t'\right\} \times \sum_{t=1}^{T}\hat{\boldsymbol{\tau}}_t'\hat{\varepsilon} \tag{5.38}$$

where $\hat{\varepsilon} = y_t - \hat{\pi}'\mathbf{x}_t$ are the residuals estimated under the null hypothesis,

$$\hat{\mathbf{h}}_t = \left.\frac{\partial G(\mathbf{x}_t;\boldsymbol{\phi},\boldsymbol{\psi})}{\partial\hat{\boldsymbol{\phi}}\partial\hat{\boldsymbol{\psi}}}\right|_{\boldsymbol{\phi}=\hat{\boldsymbol{\phi}}\wedge\boldsymbol{\psi}=\hat{\boldsymbol{\psi}}} \tag{5.39}$$

is the gradient of the model and $\hat{\boldsymbol{\tau}}_t$ contains all the nonlinear regressors in (5.35). This statistic has an asymptotic χ^2 distribution with m degrees of freedom.

Exactly as was the case with the FRBM using logistic membership function, this test can be carried out in stages:

1. Regress y_t on $\mathbf{x}_t$ and compute the residual sum of squares $SSR_0 = \sum_{t=1}^{T} \hat{\varsigma}_t{}^2$
2. Regress $\hat{\varsigma}_t$ on $\mathbf{x}_t$ and on the m nonlinear regressors of (5.35). Compute the residual sum of squares $SSR_1 = \sum_{t=1}^{T} \hat{\tau}_t{}^2$.
3. Compute the χ^2 statistic

$$\mathrm{LM}^l_{\chi^2} = T\frac{SSR_0 - SSR_1}{SSR_0}$$

or the F version of the test

$$\mathrm{LM}^l_F = \frac{(SSR_0 - SSR_1)}{m}\left(\frac{SSR_1}{(T-p-1-m)}\right)^{-1}.$$

If the value of the test statistic exceeds the appropriate value of the χ^2 or F distribution, the null hypothesis is rejected.

5.4. Estimation procedures. Properties of the estimator

The incremental building procedure that we will define in Section 5.5 requires estimation of FRBM. In this section we will focus on this problem.

There is a growing number of algorithms in the literature for estimating the parameters of FRBM and Neural Network based models. Following the reasons argued in [84, 64], we choose to estimate the parameters of the model by maximum likelihood, making use of the assumptions made previously on ε_t.

Estimation through maximum likelihood has been applied to Neural Networks but, to our notice, it has never been applied as is to FRBM. This is another idea that we borrow from classical statistic time series analysis, and it makes it feasible to obtain an idea of the uncertainty in the parameter estimates through asymptotic standard deviation estimates, which is something hardly possible through metaheuristic algorithms. It may be argued, though, that maximum

likelihood estimation of neural network models is most likely to lead to convergence problems, and that penalizing the log-likelihood function is a necessary precondition for satisfactory results. Notwithstanding, in this case, proceeding in a bottom-up manner avoids the estimation of unidentified models (a main reason for penalizing the log-likelihood), and also that the initial values for the parameters are carefully chosen.

If, as we assume here, ε_t is a Gaussian white noise with zero mean and finite variance, $\varepsilon_t \backsim \text{NID}(0,\sigma^2)$, the maximum likelihood is equivalent to nonlinear least squares. Hence, the parameter vector $\boldsymbol{\psi}$ of the model is estimated as

$$\hat{\boldsymbol{\psi}} = \arg\min_{\boldsymbol{\psi}} Q_T(\boldsymbol{\psi}) = \arg\min_{\boldsymbol{\psi}} \sum_{t=1}^{T} (y_t - G(\mathbf{x}_t;\boldsymbol{\psi}))^2. \tag{5.40}$$

The least squares estimator (LSE) defined by (5.40) belongs to the class of M estimators considered by [69]. We next discuss the conditions that guarantee existence, consistency and asymptotic normality of the LSE.

5.4.1. Existence of the estimator

Following [76], the existence of the LSE estimator is based in Lemma 2 of [41], which establishes the existence under certain conditions:

The proof of existence is based on Lemma 2 of [41], which establishes that the LSE exists under certain conditions of continuity and measurability on the mean squared error (MSE) function.

Theorem 5.4.1. *The Additive TSK FRBM satisifies the following conditions and the LSE exists:*

a. *For each* $\mathbf{x}_t \in \mathbf{X}$*, function* $G(\mathbf{x}_t;\boldsymbol{\psi})$ *is continuous in a compact subset* Ψ *of the Euclidean space.*

b. *For each* $\boldsymbol{\psi} \in \Psi$*, function* $G(\mathbf{x}_t;\boldsymbol{\psi})$ *is measurable in space* $\mathbf{X}$*.*

c. ε_t *are independent and identically distributed errors with mean* 0 *and variance* σ^2*.*

Proof. Lemma 2 of [41] shows that conditions a-c in Theorem 5.4.1 are sufficient to guarantee the existence (and measurability) of the LSE. To apply this result

to the FRBM, we need to check whether these conditions are satisfied by the model.

Condition c of Theorem 5.4.1 was already assumed when defining the model. It is easy to prove in our case that $G(\mathbf{x}_t;\boldsymbol{\psi})$ is continuous in the parameter vector $\boldsymbol{\psi}$. This follows from the fact that $p(\mathbf{x}_t;\boldsymbol{\psi}_\mu)$ and $\mu(\mathbf{x}_t;\boldsymbol{\psi}_p)$ depend continuously on $\boldsymbol{\psi}_p$ and $\boldsymbol{\psi}_\mu$ for each value of $\mathbf{x}_t$. Similarly we can see that $G(\mathbf{x}_t;\boldsymbol{\psi})$ is continuous in $\mathbf{x}_t$ and thus is measurable, for each fixed value of the parameter vector $\boldsymbol{\psi}$. Thus, conditions a and b are also satisfied. □

5.4.2. Consistence and Asymptotic Normality of the estimator

White [92, 93] established the conditions that guarantee strong consistency of the LSE. In the context of stationary time series models, the conditions that ensure (almost certain) consistency have been established in [91, 95]. Now, having proved Theorem 5.2.9, which guarantees the global identifiability of the model, we can prove existence, consistency and asymptotic normality of the FRBM estimators.

Theorem 5.4.2. *Under Assumptions 5.3.1, 5.3.2 and Theorem 5.2.9, the maximum likelihood estimator $\hat{\boldsymbol{\psi}}$ is almost surely consistent for $\boldsymbol{\psi}$ and*

$$\sqrt{T}(\hat{\boldsymbol{\psi}} - \boldsymbol{\psi}) \xrightarrow{D} N\left(0, -\operatorname*{plim}_{T\to\infty} \mathbf{A}(\boldsymbol{\psi}^{-1})\right) \tag{5.41}$$

where

$$\mathbf{A}(\boldsymbol{\psi}^{-1}) = \left(\frac{1}{\sigma^2 T}\right)\left(\frac{\partial^2 Q_T(\boldsymbol{\psi})}{\partial\boldsymbol{\psi}\partial\boldsymbol{\psi}'}\right).$$

Proof. To prove consistency, we will make use of Theorem 3.5 of [91] and follow [64]. We must show that the assumptions of that theorem are fulfilled in the case of the FRBM:

Assumptions 2.1 and 2.3, related to the probability space and to the density functions, are trivial. Let $q(\mathbf{x}_t;\boldsymbol{\psi}) = (y_t - G(x_t;\boldsymbol{\psi}))^2$. Assumption 3.1a states that for each $\boldsymbol{\psi} \in \Psi$, $-\mathrm{E}(q(\mathbf{x}_t;\boldsymbol{\psi}))$ exists and is finite for $t = 1,...,T$. Under Assumption 5.3.2 and the fact that ε_t is a zero mean normally distributed random variable with finite variance, hence, k-integrable, Assumption 3.1a in [91] follows.

Assumption 3.1b states that $-\mathrm{E}(q(\mathbf{x}_t;\boldsymbol{\psi}))$ is continuous in Ψ, $t = 1,...,T$. Let $\boldsymbol{\psi} \to \boldsymbol{\psi}^*$, since for any t, $G(x_t;\boldsymbol{\psi})$ is continuous on Ψ, then $q(\mathbf{x}_t;\boldsymbol{\psi}) \to q(\mathbf{x}_t;\boldsymbol{\psi}^*)$, $\forall t$ (pointwise convergence). From the continuity of $G(x_t;\boldsymbol{\psi})$ on the compact set Ψ, we have uniform continuity and we obtain that $q(\mathbf{x}_t;\boldsymbol{\psi})$ is dominated by an integrable function dF. Then, by Lebesgue's dominated convergence theorem, we get $\int q(\mathbf{x}_t;\boldsymbol{\psi})dF \to \int q(\mathbf{x}_t;\boldsymbol{\psi}^*)dF$ and $\mathrm{E}(q(\mathbf{x}_t;\boldsymbol{\psi}))$ is continuous.

Assumption 3.1c states that $-\mathrm{E}(q(\mathbf{x}_t;\boldsymbol{\psi}))$ obeys the strong (weak) law of large numbers (ULLN). Lemma 2 of [69] guarantees that $\mathrm{E}(q(\mathbf{x}_t;\boldsymbol{\psi}))$ obeys the strong law of large numbers. The set of hypothesis (b) of this lemma is satisfied:

a. we deal with an ergodic process,

b. from the continuity of $\mathrm{E}(q(\mathbf{x}_t;\boldsymbol{\psi}))$ and from the compactness of Ψ we have that $\inf \mathrm{E}(q(\mathbf{x}_t;\boldsymbol{\psi})) = \mathrm{E}(q(\mathbf{x}_t;\boldsymbol{\psi}^*))$ for $\boldsymbol{\psi}^* \in \Psi$, and with Assumption 3.1a in [91] we may guarantee that $\mathrm{E}(q(\mathbf{x}_t;\boldsymbol{\psi}^*))$ exists and is finite, getting that $\mathrm{E}(q(\mathbf{x}_t;\boldsymbol{\psi}^)) > -\infty$.

Assumption 3.2 is related to the unique identifiability of $\boldsymbol{\psi}^*$, which is guaranteed by Theorem 5.2.9.

Now, to prove normality, we will use Theorem 6.4 of [91] and will check its assumptions. Assumptions 2.1, 2.3 and 3.1 follow from the proof of consistency showed above. Assumptions 3.2 and 3.6 follow from the fact that $G(x_t;\boldsymbol{\psi})$ is continuously differentiable of order 2 on $\boldsymbol{\psi}$ in the compact space Ψ.

In order to check Assumptions 3.7a and 3.8a we have to prove that $\mathrm{E}\left(\nabla Q_T(\boldsymbol{\psi})\right) < \infty$ and $\mathrm{E}\left(\nabla^2 Q_T(\boldsymbol{\psi})\right) < \infty$, $\forall T$. The expected gradient and the expected Hessian of $Q_T(\boldsymbol{\psi})$ are given by

$$\mathrm{E}\left(\nabla Q_T(\boldsymbol{\psi})\right) = 2\mathrm{E}\left(\nabla G(x_t;\boldsymbol{\psi})(y_t - G(x_t;\boldsymbol{\psi}))\right)$$

and

$$\mathrm{E}\left(\nabla^2 Q_T(\boldsymbol{\psi})\right) \quad = \quad 2\mathrm{E}\left(\nabla G(x_t;\boldsymbol{\psi})\nabla G(x_t;\boldsymbol{\psi})' - \nabla^2 G(x_t;\boldsymbol{\psi})(y_t - G(x_t;\boldsymbol{\psi}))\right)$$

respectively. Assumption 3.7a and 3.8a follow considering the normality condition on ε_t, the properties of the function $G(x_t;\boldsymbol{\psi})$ and the fact that $\nabla G(x_t;\boldsymbol{\psi})$ and $\nabla^2 G(x_t;\boldsymbol{\psi})$ contain at most second order terms of $\mathbf{x}_t$.

Assumption 3.8c is guaranteed by the proof of consistency and the ULLN from [69].

Assumption 3.9 follows from the identifiability of the FRBM and the properties of function $G(x_t;\boldsymbol{\psi})$.

Assumption 6.1 requires using Theorem 2.4 of [93], by which we can show that $2\xi'\nabla G(x_t;\boldsymbol{\psi}^*)\varepsilon_t$ obeys the central limit theorem for some $(r\times 1)$ vector ξ, such that $\xi\xi'=1$. Assumptions A(i) and A(iii) both hold because ε_t is Gaussian. Assumption A(ii) also holds with $V=4\sigma^2\xi'\mathrm{E}(\nabla G(x_t;\boldsymbol{\psi}^*)\nabla' G(x_t;\boldsymbol{\psi}^*))$. Furthermore, since any measurable transformation of mixing processes is itself mixing (see [93, Lemma 2.1]), hence we have that $2\xi'\nabla G(x_t;\boldsymbol{\psi}^*)\varepsilon_t$ is a strong mixing sequence and obeys the central limit theorem. $\nabla Q_T(\boldsymbol{\psi})$ also obeys the CLT with covariance matrix $B_T^*=4\sigma^2\mathrm{E}(\nabla G(x_t;\boldsymbol{\psi}^*)\nabla' G(x_t;\boldsymbol{\psi}^*))=2\sigma^2A_T^*$, which is $O(1)$ and nonsingular. □

5.5. Determining the number of rules of a FRBM

Once we have developed the statistical theory for the FRBM, including the linearity tests, we are closer to establishing a sound statistical procedure to specify the structure of a FRBM. This specification includes the determination of the number of fuzzy rules that are sufficient to model a given time series.

Knowledge included in a FRBM is represented by fuzzy rules. Obtaining these rules is a fundamental problem in the design process of a FRBM. When an expert on the system or domain under study is available, he or she can deliver the rules. Its elicitation is a knowledge acquisition process which is affected by many well-known problems described in the literature [37] (chap. 5), [34] (chap. 3).

This is the reason why, opposed to traditional interview-based techniques, some alternatives are proposed, based in automatic learning methods. The idea is to use one of these techniques to capture or learn a set of examples that describe the behaviour of the system. Later, when this set is fixed, we translate this knowledge into fuzzy rules. This approach has given birth to a myriad of procedures to extract fuzzy rules, based on diverse algorithms or automated learning models, including classification trees, evolutive algorithms, clustering techniques, logic and neural networks. For a review on recent developments on this issue, see [35].

Notwithstanding, the most common procedure for automatic rule base determination remains the one proposed by Wang and Mendel [89] in 1992. This procedure is based in a combinatorial approach, and divides the universe of

discourse into fuzzy regions, assigning rules to those regions which cover the available data. It was already covered in section 2.4.2, algorithm 1.

Some disadvantages of these methods are the high number of rules they produce and, again, their lack of a mathematical foundation that justifies their proceedings. In this section, we will propose an alternative method, based on the formal developments that we have carried out throughout this Chapter, that overcomes these limitations. The method relies on hypothesis testing and produces parsimonious models because it proceeds in a bottom-up manner.

Suppose that we have a FRBS with $r+1$ fuzzy rules (disregarding the default rule), and write it as follows:

$$y_t = \mathrm{G}\left(\mathbf{x}_t;\boldsymbol{\psi}\right) + \varepsilon_t = \sum_{i=1}^{r} \mathbf{b}_i \mathbf{x}_t \cdot \mu_i\left(\mathbf{x}_t;\boldsymbol{\psi}_{\mu_i}\right) + \mathbf{b}_{r+1}\mathbf{x}_t \cdot \mu_{r+1}\left(\mathbf{x}_t;\boldsymbol{\psi}_{\mu_{r+1}}\right) + \varepsilon_t, \quad (5.42)$$

where $\mu(\mathbf{x}_t;\boldsymbol{\psi}_\mu)$ is either the logistic function (5.2) or the Gaussian function (5.4). Assume that we have accepted the hypothesis of model (5.42) containing h rules and we want to test for the $(h+1)$-th rule. An appropriate null hypothesis could be

$$\mathrm{H}_0 : \gamma_{h+1} = 0, \quad (5.43)$$

being the alternative $\mathrm{H}_1 : \gamma_{h+1} > 0$. Hypothesis (5.43) opens up the possibility of studying the number of rules required in the Lagrange Multiplier (LM) testing framework, exactly in the same way we used with the linearity test, Section 5.3.

Under this null hypothesis, the contribution of the $(r+1)$-th rule, C_{r+1}, is identically equal to a constant and merges with the intercept in the default rule, that is, the rule is not necessary.

At this point, the test is similar to the linearity one, so we proceed in the same way: to avoid unidentified parameters, expand into a Taylor series the contribution of the extra rule, redefine the null hypothesis using the parameters of the Taylor expansion and under Assumptions 5.3.1, 5.3.2 and 5.3.3 construct the standard Lagrange Multiplier test statistic.

5.6. Diagnostic Checking

In general, once a model is built and estimated, it has to be evaluated. This is true in the Soft Computing framework as well as in the classical Statistics

approach. By evaluating a model we understand to find out if the model satisfies a set of quality criteria that allow us to say if the interesting characteristics of the system under study are actually being captured by it or not.

Notwithstanding, this set of evaluation criteria is heavily dependent on several considerations: the final use that the model is built for, the inner characteristics of the system that are to be captured and whether the emphasis is put on the empirical behaviour of the model or if there are theoretical considerations that are considered to be more important. This is evident when we consider the evaluation means used in the Soft Computing field as opposed to those used in the statistical approach to time series analysis.

In the usually engineering-oriented Soft Computing framework, there has been an overwhelming preeminence of just one evaluation criterion, and this has been the *goodness of fit*. Generally, evaluation of a model consists on computing the prediction (or classification) error produced when it is faced with a previously unseen problem of the same type of the one used to estimate it. This measure, in its different flavours (mean squared error, mean average error and so on) is affected by some inherent limitations: it is not very meaningful for a single model unless compared against other models, and is usually range-dependent, which makes it difficult to compare the same model applied to different problems represented by data sets with different characteristics.

On the other hand, evaluation in the statistical approach to time series has usually more to do with obtaining an estimate of the probability that the model is effectively capturing the interesting characteristics of the data set, and this is achieved through developing hypothesis tests, also known as misspecification tests.

It is now when the aforementioned inclusion of the error term ε_t in the context of FRBM is justified. In Chapter 4, we introduced the main assumption behind modelling: a part of the system under study behaves according to a model but there is another part which cannot be explained by it and is usually considered to be white noise. This is the main idea encoded in the expression of the general model

$$y_t = \mathrm{G}(\mathbf{x}_t; \boldsymbol{\psi}) + \varepsilon_t, \tag{5.44}$$

and it is also behind some of the diagnostic checking procedures.

For instance, it is interesting to obtain a precise knowledge about the series of the residuals, $\{\varepsilon_t\}$, for example determining if its values are independent and

normally distributed. If the residuals were not independent, that would mean that the model is failing to capture an important part of the behaviour of the series, and hence it should be respecified.

Another desirable property that the model should satisfy refers to the variance of the series $\{\varepsilon_t\}$. If a model is properly capturing the inner behaviour of the series, the residuals should have the same variance at any point of the series. Failing to ensure this implies that the model's precission depends on time, and hence that there are parts of the state-space that are not properly modelled.

There is a third alternative to check the adequacy of the model to the series under study which refers to its parameters, which should remain constant. If a model is properly specified, its parameters should be the same at any point of the series, as varying parameters would indicate that the system enters regimes that are not considered by the model.

Following [63], we will develop these three testing approaches in the framework of FRBM.

5.6.1. Test of serial independence of the residuals

Consider the following FRBM with autocorrelated errors:

$$\begin{array}{rcl} y_t & = & \mathrm{G}(\mathbf{x}_t;\boldsymbol{\psi}) + \varepsilon_t = \sum_{i=1}^{r} \mathbf{b}_i \mathbf{x}_t \cdot \mu_i\left(\mathbf{x}_t;\boldsymbol{\psi}_{\mu_i}\right) + \varepsilon_t \\ \varepsilon_t & = & \boldsymbol{\pi}'\boldsymbol{\nu}_t + u_t \end{array} \tag{5.45}$$

where the $\boldsymbol{\pi}' = [\pi_1, \pi_2, ..., \pi_s]$ is a vector of parameters, $\boldsymbol{\nu}_t = [\varepsilon_{t-1}, \varepsilon_{t-2}, ..., \varepsilon_{t-s}]$ and $u_t \sim \mathrm{NID}(0, \sigma^2)$. We assume that ε_t is stationary, and furthermore, that under the assumption $\varepsilon_t \sim \mathrm{NID}(0, \sigma^2)$, that is, $\pi = \mathbf{0}$, $\{y_t\}$ is stationary and ergodic such that the parameters of (5.45) can be consistently estimated by nonlinear least squares.

In the context of this model, we can formulate the null hypothesis of serial independence of the residuals as $\mathrm{H}_0 : \boldsymbol{\pi} = \mathbf{0}$.

The conditional normal log-likelihood, given the fixed starting values, has the form

$$l_t = -\frac{1}{2}\ln(2\pi) - \frac{1}{2}\ln\varsigma^2 - \frac{1}{2\varsigma^2} \left\{ y_t - \sum_{j=1}^{s} \pi_j y_{t-j} - \mathrm{G}(\mathbf{x}_t;\boldsymbol{\psi}) + \sum_{j=1}^{s} \pi_j \mathrm{G}(\mathbf{x}_{t-j};\boldsymbol{\psi}) \right\}^2 . \tag{5.46}$$

The information matrix related to (5.46) is block diagonal such that the element corresponding to the second derivative of (5.46) forms its own block. The variance ς^2 can thus be treated as a fixed constant in (5.46) when deriving the test statistic. The first partial derivatives of the normal log-likelihood with respect to $\boldsymbol{\pi}$ and $\boldsymbol{\psi}$ are

$$\frac{\partial l_t}{\partial \pi_j} = \left(\frac{u_t}{\sigma^2}\right)\left\{y_{t-j} - \mathrm{G}(\mathbf{x}_{t-j};\boldsymbol{\psi})\right\}, j = 1,...,s \tag{5.47}$$

$$\frac{\partial l_t}{\partial \boldsymbol{\psi}} = -\left(\frac{u_t}{\sigma^2}\right)\left\{\frac{\partial \mathrm{G}(\mathbf{x}_t;\boldsymbol{\psi})}{\partial \boldsymbol{\psi}} - \sum_{j=1}^{s}\pi_j \frac{\partial \mathrm{G}(\mathbf{x}_{t-j};\boldsymbol{\psi})}{\partial \boldsymbol{\psi}}\right\} \tag{5.48}$$

Under the null hypothesis, the consistent estimators of (5.47) are

$$\left.\frac{\partial \hat{l}_t}{\partial \pi_j}\right|_{\mathrm{H}_0} = \frac{1}{\hat{\sigma}^2}\hat{\varepsilon}_t \hat{\boldsymbol{v}}_t \quad \text{and} \quad \left.\frac{\partial \hat{l}_t}{\partial \boldsymbol{\psi}}\right|_{\mathrm{H}_0} = -\frac{1}{\hat{\sigma}^2}\hat{\varepsilon}_t \hat{\mathbf{h}}_t, \tag{5.49}$$

where $\hat{\boldsymbol{v}}_t = [\hat{\varepsilon}_{t-1}, \hat{\varepsilon}_{t-2}, ..., \hat{\varepsilon}_{t-s}]$, $\hat{\varepsilon}_{t-j} = y_{t-j} - \mathrm{G}(\mathbf{x}_{t-j};\boldsymbol{\psi}), j = 1,...,s$, $\hat{\mathbf{h}}_t = \nabla \mathrm{G}(\mathbf{x}_t;\hat{\boldsymbol{\psi}})$ and $\hat{\sigma}^2 = (1/T)\sum_{t=1}^{T}\hat{\varepsilon}_t$.

The LM statistic is

$$\mathrm{LM} = \frac{1}{\hat{\sigma}^2}\sum_{t=1}^{T}\hat{\varepsilon}_t \hat{\boldsymbol{v}}_t' \times \left\{\sum_{t=1}^{T}\hat{\boldsymbol{v}}_t \hat{\boldsymbol{v}}_t' - \sum_{t=1}^{T}\hat{\boldsymbol{v}}_t \hat{\mathbf{h}}_t' \times \left(\sum_{t=1}^{T}\hat{\mathbf{h}}_t' \hat{\mathbf{h}}_t\right)^{-1} \times \sum_{t=1}^{T}\hat{\mathbf{h}}_t \hat{\boldsymbol{v}}_t'\right\} \times \sum_{t=1}^{T}\hat{\boldsymbol{v}}_t' \hat{\varepsilon}_t \tag{5.50}$$

where $\hat{\mathbf{h}}_t = \nabla \mathrm{G}(\mathbf{x}_t;\hat{\boldsymbol{\psi}})$ and $\hat{v}_t = \left[t\mathbf{x}_t', t\mathbf{x}_t'\mu_1\left(\mathbf{x}_t;\boldsymbol{\psi}_{\mu_1}\right), ..., t\mathbf{x}_t' \;\; \mu_s\left(\mathbf{x}_t;\boldsymbol{\psi}_{\mu_s}\right)\right]'$.

Under the condition that the moments implied by (5.50) exist, LM is asymptotically distributed as a χ^2 with s degrees of freedom.

As before, the test can be performed in three stages as follows:

1. Estimate model (5.1) under the assumption of uncorrelated errors and and compute the residuals $\hat{\varepsilon}_t$. Orthogonalize the residuals by regressing $\hat{\varepsilon}_t$ on $\hat{\mathbf{h}}_t$, and compute the residual sum of squares $SSR_0 = (1/T)\sum_{t=1}^{T}\tilde{\varepsilon}_t^2$.

2. Regress $\tilde{\varepsilon}_t$ on $\hat{\mathbf{h}}_t$ and $\hat{v}_t$. Compute the residual sum of squares $SSR_1 = (1/T)\sum_{t=1}^{T}\hat{v}_t^2$.

3. Compute the χ^2 statistic

$$\mathrm{LM}_{\chi^2}^{si} = T\frac{SSR_0 - SSR_1}{SSR_0}$$

or the F version of the test

$$\mathrm{LM}_F^{si} = \frac{(SSR_0 - SSR_1)}{s}\left(\frac{SSR_1}{(T-s-n)}\right)^{-1}.$$

Under H_0, $\mathrm{LM}_{\chi^2}^{si}$ is asymptotically distributed as a χ^2 with s degrees of freedom and LM_F^{si} has approximately an F distribution with s and $T-s-n$ degrees of freedom.

5.6.2. Test of homoscedasticity against smoothly changing variance of the residuals

Let us consider a test of constant variance against the following specification:

$$\sigma_t^2 = \sigma^2 + \sum_{i=1}^{r} \sigma_i^2 \mu_{\sigma,i}\left(\mathbf{x}_t; \boldsymbol{\psi}_{\mu_{\sigma,i}}\right) \tag{5.51}$$

where $\mu_{\sigma,i}$ are logistic or Gaussian function satisfying the identifiability restrictions defined in Section 5.2.2. This formulation allows the variance to change smoothly between regimes.

Following [63], we rewrite equation (5.51) as

$$\sigma_t^2 = \exp\left(\mathrm{G}_\sigma\left(\mathbf{x}_t; \boldsymbol{\psi}_\sigma, \boldsymbol{\psi}_{\mu_{\sigma,i}}\right)\right) = \exp\left(\varsigma + \sum_{i=1}^{r} \varsigma_i \mu_{\sigma,i}\left(\mathbf{x}_t; \boldsymbol{\psi}_{\mu_{\sigma,i}}\right)\right), \tag{5.52}$$

where $\boldsymbol{\psi}_\sigma = [\varsigma, \varsigma_1, ..., \varsigma_r]'$ is a vector of real parameters.

To derive the test, let us consider $r = 1$. This is not a restrictive assumption because the test statistic remains unchanged if $h > 1$. We rewrite model (5.52) as

$$\sigma_t^2 = \exp\left(\varsigma + \varsigma_1 \mu_\sigma\left(\mathbf{x}_t; \boldsymbol{\psi}_{\mu_\sigma}\right)\right), \tag{5.53}$$

where μ_σ is defined as (5.2) or as (5.4), page 68, depending on the membership function used by the model.

In both cases, logistic or Gaussian, the null hypothesis of parameter constancy is $\mathrm{H}_0 : \gamma_\sigma = 0$. As usual, model (5.53) is only identified under the alternative $\gamma_\sigma \neq 0$ and we expand the membership function into a first-order Taylor expansion

around $\gamma_\sigma = 0$. Replacing the function by its Taylor approximation and ignoring the remainder, both the logistic and the Gaussian case result in

$$\sigma_t^2 = \exp\left(\rho + \sum_{i=1}^{q} \rho_i x_{i,t}\right), \tag{5.54}$$

so the null hypothesis becomes $H_0 : \rho_1 = \rho_2 = ... = \rho_q = 0$. Under H_0, $\exp(\rho) = \sigma^2$.

The local approximation to the normal log-likelihood function in a neighbourhood of H_0 for observation t is

$$l_t = -\frac{1}{2}\ln(2\pi) - \frac{1}{2}\left(\rho + \sum_{i=1}^{q} \rho_i x_{i,t}\right) - \frac{\varepsilon_t^2}{2\exp(\rho + \sum_{i=1}^{q} \rho_i x_{i,t})}. \tag{5.55}$$

In order to derive a LM type test, we need the partial derivatives of the log-likelihood:

$$\frac{\partial l_t}{\partial \rho} = -\frac{1}{2} + \frac{\varepsilon_t^2}{2\exp(\rho + \sum_{i=1}^{q} \rho_i x_{i,t})}, \tag{5.56}$$

$$\frac{\partial l_t}{\partial \rho_i} = -\frac{x_i}{2} + \frac{\varepsilon_t^2 x_i}{2\exp(\rho + \sum_{i=1}^{q} \rho_i x_{i,t})}, \tag{5.57}$$

and their consistent estimators under the null hypothesis:

$$\left.\frac{\partial \hat{l}_t}{\partial \rho}\right|_{H_0} = \frac{1}{2}\left(\frac{\varepsilon_t^2}{\hat{\sigma}^2} - 1\right), \tag{5.58}$$

$$\left.\frac{\partial \hat{l}_t}{\partial \rho_i}\right|_{H_0} = \frac{x_{i,t}}{2}\left(\frac{\varepsilon_t^2}{\hat{\sigma}^2} - 1\right), \tag{5.59}$$

where $\hat{\sigma}^2 = 1/T\sum_{t=1}^{T} \hat{\varepsilon}_t^2$. The LM statistic can then be written as

$$LM = \frac{1}{2}\left\{\sum_{t=!}^{T}\left(\frac{\varepsilon_t^2}{\hat{\sigma}^2} - 1\right)\tilde{\mathbf{x}}_t\right\}' \left\{\sum_{t=1}^{T} \tilde{\mathbf{x}}_t \tilde{\mathbf{x}}_t'\right\}^{-1} \left\{\sum_{t=!}^{T}\left(\frac{\varepsilon_t^2}{\hat{\sigma}^2} - 1\right)\tilde{\mathbf{x}}_t\right\} \tag{5.60}$$

where $\tilde{\mathbf{x}}_t = [1, \mathbf{x}_t]'$. For details, see [63].

Again, the test can be carried out in stages as follows:

1. Estimate model (5.1) assuming homoscedasticity and compute the residuals $\hat{\varepsilon}_t$. Orthogonalize the residuals by regressing them on $\nabla G(\mathbf{x}_t; \hat{\psi})$ and as before compute the $SSR_0 = \frac{1}{T}\sum_{t=1}^{T}\left(\frac{\tilde{\varepsilon}_t^2}{\hat{\sigma}^2_{\tilde{\varepsilon}_t}} - 1\right)^2$, where $\hat{\sigma}^2$ is the unconditional variance of $\tilde{\varepsilon}_t$.

2. Regress $\left(\frac{\tilde{\varepsilon}_t^2}{\hat{\sigma}^2_{\tilde{\varepsilon}_t}} - 1\right)$ on $\tilde{\mathbf{x}}_t$ and compute the residual sum of squares $SSR_1 = \frac{1}{T}\sum_{t=1}^{T} \tilde{v}_t^2$.

3. Compute the χ^2 statistic

$$\mathrm{LM}^{\sigma}_{\chi^2} = T\frac{SSR_0 - SSR_1}{SSR_0}$$

or the F version of the test

$$\mathrm{LM}^{\sigma}_{F} = \frac{(SSR_0 - SSR_1)}{s}\left(\frac{SSR_1}{(T-s-n)}\right)^{-1}.$$

Where T is the number of observations. Under H_0, $\mathrm{LM}^{\sigma}_{\chi^2}$ is asymptotically distributed as a χ^2 with s degrees of freedom and LM^{σ}_{F} has approximately an F distribution with s and $T-s-n$ degrees of freedom.

5.6.3. Test of parameter constancy

When testing for parameter constancy, several approaches can be taken. Many available tests are tests against unspecified alternatives or single structural breaks. Following [63], we present a parametric alternative to parameter constancy which allows the parameters to change smoothly as a function of time under the alternative hypothesis.

In the following we will assume that the membership function has constant parameters whereas $\mathbf{b}_i,\ i = 1,...,r$ may be subject to changes over time. That is, we test against varying consequents. To develop the test we rewrite model (5.1) as

$$y_t = \tilde{\mathrm{G}}\left(\mathbf{x}_t;\boldsymbol{\psi},\tilde{\boldsymbol{\psi}}\right) + \varepsilon_t = \sum_{i=1}^{r} \tilde{\mathbf{b}}_i(t)\mathbf{x}_t \cdot \mu_i\left(\mathbf{x}_t;\boldsymbol{\psi}_{\mu_i}\right) + \varepsilon_t, \tag{5.61}$$

where

$$\tilde{\mathbf{b}}_i(t) = \mathbf{b}_i + \sum_{j=1}^{B} \check{\mathbf{b}}_{ij} f_j\left(t;\zeta_j,\eta_j\right). \tag{5.62}$$

The parameter vectors are defined as $\boldsymbol{\psi} = [\mathbf{b}_1,...,\mathbf{b}_r,\boldsymbol{\psi}_{\mu_1},...,\boldsymbol{\psi}_{\mu_r}]'$ and $\tilde{\boldsymbol{\psi}} = [\check{\mathbf{b}}_{11},...,\check{\mathbf{b}}_{1B},\check{\mathbf{b}}_{r1},...,\check{\mathbf{b}}_{rB},\zeta_1,...,\zeta_B,\eta_1,...,\eta_B]'$, where each pair (ζ_j,η_j) refers to the parameters of an unidimensional logistic function on t.

In order to guarantee the identifiability of the model we impose two aditional restrictions: $\eta_1 \leq \eta_2 \leq ... \leq \eta_B$ and $\zeta_j > 0, j = 1,...,B$. The parameters ζ_j are responsible for the smoothness of the changing in the autoregressive parameters. When $\zeta_j \to \infty$, equation (5.62) represents a model with B structural breaks. Substituting (5.62) in (5.61), we have

$$y_t = \sum_{i=1}^{r} \left\{ \mathbf{b}_i + \sum_{j=1}^{B} \check{\mathbf{b}}_{ij} f_j \left(t; \zeta_j, \eta_j\right) \right\} \mathbf{x}_t \cdot \mu_i \left(\mathbf{x}_t; \boldsymbol{\psi}_{\mu_i}\right) + \varepsilon_t. \tag{5.63}$$

Now let us consider $B = 1$ and rewrite (5.63) as

$$y_t = \sum_{i=1}^{r} \left\{ \mathbf{b}_i + \check{\mathbf{b}}_i f \left(t; \zeta, \eta\right) \right\} \mathbf{x}_t \cdot \mu_i \left(\mathbf{x}_t; \boldsymbol{\psi}_{\mu_i}\right) + \varepsilon_t. \tag{5.64}$$

The null hypothesis of parameter constancy is $\mathrm{H}_0 : \zeta = 0$. Note that model (5.64) is only identified under the alternative $\zeta \neq 0$. Of course, as a consequence of this fact, the standard asymptotic distribution theory for the likelihood ratio or other classical test statistics for testing H_0 is not available. As we did before, we will expand $f(\mathbf{x}_t; \zeta, \eta)$ into a first-order Taylor series around $\zeta = 0$:

$$\hat{f} = \frac{1}{4} \zeta (t - \eta) + R(t; \zeta, \eta), \tag{5.65}$$

where $R(t; \zeta, \eta)$ is the remainder. Replacing $f\left(t; \zeta, \eta\right)$ in (5.64) by (5.65), we get

$$y_t = \sum_{i=1}^{r} (\boldsymbol{\theta}_{0i} + \boldsymbol{\theta}_{1i}\ t) \mathbf{x}_t \cdot \mu_i \left(\mathbf{x}_t; \boldsymbol{\psi}_{\mu_i}\right) + \varepsilon_t^* \tag{5.66}$$

where $\boldsymbol{\theta}_{0i} = \mathbf{b}_i - \check{\mathbf{b}}_i \eta / 4$ and $\boldsymbol{\theta}_{1i} = \check{\mathbf{b}}_i / 4$.

The null hypothesis now becomes $\mathrm{H}_0 : \boldsymbol{\theta}_{11} = \boldsymbol{\theta}_{12} = ... = \boldsymbol{\theta}_{1r} = 0$, under which $\varepsilon_t^* = \varepsilon_t$. The local approximation to the normal log-likelihood function in a neighbourhood of H_0 for observation t and ignoring $R(t; \zeta, \eta)$ is

$$l_t \ = \ -\frac{1}{2} \ln(2\pi) - \frac{1}{2} \ln \varsigma^2 - \frac{1}{2\varsigma^2} \left\{ y_t - \sum_{i=1}^{r} (\boldsymbol{\theta}_{0i} + \boldsymbol{\theta}_{1i}\ t) \mathbf{x}_t \mu_i \left(\mathbf{x}_t; \boldsymbol{\psi}_{\mu_i}\right) \right\}^2. \tag{5.67}$$

The *LM* statistic can hence be written as (5.50) with $\hat{\mathbf{h}}_t$ and $\hat{v}_t$ defined as above, and again, the test can be carried out in stages, as follows:

1. Estimate model (5.1) assuming parameter constancy and compute the residuals $\hat{\varepsilon}_t$. When the sample size is small and the model is difficult to estimate, numerical problems in applying the nonlinear least squares algorithm may lead to a solution where the residual vector is not exactly orthogonal to the gradient matrix of the nonlinear function $\mathrm{G}(\mathbf{x}_t;\hat{\boldsymbol{\psi}})$. This has an adverse effect on the empirical size of the test. To solve this problem, we regress the residuals $\hat{\varepsilon}_t$ on $\nabla\mathrm{G}(\mathbf{x}_t;\hat{\boldsymbol{\psi}})$, and compute the residual sum of squares $SSR_0 = (1/T)\sum_{t=1}^{T}\tilde{\varepsilon}_t^2$.

2. Regress $\tilde{\varepsilon}_t$ on $\hat{\mathbf{h}}_t$ and $\hat{v}_t$. Compute the residual sum of squares $SSR_1 = (1/T)\sum_{t=1}^{T}\hat{v}_t^2$.

3. Compute the χ^2 statistic

$$\mathrm{LM}_{\chi^2}^{pc} = T\frac{SSR_0 - SSR_1}{SSR_0}$$

   or the F version of the test

$$\mathrm{LM}_{F}^{pc} = \frac{(SSR_0 - SSR_1)}{m}\left(\frac{SSR_1}{(T-m-n)}\right)^{-1}.$$

Under H_0, $\mathrm{LM}_{\chi^2}^{pc}$ is asymptotically distributed as a χ^2 with m degrees of freedom and LM_{F}^{pc} has approximately an F distribution with m and $T-m-n$ degrees of freedom.

The extension to an arbitrary value of B is straightforward, and it will allow us to test against smoothly changing residuals that move across any number of regimes.

5.7. A hybrid modelling cycle for FRBM

In previous sections of this Chapter we have developed some elements of a new approach to fuzzy rule-based modelling which was originated in the relation of a TSK rule and a linear autoregressive model. This relation allowed us to link FRBM with a group of statistical models, the regime-switching family, and, in turn, to import the ideas of these well established models to this Soft Computing field.

It is now possible to gather all these ideas to propose a full modelling cycle for FRBM in the framework of time series that, opposed to the approaches already found in the literature, integrates, together with the usual Soft Computing methods, the theoretical and practical considerations developed throughout decades by statisticians devoted to time series analysis.

5.7.1. Exploratory analysis

Such a modelling cycle should always start by studying, in a classical statistical manner, the properties of the series under study. Mean, variance, histogram, autocorrelogram, phase diagram and so on are important indicators whose information should never be left aside. Then, seasonal factors and trends should be identified and, in some cases, removed. If the series is not stationary, it is important to consider differencing at least once. Normalisation is also to be considered, as most models have this as a pre-requisite.

It can not be stressed enough how important this first step is, as the best modelling and estimation techniques will inevitably fail if the series is not properly studied and prepared for them. Also, from the study of the autocorrelation function (ACF) and the partial autocorrelation function (PACF) we obtain valuable information for identifying the structure of the model, including the variables used in the consequents and those used in the membership functions. Neglecting this information can only decrease the chances of obtaining a good model.

5.7.2. Variable selection

Once we have a knowledge of the statistical characteristics of the series, we should make some fundamental decisions: how many past values are to be considered by the model? And which, of those values, are to be considered for the transition or membership functions? These two decisions are critical and will determine the failure or success of the modelling process.

Several approaches are available for this task, from trial and error to the latests developments in feature selection. In the statistical field, it is usual to linearize the model and to apply linear variable selection techniques. In [73], the stationary nonlinear model is approximated by a polynomial of sufficiently high order, and then different combinations of variables are tried and evaluated using a model selection criterion like the Akaike Information Criterion (AIC)

or the Bayesian Information Criterion (BIC). Another possibility is to test for linearity on the models built with each combination and choose the one that yielded a smaller p-value in the test.

5.7.3. Linearity testing

When the variable structure is defined, we need to know if the series should actually be modelled by a FRBM that uses such set of variables or if its behaviour is already properly explained with a linear model. It is now when we should apply the linearity tests developed in section 5.3. If the null hypothesis is rejected, we proceed to the next step, else we consider our modelling procedure finished.

5.7.4. Determining the number of rules

Discovering how many of rules are enough to obtain a good model is the next thing to be accomplished. We apply hypothesis testing in an iterative manner, up to the first acceptance of the null hypothesis, as explained in section 5.5. Throughout this procedure, estimation is done through log-likelihood, to guarantee consistency of the estimates of the parameters of the $r+1$ rule. The process is started with a significance level of 0.95 which is halved on each iteration, aiming to parsimonious models. This is a conservative approach followed by [64].

Important care must be taken in choosing the initial values for the optimisation algorithm. In [64], $K = 1000$ sets of the model's parameters are drawn and the resulting K models are compared in terms of log-likelihood[1] . The model which maximises the log-likelihood is then chosen.

5.7.5. Tuning

Once we know how many rules are sufficient to properly model the series, we can consider that the model identification phase is finished and we can start with the model estimation part. Although the model resulting from the iterative procedure above is already estimated, we could now consider applying some of

[1] It is questionable if 1000 initial values is not too big a number, and experience tells us that, even if no optimisation algorithm is used, such a number of initial values is enough as to find already very good candidate points. In order to assess the actual performance of the algorithm, in Chapter 6 we reduced K to just 25 points.

the Soft Computing ideas to fine-tune the parameters. For example, optimizing them through meta-heuristics as a genetic algorithms or simulated annealing could result in much more accurate estimates, even if not much could be said about its consistency.

5.7.6. Model evaluation

At this point the model is identified and estimated. We can now turn to its application to the series in order to evaluate it, in the last step of this procedure. As was explained in section 5.6, it is fundamental to study the behaviour in time of the residuals and the model parameters. Applying the tests developed in that section should be mandatory prior to considering this modelling cycle finished. If there is no rejection of the null hypothesis, we have obtained a satisfying model that captures the behaviour of the series and that can be used to forecast its future values.

If the null hypothesis is rejected in any of the tests, we should reconsider some of the decisions taken throughout the modelling cycle, as the variable selection or even the choice of the model itself. It is important to remember that, by using FRBM we are making some assumptions about the system to be modelled (its regime-switching nature, for example), and that it could easily be the case that these assumptions were not realistic. In that case, it might be a much better idea to try a different model (even in spite of the *universal approximation* property of FRBM).

5.7.7. Modelling cycle

Next we present an algorithmic summary of the proposed modelling cycle. It must be noted that these steps are not to be followed blindly but that the experience and insight of the practitioner must guide the process and adapt it to the real situation that he or she is facing.

5.8. Discussion

In this Chapter, we have developed some consequences of the results presented in Chapter 4. In section 5.2, asymptotic stationarity for the FRBM was de-

veloped and the conditions for its identifiability were established. Both results contribute to a better insight of the model, and are basic elements for generating a formal statistical ground which allow for a new modelling approach to FRBM.

Another important element of this approach is the linearity test presented in section 5.3. This test allows the practitioner to know if the series under study is linear or if it could be better modelled with a FRBM. Modelling with a FRBM a series whose data generating process (DGP) is linear is undesirable because it constitutes a waste of resources (and time).

But if a series is proven to be nonlinear and we want to model it using a FRBM, it is crucial to have means to identify it. One of the most important steps in this identification process is determining how many rules are sufficient to capture the complexity of the series. Section 5.5 presents an iterative procedure, based on hypothesis testing, to fix the required number of rules that would be enough to properly model a given series. This procedure results in parsimonious models and uses the log-likelihood estimator to end up with an already estimated model wich contains the proper number of rules.

As estimation is necessary for identifying the required number of rules, section 5.4 studies the properties of the maximum likelihood estimator of the FRBM. Existence, consistence and asymptotic normality of this estimator are derived, as another step towards formally establishing the statistical ground of these models.

Finally, a model should never be accepted without an evaluation phase that should take place once the model is identified and estimated. In section 5.6, we presented three misspecification tests that are useful to know if the model is effectively capturing the behaviour of the series. In FRBM, it is now possible to check the independence and homoscedasticity of the residuals, together with parameter constancy, through hypothesis tests, and this contributes to a better understanding and use of the model.

Putting together all these pieces results in a new hybrid modelling cycle for FRBM, proposed in section 5.7. This procedure combines the statistical results presented above with some well-known tools coming from the Soft Computing field, summing up the advantages of both.

Algorithm 2 Hybrid modelling cycle for FRBM.

[1.] Study the statistical properties of the series: mean, variance, skewness, kurtosis, autocorrelation function (ACF), partial autocorrelation function (PACF).
if necessary **then**
  Identify (and remove) seasonal factor and trend.
  Perform other transformations to the series (normalization, ...).
end if
[2.] Define the structure of the model (input variables, membership variables).
[3.] Test for linearity ($H_0 : \gamma = 0$).
if H_0 is accepted **then**
  The series is linear: modelling cycle is finished.
else
  while H_0 is rejected **do**
    [4.] Add a new rule.
    Determine a good set of initial parameters.
    Estimate the model's parameters through log-likelihood.
    Test for the addition of another rule ($H_0 : \gamma = 0$).
  end while
end if
[5.] Re-estimate the model's parameters using a meta-heuristic (SANN, GA, GAD).
[6.] Apply misspecification tests (parameter constancy, serial independence and homoscedasticity of the residuals).
if H_0 is rejected in any misspecification test **then**
  Consider starting all over again, or using a different model.
else
  The model is properly built: modelling cycle is finished.
end if

6. Experiments and Applications

In this chapter we turn to the practical implications of the theoretical results developed in Chapters 4 and 5. We will apply the statistical approach to building Fuzzy Rule-based Models, including automatically determining the number of rules and the model evaluation tests. We will also apply a set of estimation procedures to the models, including metaheuristics as genetic algorithms and simulated annealing. The results of the experiments will be discussed, and we will obtain an empirical evidence of the benefits of this work.

6.1. Motivation

As we know, when two different scientific disciplines are linked, there is an immediate transfer of knowledge from one to another. In our case, the equivalence results shown in Chapter 4 are entailing us to transfer methods and techniques for time series analysis from a class of statistical models to a class of Soft Computing models, and vice versa.

The contributions included in Chapter 5 resulted in a new proposal for a modelling cycle for FRBM in the time series context. This cycle combines ideas from the statistical regime-switching paradigm with the latest developments in the area of Soft Computing.

It is now the time to practically test and apply this modelling cycle, providing evidences of the empirical validity of all the theoretical developments unfolded above. In order to do so, we will perform two different types of experiments.

On the one hand, we will generate a set of artificial (synthetic) time series, with known behaviour and properties, and we will apply the modelling cycle to them. The use of synthetic datasets will allow us to evaluate the actual modelling capabilities of our proposal, as we will argue below.

On the other hand, two real-world time series will be studied and modelled. One of them is a well-known series, deeply studied in the literature: the so-called "lynx" series. The other two are original series coming from fellow researchers in different disciplines and pose the greatest challenge for our proposal, as their chaotic component seems to be big.

6.2. Montecarlo experiments

The use of synthetic datasets has been recently studied in the framework of Soft Computing. For a detailed state-of-the-art, see [5]. Nonetheless, in the statistical field, it is a common practice to use this type of experiments to check the modelling capabilities of the proposals.

The basic assumption is that any series is considered to be generated by a usually unknown data generating process (DGP) to which a noise component is added (see equation (5.44), page 88). As a reverse result of this, to generate an artificial time series, we need to define a DGP and a noise distribution, whose sum in iterative application will produce the data. This artificial series could then be studied under the chosen modelling scheme, identifying and estimating a model for it. If the parameters of this model are (or tend to be) equal to the parameters of the original DGP, we obtain a clear evidence that the modelling scheme is correct.

In order to simulate a series according to the aforementioned basic assumption, we must go back again to the expression of the general model, equation (5.44). The first part of the right hand side of that expression is called in this context the *model skeleton*, and of course is the part which is to be modelled. Having defined a model skeleton or DGP, we generate the series by seeding a random starting point $\mathbf{x}_{t_0}$ and successively obtaining the y_t, $t = 1...T$, by applying the skeleton function and adding a n.i.d. value given by the random series ε_t. It is usually a good idea to discard the first N observations to avoid initialisation effects.

In this study, we generated six synthetic time series. The first five of them are the ones used by [64], that we reuse in order to test them in the FRBM framework. The sixth one is similar, but it only has a higher number of rules.

Except for Experiment 1, which dealt only with the linearity test, the rest of the experiments consisted in the application of the steps **3** to **6** of Algorithm 2.

Steps **1** and **2** were not considered as the variable structure of the models was known beforehand. All the experiments were run on 500 replications of each model.

Hence, we first applied the iterative testing procedure to determine the number of rules needed by the model, and obtained a first estimation of the parameters. Then, we applied three meta-heuristics to the estimated models to try and further optimise their parameters. These meta-heuristics were simulated annealing, a standard genetic algorithm and a genetic algorithm combined with a least squares method as BFGS. We describe them below.

Simulated Annealing

Simulated annealing (SA) is a generic probabilistic meta-heuristic for the global optimisation problem, namely locating a good approximation to the global optimum of a given function in a large search space. For certain problems, simulated annealing may be more effective than exhaustive enumeration — provided that the goal is merely to find an acceptably good solution in a fixed amount of time, rather than the best possible solution.

The name and inspiration come from annealing in metallurgy, a technique involving heating and controlled cooling of a material to increase the size of its crystals and reduce their defects. The heat causes the atoms to become unstuck from their initial positions (a local minimum of the internal energy) and wander randomly through states of higher energy; the slow cooling gives them more chances of finding configurations with lower internal energy than the initial one.

By analogy with this physical process, each step of the SA algorithm replaces the current solution by a random "nearby" solution, chosen with a probability that depends on the difference between the corresponding function values and on a global parameter T (called temperature), that is gradually decreased during the process. The dependency is such that the current solution changes almost randomly when T is large, but increasingly "downhill" as T goes to zero. The allowance for "uphill" moves saves the method from becoming stuck at local minima —which are the bane of greedier methods.

The method was independently described by [47] and [15]. The method is an adaptation of the Metropolis-Hastings algorithm, a Monte Carlo method to generate sample states of a thermodynamic system, invented by N. Metropolis

et al. in 1953 [65].

Genetic Algorithms

Genetic algorithms (GA) are a class of optimisation algorithms that use techniques inspired in biological evolution to solve optimisation problems. Similarly to Monte Carlo optimisation they use a stochastic approach, but the encoding of the optimisation parameters in a DNA-like fashion allows them to exchange information between different models under consideration. This usually increases the convergence rate, resulting in less objective function evaluations.

The usual flow of a genetic algorithm consists of several steps, which can be performed in a number of ways, depending on the requirements of the user and the optimisation problem at hand.

The first and very important step is to chose the way the optimisation parameters are encoded. In a lot of cases they are represented by a string of 1s and 0s, (*binary encoding*) similar to the usual representation of integers but with a user controllable range and discretisation step for each parameter. All operations of the GA are performed on this representation, giving an abstraction from the individual parameters. Real encoding is also allowed, and it was used in this work.

After a representation has been chosen, the algorithm randomly generates a number of DNA strings of the required length, the so called population of a chosen size N. This population is the basis for the rest of the algorithm. For each member of the population an objective function value is calculated by transforming their DNA string back to the optimisation parameters and evaluating the objective function with those parameters. After all objective function evaluations have been performed, these values are transformed into a probability. Population members with a low objective function value (for a minimisation problem) receive high probabilities and vice versa. Based on this probability members are selected for the next iteration (or generation).

At the beginning of each generation, a new population of the same size N is created by choosing members from the last generation according to the calculated probability. Duplicates of members are possible. If this was the only step in the iteration there would be no innovation. Therefore two steps are introduced to create new members.

Crossover is a systematic exchange of information. With a chosen probability

two models exchange their DNA after a chosen point. This creates new members that have some similarity with the old population members. The idea is that combining characteristics of two good models has a good chance of yielding an even better model.

Mutation on the other hand is completely unsystematic, but can introduce new members into the population. With a chosen probability, one of the bits in the parameter representation changes its value. The mutation probability is usually chosen fairly low, so that only about 10 percent of the population undergo mutation.

Genetic Algorithm with Derivatives

When a statistical model's estimating function (for example, a log-likelihood) is nonlinear in the model's parameters, the function to be optimised will generally not be globally concave and may have irregularities such as saddlepoints or discontinuities. Optimisation methods that rely on derivatives of the objective function may be unable to find any optimum at all. Multiple local optima may exist, so that there is no guarantee that a derivative-based method will converge to the global optimum. On the other hand, algorithms that do not use derivative information (such as pure genetic algorithms) are for many problems needlessly poor at local hill climbing. Most statistical problems are regular in a neighbourhood of the solution. Therefore, for some portion of the search space, derivative information is useful.

Method BFGS [74, 32, 28, 9] is a quasi-Newton method (also known as a variable metric algorithm), that was published simultaneously in 1970 by Broyden, Fletcher, Goldfarb and Shanno. It uses function values and gradients to build up a picture of the surface to be optimised.

Restrictions to the parameters In both types of genetic algorithms it was necessary to fix some restrictions to the search space of the parameters. The parameters measured in the dimension of the series, i.e., the threshold parameters, were fixed to have a range between the minimum and the maximum of the series plus the standard deviation of it. The γ parameters ranged between zero and a hundred. In the case of the NCSTAR, the ω parameters were fixed between -1 and 1, and the unitary norm condition was also enforced. As well,

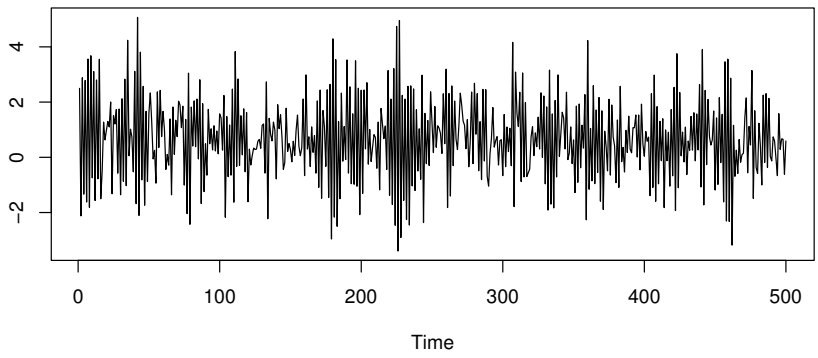

Figure 6.1.: Generated series for Experiment 1, stationary linear autoregressive model.

the lexicographical ordering of the rules mentioned in Restriction (R. 1), Section 5.2.2, was imposed over each new set of parameters.

6.2.1. Experiment 1

We start by simulating a stationary linear autoregressive model:

$$y_t = 0.8 - 0.5y_{t-1} + 0.3y_{t-2} + \varepsilon_t, \qquad \varepsilon_t \sim \text{NID}(0, 1^2), \tag{6.1}$$

whose simple formulation produces a series as the one shown in Figure 6.1.

Knowing that this series is linear, we first wanted to check if the null hypothesis of the linearity test would be accepted or not. By using the skeleton and the random noise series, we simulated 500 replications of the model and applied the test to them. The results were conclusive: over the 500 series, the null hypothesis was accepted in 95.2% of the cases. There were just 24 series where the test failed.

Then, to compare this result with standard FRBM modelling, suppose that, when faced to this series, we decide to apply ANFIS (described in Section 2.4.1,

page 26), in its standard grid partitioning style. If 3 labels were assigned to each of the two input variables, y_{t-1} and y_{t-2}, the model would have 9 rules and a total of 39 parameters (9×3 linear parameters and 6×2 nonlinear parameters).

To make the comparison more fair, we also tried to model the series with an ANFIS using, instead of grid partition, the substractive clustering method with default parameters to determine the number of rules. In this case, the model ended up with just 3 fuzzy rules (also fixing 3 linguistic labels per input), counting a total of 21 parameters (3×3 linear and 6×2 nonlinear).

Comparing the complexity of this ANFIS model with the DGP, which has a total of 3 parameters, the importance of the linearity tests becomes evident.

6.2.2. Experiment 2

The second simulated model is similar to the specification studied by [83], and is a two regime STAR model with two extreme regimes:

$$\begin{aligned} y_t = 1.8y_{t-1} - 1.06y_{t-2} + \\ (0.02 - 0.9y_{t-1} + 0.795y_{t-2}) \times \mu_S(\mathbf{x}_t;\boldsymbol{\psi}) + \varepsilon_t, \\ \varepsilon_t \sim NID(0, 0.02^2) \end{aligned} \quad (6.2)$$

where the nonlinear parameters are $\boldsymbol{\psi} = [\gamma, \boldsymbol{\omega}, c] = [20, (1,0), 0.02]$.

The first regime of this model, corresponding to $\mu_S(\mathbf{x}_t;\boldsymbol{\psi}) = 0$ is explosive, while the other regime, determined by $\mu_S(\mathbf{x}_t;\boldsymbol{\psi}) = 1$, is not. For the long term behaviour, the model has a unique stable stationary point, $y_\infty = 0.036$. A realisation of the series generated by this model is shown in Figure 6.2.

We applied the linearity test to this series, obtaining a 98.3% of correct rejections of the null hypothesis. Then, assuming the alternative hypothesis to be true, we applied the number of rules determination procedure, and obtained the correct number of rules in the 97.7% of the cases. Over the 500 replications, only 6 were determined to have more than 2 rules.

Then we turned our attention to estimating. Once the number of rules of the model was fixed, and its parameters estimated, we applied a set of metaheuristics to optimise them. Table 6.1 show the results of this double optimisation.

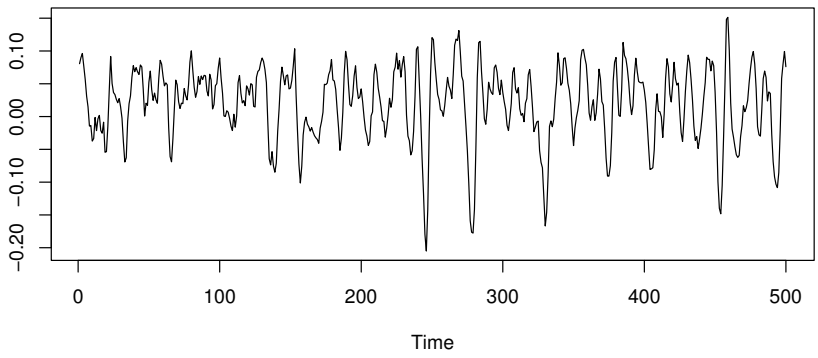

Figure 6.2.: Generated series for Experiment 2, smooth transition autoregressive model.

In this table[1], only the parameters related to the membership function are shown. The rest of the parameters, the linear parameters, are not actually es-

[1]The next four paragraphs are applicable to tables 6.2, 6.3, 6.4 and 6.5, so they are ommited in the following sections.

Table 6.1.: Estimation results for Experiment 2.

		Algorithm			
Parameter	Value	BFGS	SA	GA	GAD
γ	20.0	22.0901 (22.2755)	33.2197 (3.9459)	12.3945 (5.4472)	11.0453 (3.9534)
c	0.02	0.0197 (0.0462)	0.0165 (0.0279)	0.0250 (0.1494)	0.0279 (0.2459)
ω_1	1.0	0.9763 (0.0337)	0.9950 (0.0061)	0.9802 (0.0280)	0.9638 (0.0524)
ω_2	0.0	-0.0116 (0.3177)	-0.0996 (0.0942)	-0.0835 (0.3055)	-0.0796 (0.3997)

timated but analytically computed once the nonlinear parameters are fixed, in what is called concentrated least squares method. For the sake of clarity, they are removed from the tables.

The first numeric column of this table, labelled Value, contains the actual values of the parameters, i.e. those used when generating the 500 instances of the series. The column labelled BFGS shows the median of the result of applying step **[4.]** of the modelling cycle to each of the 500 series, that is, the median of the parameters estimated during the incremental rule addition. The next three columns show the median results of the application of the three metaheuristics explained in sections 6.2, 6.2 and 6.2 to the model, using as starting values the values contained in column BFGS.

Below each estimated value, in parenthesis, we show the median absolute deviation, MAD, which corresponds to the median of the absolute value of the deviations from the data's median. It is computed as

$$\text{MAD} = \underset{i}{\text{median}}(\|X_i - \underset{j}{\text{median}}\, X_j\|). \tag{6.3}$$

In order to study these tables, we must take into account that the parameters are not scale-free, and that the γ parameter is usually much larger than the rest. Hence the errors obtained on each parameter must be considered relatively to the scale of it.

If we finally turn to Table 6.1, the data contained in it shows interesting results. As we can see, the values obtained using just the BFGS algorithm are quite close to the real values. When the metaheuristics are applied over this solution, they find a range of alternative solutions placed in local minima not too far away from the true solution. As we can see, the main differences are found in the γ parameter. Notwithstanding, the MAD shows that the estimation through metaheuristics seems to produce more consistent estimates of γ, as its value is several times bigger in the case of the BFGS approach.

It is worth noting that, in all the algorithms, the MAD of the ω_2 is also quite big if compared against the same measure for parameter ω_1. This indicates that they find it difficult to estimate a null parameter, and that removing it from the model could be a good idea.

We must note that this problem is small in the number of parameters, and we could not expect a big improvement from the metaheuristics over an already very good solution. Notwithstanding, the MAD indicates that, in average, the

solutions found by the combination of BFGS and metaheuristics are more consistent than those found by just the BFGS.

6.2.3. Experiment 3

The third simulated model corresponds to a three regime STAR model:

$$\begin{aligned} y_t = -0.1 + 0.3y_{t-1} + 0.2y_{t-2} + \\ (-1.2y_{t-1} + 0.5y_{t-2}) \times \mu_S(\mathbf{x}_t; \boldsymbol{\psi}_1) + \\ (1.8y_{t-1} - 1.2y_{t-2}) \times \mu_S(\mathbf{x}_t; \boldsymbol{\psi}_2) + \varepsilon_t, \\ \varepsilon_t \sim NID(0, 0.5^2) \end{aligned} \quad (6.4)$$

where the nonlinear parameters are $\boldsymbol{\psi}_1 = [\gamma_1, \boldsymbol{\omega}_1, c_1] = [20, (1,0), -0.6]$ and $\boldsymbol{\psi}_2 = [\gamma_2, \boldsymbol{\omega}_2, c_2] = [20, (1,0), 0.6]$.

This model has three limiting regimes, of which the "lower" one corresponds to $\mu_S(\mathbf{x}_t; \boldsymbol{\psi}_1) = \mu_S(\mathbf{x}_t; \boldsymbol{\psi}_2) = 0$ and is stationary, the "middle" regime has $\mu_S(\mathbf{x}_t; \boldsymbol{\psi}_1) = 1$ and $\mu_S(\mathbf{x}_t; \boldsymbol{\psi}_2) = 0$ and is explosive, while the "upper" regime is determined by $\mu_S(\mathbf{x}_t; \boldsymbol{\psi}_1) = \mu_S(\mathbf{x}_t; \boldsymbol{\psi}_2) = 1$ and is also explosive. Concerning its long term behaviour, the model has a limit cycle with a period of 8 time units. A realisation of the series generated by this model is shown in Figure 6.3.

In this case, the linearity test determined the nonlinearity of the series in the 100% of the cases. On the other hand, the incremental building procedure fixed the correct number of regimes in 90% of the cases, adding extra rules in 49 of the 500 models.

Table 6.2 shows the estimation results. The experimental design was identical to the one used in Experiment 2, but in this case the table shows significant improvements from the metaheuristics, as we shall see.

The BFGS algorithm in this case fails to properly find the nonlinear parameters corresponding to the first nonlinear regime, γ_1 and c_1. Hence, in this problem there is room for improvement for the metaheuristics, which indeed manage to find the actual values of the parameters.

The SANN finds good estimates, but their MAD is quite high compared to the MAD of the genetic algorithms. These manage to properly solve the problem, obtaining relatively low values of the MAD and hence justifying their inclusion in the proposed modelling cycle.

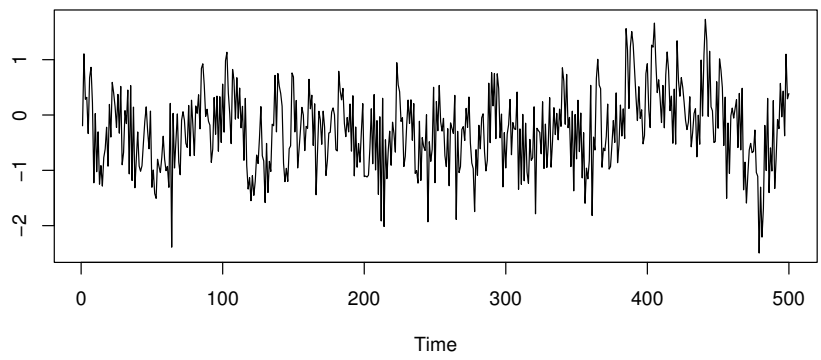

Figure 6.3.: Generated series for Experiment 3, three regime STAR.

The higher number of parameters, together with the characteristic shape of the solution's space makes this problem difficult to solve with the standard hill-climbing approach. Hence, as opposed to the previous experiment, the advantages of using a metaheuristic in a second estimation phase are evident.

6.2.4. Experiment 4

The model simulated in this fourth experiment is a two regime NCSTAR:

$$y_t = 0.5 + 0.8y_{t-1} - 0.2y_{t-2} + (-0.5 - 1.2y_{t-1} + 0.8y_{t-2}) \times \mu_S(\mathbf{x}_t;\boldsymbol{\psi}) + \varepsilon_t$$
$$\varepsilon_t \sim \text{NID}(0, 0.5^2) \quad (6.5)$$

with $\boldsymbol{\psi} = [\gamma, \boldsymbol{\omega}, c] = [11.31, (0.7071, -0.7071), 0.1414]$.

This model has two extreme regimes, given by $\mu_S(\mathbf{x}_t;\boldsymbol{\psi}) = 0$ which is a stationary regime and $\mu_S(\mathbf{x}_t;\boldsymbol{\psi}) = 1$ which has a unit root and hence is non stationary. Still, for its long term behaviour, the process has two stable stationary points, 0.38 and -0.05. A realisation of the series generated by this model is shown in Figure 6.4.

In this case, the linearity test worked in the totality of the 500 runs of the experiment and the iterative building strategy found the proper number of rules in the 98.2% of the cases. Only 9 series were set to be modelled with more than 2 regimes, proving the effectiveness of the testing procedure.

If we turn to the estimation process, we find ourselves in a situation which is close to the one studied in Experiment 2. This is a small problem and it is already properly solved through the hill-climbing approach using the BFGS algorithm in the incremental building and estimation phase.

Indeed, there are no significant differences amongst the median results of the BFGS algorithm and the metaheuristics. However, again, the MAD helps us to realise that the results of the BFGS are of a worst quality, as they show several times more dispersion than the one obtained with the metaheuristics.

Table 6.2.: Estimation results for Experiment 3.

		Algorithm			
Parameter	Value	BFGS	SANN	GA	GAD
γ_1	20.0	10.8616 (10.9488)	20.6136 (19.4400)	21.4366 (13.3229)	21.1886 (14.4303)
γ_2	20.0	17.0280 (18.2518)	28.4699 (12.9456)	20.3515 (7.7956)	21.3101 (9.8493)
c_1	-0.6	-0.2536 (0.5531)	-0.5770 (0.0731)	-0.5948 (0.0520)	-0.5973 (0.0540)
c_2	0.6	0.5819 (0.0703)	0.6001 (0.0355)	0.5972 (0.0365)	0.5981 (0.0361)
ω_{11}	1.0	0.9970 (0.0043)	0.9962 (0.0041)	0.9985 (0.0020)	0.9985 (0.0021)
ω_{12}	0.0	-0.0328 (0.1202)	-0.0864 (0.1215)	-0.0009 (0.0797)	-0.0033 (0.0789)
ω_{21}	1.0	0.9992 (0.0012)	0.9999 (0.0006)	0.9995 (0.0006)	0.9996 (0.0006)
ω_{22}	0.0	0.0167 (0.0635)	0.0011 (0.0044)	-0.00008 (0.0440)	0.0008 (0.0435)

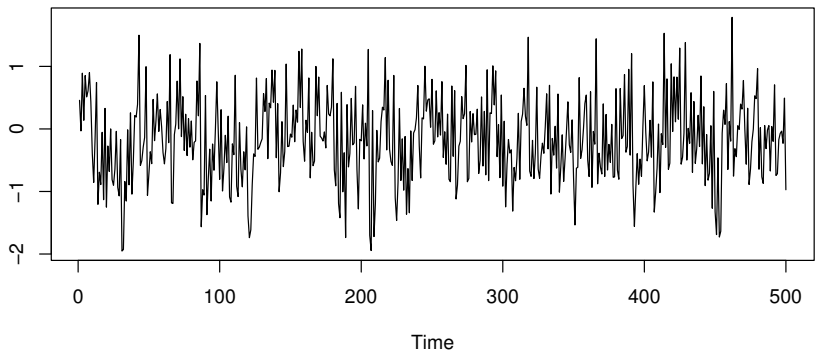

Figure 6.4.: Generated series for Experiment 4, two regime NCSTAR.

Table 6.3.: Estimation results for Experiment 4.

		Algorithm			
Parameter	Value	BFGS	SA	GA	GAD
γ	11.31	12.8377 (7.5582)	12.9615 (1.4807)	12.4234 (2.2625)	13.1617 (7.1550)
c	0.1414	0.1467 (0.0840)	0.1420 (0.0321)	0.1505 (0.0681)	0.1511 (0.0601)
ω_1	0.7071	0.7156 (0.0527)	0.7080 (0.0276)	0.7071 (0.0457)	0.7080 (0.0485)
ω_2	-0.7071	-0.6985 (0.0545)	-0.7062 (0.0271)	-0.7071 (0.0442)	-0.7063 (0.0486)

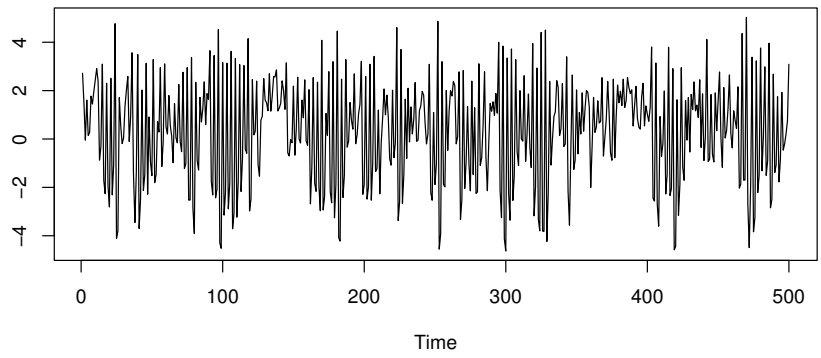

Figure 6.5.: Generated series for Experiment 5, three regime NCSTAR.

6.2.5. Experiment 5

The fifth simulated model is a full three regime NCSTAR, given by

$$\begin{aligned} y_t = 0.5 + 0.8y_{t-1} - 0.2y_{t-2} + \\ (1.5 - 0.6y_{t-1} - 0.3y_{t-2}) \times \mu_S(\mathbf{x}_t; \boldsymbol{\psi}_1) + \\ (-0.5 - 1.2y_{t-1} + 0.7y_{t-2}) \times \mu_S(\mathbf{x}_t; \boldsymbol{\psi}_2) + \varepsilon_t, \\ \varepsilon_t \sim NID(0, 1^2) \end{aligned} \quad (6.6)$$

where $\boldsymbol{\psi}_1 = [\gamma_1, \boldsymbol{\omega}_1, c_1] = [8.49, (0.7071, -0.7071), -1.0607]$ and $\boldsymbol{\psi}_2 = [\gamma_2, \boldsymbol{\omega}_2, c_2] = [8.49, (0.7071, -0.7071), 1.0607]$.

This model has also three limiting regimes: in the "lower" one, $\mu_S(\mathbf{x}_t; \boldsymbol{\psi}_1) = \mu_S(\mathbf{x}_t; \boldsymbol{\psi}_2) = 0$ and it is stationary. The "middle" regime, given by $\mu_S(\mathbf{x}_t; \boldsymbol{\psi}_1) = 1$ and $\mu_S(\mathbf{x}_t; \boldsymbol{\psi}_2) = 0$, is also stable, as well as the "upper" regime, characterised by $\mu_S(\mathbf{x}_t; \boldsymbol{\psi}_1) = \mu_S(\mathbf{x}_t; \boldsymbol{\psi}_2) = 1$. This process has an unique stable stationary point at $y_\infty = 0.99$.

We applied the linearity test to the 500 series of this dataset, and linearity was rejected in 100% of the cases. The results were not so precise when the test was applied iteratively to determine the appropriate number of regimes of the model:

of the 500 series, an 83.2% of the models were set to have 3 regimes, while up to 84 models were built with only 2 regimes. This conservative behaviour contrasts with what happened in the other 3 regime model, in Experiment 3, where the mistaken models had more rules instead of fewer.

As we could expect, the results on this case, shown in Table 6.4, are on the same spirit of those found for Experiment 3, which also had 3 regimes. Notwithstanding, in this case the BFGS algorithm managed to find already good values for most of the parameters, including the usually difficult γ. The room for improvement was much smaller, but still the genetic algorithms managed to improve the results of the hill-climbing method.

The results of the genetic algorithms are significantly better than the rest concerning the thresholds and the ω parameters. Nevertheless, the improvement is more evident if, aside the median results, we compare the MAD obtained by each algorithm. The MAD of the simulated annealing is comparable to the one obtained by the BFGS, but the genetic algorithms are much more precise in all the parameters.

Table 6.4.: Estimation results for Experiment 5.

		Algorithm			
Parameter	Value	BFGS	SA	GA	GAD
γ_1	8.49	9.0818 (8.9122)	9.9746 (9.1548)	9.9746 (1.2291)	11.5150 (8.1389)
γ_2	8.49	8.6295 (4.6815)	5.9998 (4.8874)	7.1576 (1.2893)	9.2428 (4.1550)
c_1	-1.0607	-1.0384 (0.2015)	-1.0728 (0.2512)	-1.0696 (0.1279)	-1.0733 (0.1381)
c_2	1.0607	1.0519 (0.1111)	1.0353 (0.2313)	1.0601 (0.0876)	1.0701 (0.0919)
ω_{11}	0.7071	0.7022 (0.0361)	0.7040 (0.0331)	0.7085 (0.0242)	0.7087 (0.0260)
ω_{12}	-0.7071	-0.7112 (0.0356)	-0.7101 (0.0328)	-0.7052 (0.0243)	-0.7055 (0.0263)
ω_{21}	0.7071	0.7035 (0.0312)	0.7023 (0.0330)	0.7074 (0.0265)	0.7083 (0.0256)
ω_{22}	-0.7071	-0.7080 (0.0300)	-0.7118 (0.0298)	-0.7067 (0.0262)	-0.7060 (0.0255)

6.2.6. Experiment 6

The sixth model is not referenced in the literature, and we built it in order to test the behaviour of the modelling cycle when dealing with more complicated models. It contains five regimes, and is given by

$$\begin{aligned} y_t = 0.5 + 0.8y_{t-1} - 0.2y_{t-2} + \\ (1.5 - 0.6y_{t-1} - 0.3y_{t-2}) \times \mu_S(\mathbf{x}_t; \boldsymbol{\psi}_1) + \\ (0.2 + 0.3y_{t-1} - 0.9y_{t-2}) \times \mu_S(\mathbf{x}_t; \boldsymbol{\psi}_2) + \\ (-1.2 + 0.6y_{t-1} + 0.8y_{t-2}) \times \mu_S(\mathbf{x}_t; \boldsymbol{\psi}_3) + \\ (-0.5 - 1.2y_{t-1} + 0.7y_{t-2}) \times \mu_S(\mathbf{x}_t; \boldsymbol{\psi}_4) + \varepsilon_t, \\ \varepsilon_t \sim NID(0, 0.2^2) \end{aligned} \quad (6.7)$$

where

$$\begin{aligned} \boldsymbol{\psi}_1 &= [\gamma_1, \boldsymbol{\omega}_1, c_1] = [8.49, (0.7071, -0.7071), -1.0607] \\ \boldsymbol{\psi}_2 &= [\gamma_1, \boldsymbol{\omega}_1, c_1] = [8.49, (0.7071, -0.7071), -0.59] \\ \boldsymbol{\psi}_1 &= [\gamma_1, \boldsymbol{\omega}_1, c_1] = [14.23, (0.7071, -0.7071), 0.59] \\ \boldsymbol{\psi}_4 &= [\gamma_2, \boldsymbol{\omega}_2, c_2] = [14.23, (0.7071, -0.7071), 1.0607]. \end{aligned}$$

Testing for linearity, not surprisingly 100% of the series were determined to be nonlinear. The problem arose when determining the number of regimes, as only in 30% of the 500 series the procedure found the correct number of regimes.

Nevertheless, for a model with such a big number of parameters (15 linear plus 16 nonlinear), it is clear that the length of the series (500 points) is insufficient. In order to remove the influence of the length of the series, we created a new set of 500 series with 5000 points each. We repeated the experiment with these longer series, and the results were much better: of the 500 series, 96% were found to have 5 regimes. There were only 21 series that yielded models with an incorrect number of regimes, always higher than 5 except for one, which was fixed to have 4 regimes.

The median estimated values of the parameters of the models created with this new set of series are shown in Table 6.5, under column BFGS. It is clear that, in this case, the estimation was far from obtaining the solution of the problem. The standard BFGS procedure had particular problems determining the

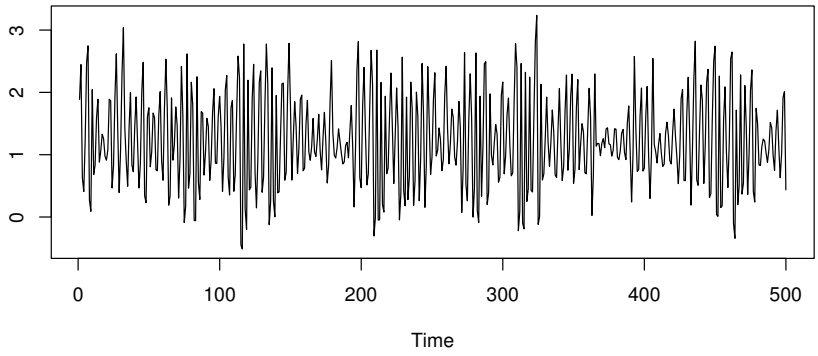

Figure 6.6.: Generated series for Experiment 6, five regime NCSTAR.

last two regimes, whose parameters were fixed to very similar values. Again, we applied the metaheuristics to try to improve the results.

As we can see in the table, the results obtained by the metaheuristics are mixed. On the one hand, the simulated annealing algorithm failed to improve the BFGS results, especially when dealing with the γ and c parameters. This might come from the difficulty of imposing identifiability restrictions to the algorithm, which gets too confused when the number of regimes is higher.

Notwithstanding, the results of the genetic algorithms are quite impressive. Both the GA and the GAD managed to find the actual values of the parameters to a great degree of precision, even in the usually difficult to estimate γ parameter.

For the first time in all the Montecarlo experiments, the GAD algorithm performed better than the standard genetic algorithm, which suggests that the improvement of applying a hill-climbing step is more useful as the problem complexity increases.

Table 6.5.: Estimation results for Experiment 6.

		Algorithm			
Parameter	Value	BFGS	SA	GA	GAD
γ_1	8.49	6.3646 (5.8396)	8.5591 (5.8492)	7.9029 (0.7765)	7.7043 (1.5467)
γ_2	8.49	3.1200 (3.2934)	7.1850 (0.8892)	7.4637 (0.4770)	8.1967 (1.1869)
γ_3	14.23	3.9465 (4.8355)	9.4739 (0.5915)	11.1447 (0.7600)	14.1965 (1.0476)
γ_4	14.23	4.8624 (4.8024)	10.6610 (0.7921)	11.7180 (0.3713)	14.1688 (0.8086)
c_1	-1.0607	-0.8663 (0.5193)	-0.8648 (0.1254)	-1.0322 (0.0617)	-1.0358 (0.0893)
c_2	-0.59	0.6663 (0.2546)	-0.5882 (0.1121)	-0.5571 (0.1046)	-0.6012 (0.0846)
c_3	0.59	0.9183 (0.2842)	0.8245 (0.1048)	0.6021 (0.0191)	0.5911 (0.0182)
c_4	1.0607	0.9195 (0.1117)	1.0607 (0.0699)	1.0477 (0.0169)	1.0599 (0.0173)
ω_{11}	0.7071	0.6955 (0.1320)	0.6973 (0.0422)	0.6948 (0.0306)	0.7009 (0.0301)
ω_{12}	-0.7071	-0.7186 (0.1162)	-0.7168 (0.0749)	-0.7192 (0.0297)	-0.7132 (0.0302)
ω_{21}	0.7071	0.7200 (0.0426)	0.7042 (0.0385)	0.7115 (0.0146)	0.7070 (0.0138)
ω_{22}	-0.7071	-0.6939 (0.0437)	-0.7100 (0.0959)	-0.7027 (0.0146)	-0.7072 (0.0137)
ω_{31}	0.7071	0.7223 (0.0275)	0.7064 (0.0389)	0.7057 (0.0035)	0.7071 (0.0038)
ω_{32}	-0.7071	-0.6916 (0.0281)	-0.7078 (0.0802)	-0.7085 (0.0035)	-0.7071 (0.0038)
ω_{41}	0.7071	0.7249 (0.0446)	0.7285 (0.0271)	0.7065 (0.0062)	0.7073 (0.0057)
ω_{42}	-0.7071	-0.6889 (0.0469)	-0.6851 (0.0377)	-0.7078 (0.0062)	-0.7070 (0.0057)

6.3. Real world problems

Once we have tested the modelling cycle against synthetic data sets, and we have gained an insight of its validity, we can now try it in real situations. We chose to apply it to three existent scientific problems which involve time series: the lynx captures in a period of time in a region of Canada, the calls received in an emergency call centre in the region of Castilla y León and the airborne pollen concentration in the atmosphere of the city of Granada, Spain.

6.3.1. Canadian lynx dataset

The Canadian lynx data set is a commonly used series, corresponding to the annual record of the number of the Canadian lynx "trapped" in the Mackenzie River district of the North-West Canada for the period 1821 to 1934. These data are actually the total fur returns, or total sales, from the London archives of the Hudson's Bay Company in the years of 1821 to 1891 and 1887 to 1913; and those for 1915 to 1934 are from detailed statements supplied by the Company's Fur Trade Department in Winnipeg; those for 1892 to 1896 and 1914 are from a series of returns for the MacKenzie River District; those for the years 1863 to 1927 were supplied by Ch. French, then Fur Trade Commissioner of the Company in Canada. By considering the time lag between the year in which a lynx was trapped and the year in which its fur was sold at auction in London, these data were converted in [24] into the number that were presumably caught in a given year for the years 1821 to 1934 as shown in Figure 6.7(a).

The above time lag was not constant. It depended on the month in which the animal was trapped and the date of shipment. It was also noticed that the catchment area of the animal did not remain constant throughout the period 1821 to 1934.

In 1974, [10] observed that the data on many animal populations in North Canada were periodic with a period of 9.63 years for each cycle. The lynx population was one of them having this cycle. He also found that the tendency to cycle is most pronounced in the Midwest of Canada but this tendency became weaker (and later) as one moved away from this region. The simplest explanation for the cause of the cycle in all species which is acceptable to biologists is their relation, through the food chain, with the corresponding cycle in the snowshoe hare population. For example, the snowshoe hare is a dominant item in the

food of the lynx, coyote, red fox, and fisher.

A first time series model of the Canadian lynx data was fitted by P.A.P Moran in [66]. He observed that the cycle is very asymmetrical with a sharp and large peak and a relatively smooth and small trough. The log transformation gives a series which appears to vary symmetrically about the mean. As the actual population of lynx is not exactly proportional to the number caught, a better representation would perhaps be obtained by incorporating an additional "error of observation" in the model, thereby resulting in a more complicated model. The log transformation substantially reduces the effect of ignoring this error of observation; therefore, after Moran, nearly all the time series analysis of the

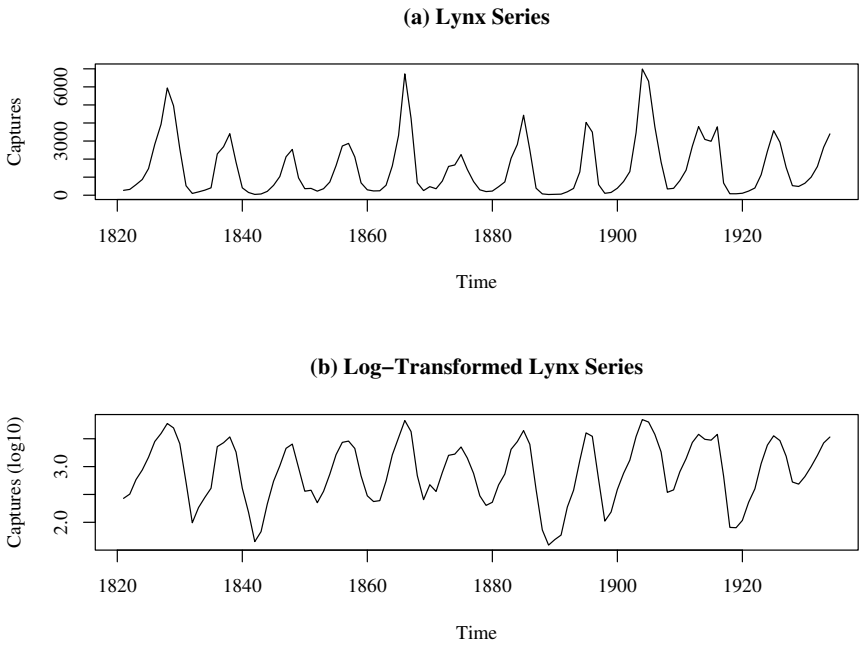

Figure 6.7.: Number of lynx caught in the Mackenzie River district of the North-West Canada from year 1821 to 1934.

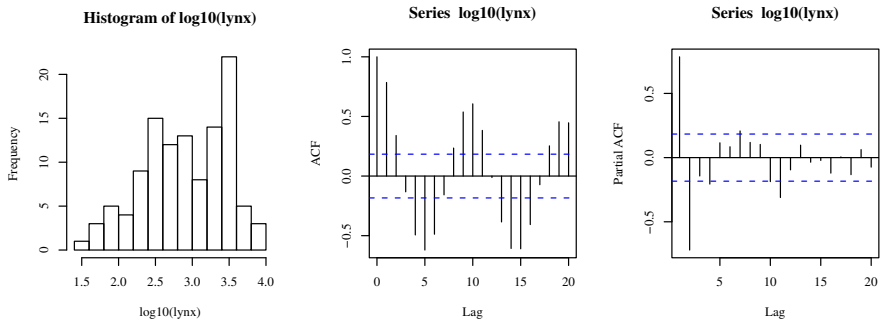

Figure 6.8.: Histogram, autocorrelation and partial autocorrelation functions for the transformed lynx series.

lynx data in the literature have used the log-transformed data. Figure 6.7(b) shows the transformed data.

Figure 6.8 shows the histogram of the transformed series, which shows certain bimodality. We computed its kurtosis and skewness, giving values of -0.773 and -0.357 respectively. Figure 6.8 also shows the ACF and PACF functions. The ACF displays a cyclic behaviour with a period of around 5, while the PACF shows a significant autocorrelation in the first two values. As this transformed dataset is the standard in the literature, we conclude here step 1 of the modelling cycle. Concerning step 2, we will also rely on previous works. The aforementioned study, [66], proposed an AR(2) model considering the sample correlogram, and second order autoregression was also chosen by [11] in a harmonic-autoregressive combined model and by [64] for the NCSTAR model. We fix the order of our model also to 2, for these reasons.

The linearity test against a NCSTAR with sigmoid transition function threw a p-value of 0.000259, while the test against a Gaussian-based NCSTAR obtained a 0.000115. Both tests indicate that the series is nonlinear and suggest the use of the advanced models. Further evidence for this non linearity can be obtained by looking at the nonparametric autoregressions shown in Figure 6.9, where it becomes clear that, while the relation between y_t and y_{t-1} can be well explained using a linear model, this is not the case for the relation between y_t and y_{t-2}.

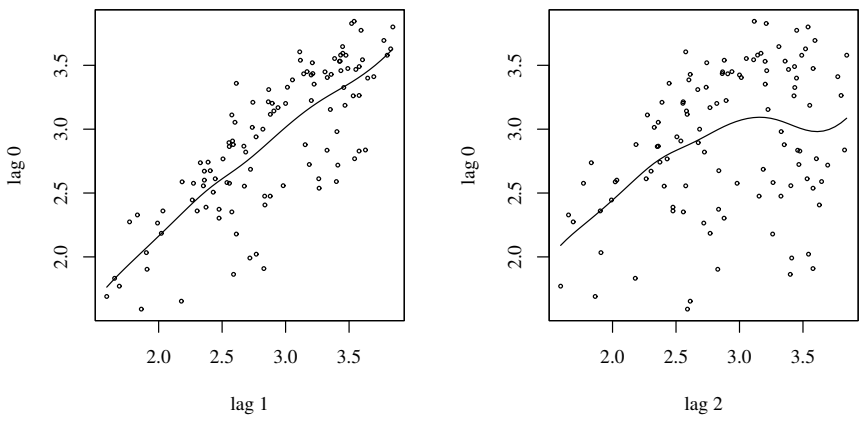

Figure 6.9.: Nonparametric regression function of y_t versus y_{t-1} and of y_t versus y_{t-2}.

The modelling cycle ended in both cases when the second regime was added, so the estimated models have two regimes given by

$$y_t = 0.9599 + 1.2514y_{t-1} - 0.3398y_{t-2} + (2.5466 + 0.3764y_{t-1} - 0.7973y_{t-2})\mu_{\mathsf{S}}(\mathbf{x}_t;\boldsymbol{\psi}_{\mathsf{S}}) + \varepsilon_t \quad (6.8)$$

in the sigmoid case, with $\boldsymbol{\psi}_{\mathsf{S}} = (\gamma, \boldsymbol{\omega}, c) = (103.1266, [0.4630, 0.8863], 9.4274)$, and

$$y_t = 0.8749 + 1.2302y_{t-1} - 0.3074y_{t-2} + (2.0084 + 0.2961y_{t-1} - 0.6486y_{t-2})\mu_{\mathsf{G}}(\mathbf{x}_t;\boldsymbol{\psi}_{\mathsf{G}}) + \varepsilon_t \quad (6.9)$$

in the Gaussian case, where $\boldsymbol{\psi}_{\mathsf{G}} = (\gamma, \mathbf{c}) = (11.0129, [5.8417, 3.6653])$.

The first model obtained a residual standard deviation of $\hat{\sigma}_{\varepsilon,\mathsf{S}} = 0.196$, while the second obtained a value of $\hat{\sigma}_{\varepsilon,\mathsf{G}} = 0.207$. The value obtained for the AIC were $\mathrm{AIC}_{\mathsf{S}} = -314$ and $\mathrm{AIC}_{\mathsf{G}} = -306$ respectively, while the median average percentage error was $\mathrm{MAPE}_{\mathsf{S}} = 5.94\%$ and $\mathrm{MAPE}_{\mathsf{G}} = 6.31\%$.

Once both models were estimated through the standard procedure, we applied a metaheuristic to fine-tune the parameters. After the results obtained in Section 6.2, where the GA tended to obtain the best results, we decided to use just the Genetic Algorithm in this step.

Using the GA to fine tune the parameters, left us with the following two models:

$$y_t = 0.3978 + 1.2560 y_{t-1} - 0.3359 y_{t-2} + (1.0193 + 0.3744 y_{t-1} - 0.7736 y_{t-2}) \mu_{\mathsf{S}}(\mathbf{x}_t; \boldsymbol{\psi}_{\mathsf{S}}) + \varepsilon_t \quad (6.10)$$

in the sigmoid case, with $\boldsymbol{\psi}_{\mathsf{S}} = (\gamma, \boldsymbol{\omega}, c) = (38.9935, [0.4969, 0.8678], 4.1306)$, and

$$y_t = 0.4023 + 1.2224 y_{t-1} - 0.3103 y_{t-2} + (0.8099 + 0.3751 y_{t-1} - 0.7074 y_{t-2}) \mu_{\mathsf{G}}(\mathbf{x}_t; \boldsymbol{\psi}_{\mathsf{G}}) + \varepsilon_t \quad (6.11)$$

in the Gaussian case, where $\boldsymbol{\psi}_{\mathsf{G}} = (\gamma, \mathbf{c}) = (10.000, [2.576, 6.831])$. For these models tuned with the GA, the obtained residual standard deviation was $\hat{\sigma}_\varepsilon = 0.191$ for the sigmoid and $\hat{\sigma}_\varepsilon = 0.205$ for the Gaussian membership function. The value obtained for the AIC were $\text{AIC}_{\mathsf{S}} = -313$ and $\text{AIC}_{\mathsf{G}} = -307$ respectively, while the median average percentage error was $\text{MAPE}_{\mathsf{S}} = 5.90\%$ and $\text{MAPE}_{\mathsf{G}} = 6.26\%$.

At this point, we consider our models built, and we turn our attention to the misspecification tests. The three tests mentioned in Section 5.6 were applied to both models, obtaining the p-values shown in Table 6.6. The tests indicate that both models are correctly specified, as there is no serial correlation amongst the residuals up to order 12, there is no change in the parameters amongst regimes and the variance of the residuals remains constant through time.

As a further, visual evidence that our models are effectively capturing the dynamics of the series, Figure 6.10 shows the residual series and its ACF and PACF.

Finally, to check the forecasting capabilities of the model, we reestimated it using only the data up to 1924. Then, the rest of the data, from 1925 to the end was predicted with the models, and the results of the forecasts is shown in Figure 6.11. Table 6.7 shows the root mean squared error for each model.

Table 6.6.: Results of misspecification tests for the lynx problem.

Test for q-order serial correlation			Test for parameter constancy	
	NCSTAR	NCGSTAR	NCSTAR	NCGSTAR
q	p-value	p-value	p-value	p-value
1	0.452	0.411	0.881	0.921
2	0.455	0.622		
3	0.354	0.587		
4	0.234	0.733	Test for constant variance	
5	0.834	0.234	p-value	p-value
6	0.236	0.834	0.179	0.645
7	0.716	0.532		
8	0.458	0.424		
9	0.347	0.672	Test for an extra rule	
10	0.673	0.562	p-value	p-value
11	0.702	0.623	0.212	0.328
12	0.422	0.789		

Table 6.7.: Root mean squared error in the forecasts of the lynx problem.

	NCSTAR	NCGSTAR
BFGS	0.0158	0.0392
GA	0.0156	0.0385

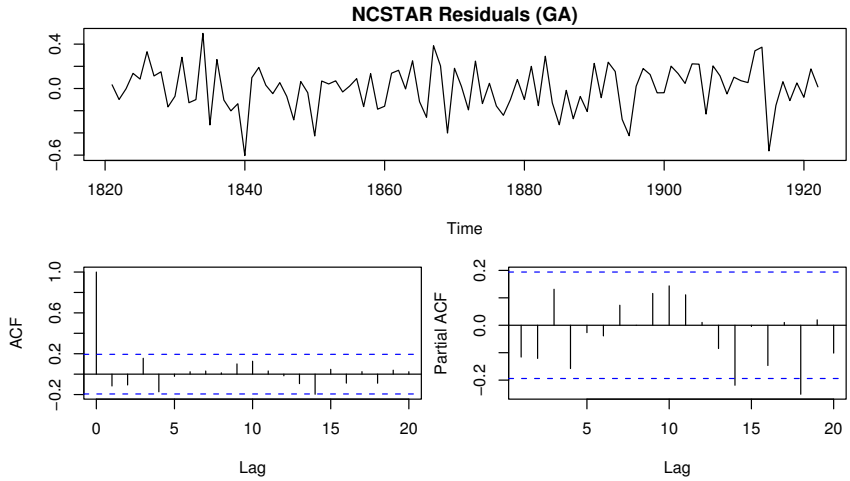

Figure 6.10.: NCSTAR residuals, ACF and PACF of the transformed lynx series.

6.3.2. Emergency call centre problem

An emergency call centre is a platform involving technological and human resources: computers, communication equipment and specially trained agents whose aim is to support the coordination between citizens and public safety institutions, with the final objective of solving incidents or alerts.

From an operative point of view, the management of such a call centre needs to know in an approximate manner, the work load to be faced in a specific period, so as to properly dimension the resources needed to attend it.

Traditionally, call centres have been modelled through queue models based in the producer/consumer paradigm. In this case, the alerting citizens would be the producers of calls and the centre operators would be the consumers of that work load.

In our case, we will be looking at data coming from an emergency call centre which attends calls from the region of Castilla y León, in Spain. This dataset was kindly provided by colleagues in the Universidad de Valladolid and the CARTIF foundation. Actually, in this centre there are no methods to forecast the work

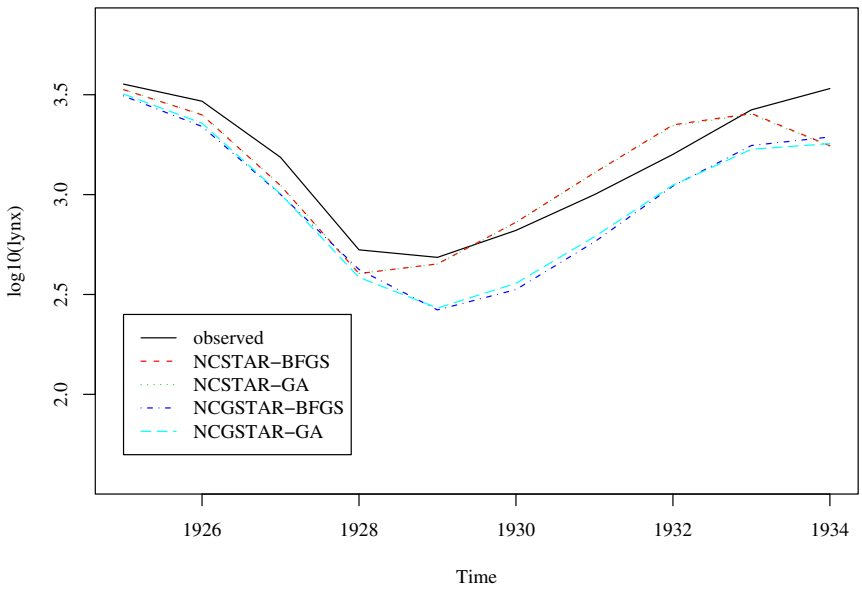

Figure 6.11.: Forecasting results for the transformed Lynx series.

load for a determined period. Operator dimensioning is performed through basic calculations in spread sheets of averages of previous daily values. In order to guarantee the level of attention, the centre has mechanisms that allow to face punctual call saturation moments generated by big emergencies. These mechanisms are a controlled overdimensioning of operators, the availability of auxiliary attention rooms and a number of reserve teams which can be called anytime and are available in less than an hour.

Planning of operators is performed early each morning, with a temporal horizon of 24h, building work shifts from an hourly profile of the work load based on previous experience. The availability of a forecast of the call number in a day, with some precision, would reduce the uncertainty affecting the number of operators really needed on each moment.

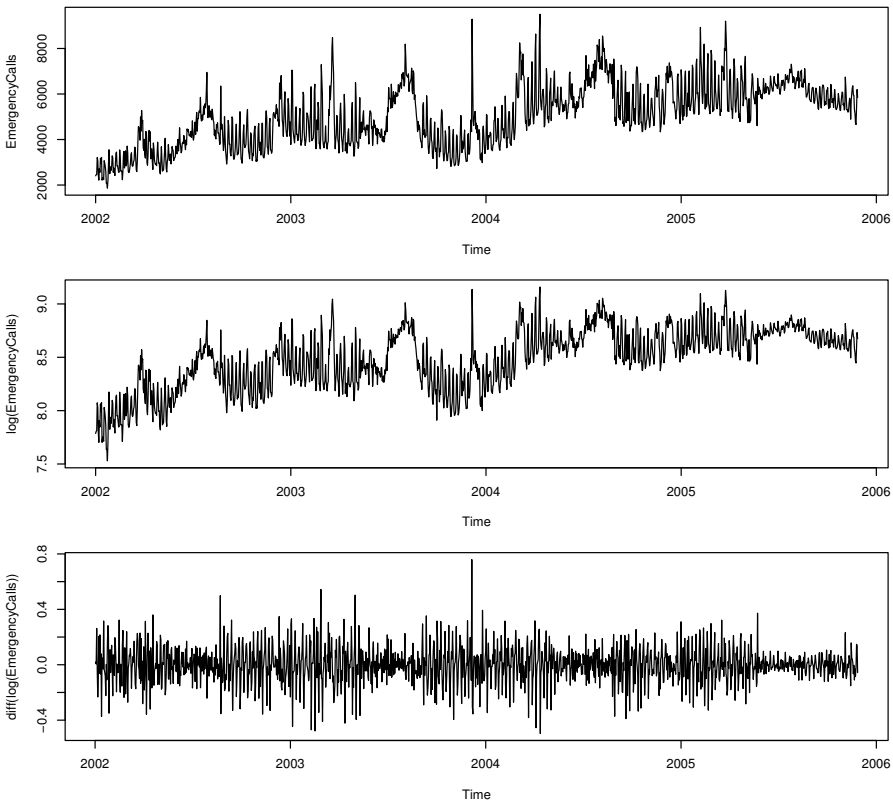

Figure 6.12.: Emergency call centre problem series (up), log-transformed series (centre) and differenced log-transformed series (down).

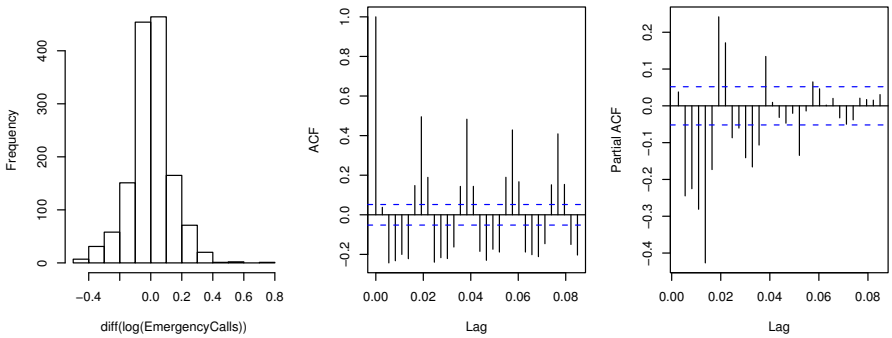

Figure 6.13.: Histogram, autocorrelation and partial autocorrelation functions for the transformed call centre series.

The aim of this study is to model the attended call series, using the logs of the centre from year 2002 to year 2006. Figure 6.12 displays the series in the top part. As we can see, the series is not stationary and shows high variability. For this reason we decided to transform it using a logarithm (Figure 6.12, centre) and finally to difference it (Figure 6.12, bottom).

Figure 6.13 shows the histogram of the transformed series, which shows a long right tail, being the computed kurtosis -0.5452, while the skewness had a value of 0.1285. The figure also shows the ACF and PACF functions, of which the first one shows a clear cyclic behaviour with period 7 and the second one shows significant partial autocorrelation in the first 7 lags. Attending to these diagrams, and the weekly nature of the series, we decided to fix to 7 the order of our models.

Figure 6.14 shows the autoregressions for the first 9 lags of the series. As we can see, in lag 7 there is some correlation, while the rest of the lags are not very clearly correlated. This suggests that this is a difficult to model problem, and we turn now to the next step of the modelling cycle.

The linearity tests against the NCSTAR and the NCGSTAR threw low values: $1.920692e{-}08$ and $1.321787e{-}22$, respectively. The tests indicate that the series is nonlinear and that it could be explained with these models.

As it was the case for the lynx problem, the modelling cycle ended up here by

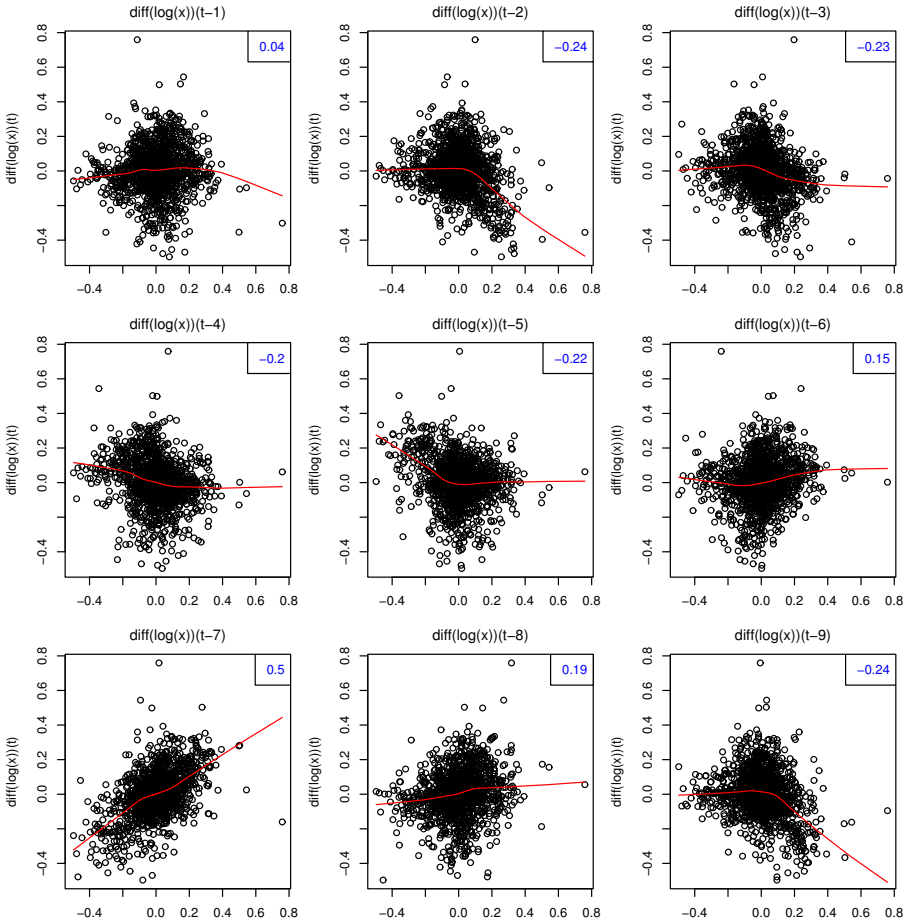

Figure 6.14.: Autoregressions for the transformed call centre series.

assigning two regimes to each model, which were fixed to:

$$\begin{aligned} y_t = &-0.0112 - 0.1827y_{t-1} - 0.1404y_{t-2} - 0.1663y_{t-3} - 0.1643y_{t-4} \\ &- 0.0666y_{t-5} - 0.0315y_{t-6} + 0.1110y_{t-7} + \\ &(0.03271 - 0.1952y_{t-1} - 0.3390y_{t-2} - 0.2442y_{t-3} - 0.0654y_{t-4} \\ &- 0.2683y_{t-5} - 0.09218y_{t-6} + 0.2282y_{t-7}) \\ &\times \mu_{\mathsf{S}}(\mathbf{x}_t; \boldsymbol{\psi}_{\mathsf{S}}) + \varepsilon_t \end{aligned} \quad (6.12)$$

in the sigmoid case, with $\boldsymbol{\psi}_{\mathsf{S}} = (\gamma, \boldsymbol{\omega}, c) = (53.5264, [0.5529, 0.4822, 0.3892, -0.1493, -0.3677, 0.2943, -0.2248], -0.0114)$, and

$$\begin{aligned} y_t = &-0.0956 - 0.4405y_{t-1} - 0.4771y_{t-2} - 0.2709y_{t-3} - 0.1915y_{t-4} \\ &- 0.05091y_{t-5} - 0.0247y_{t-6} + 0.2868y_{t-7} + \\ &(1.4904 + 4.4218y_{t-1} + 4.4400y_{t-2} + 1.4245y_{t-3} + 0.4831y_{t-4} \\ &- 1.97562y_{t-5} + 1.1356y_{t-6} - 2.8353y_{t-7} +) \\ &\times \mu_{\mathsf{G}}(\mathbf{x}_t; \boldsymbol{\psi}_{\mathsf{G}}) + \varepsilon_t \end{aligned} \quad (6.13)$$

in the Gaussian case, where $\boldsymbol{\psi}_{\mathsf{G}} = (\gamma, \mathbf{c}) = (1.0000, [-0.4964, -0.4964, -0.4964, -0.4964, -0.4964, -0.4964, 0.7591])$.

The first model obtained a residual standard deviation of $\hat{\sigma}_{\varepsilon,\mathsf{S}} = 0.0975$, while the second obtained a value of $\hat{\sigma}_{\varepsilon,\mathsf{G}} = 0.0989$. The value obtained for the AIC were $\mathrm{AIC}_{\mathsf{S}} = -6587$ and $\mathrm{AIC}_{\mathsf{G}} = -6548$ respectively, while the median average percentage error was $\mathrm{MAPE}_{\mathsf{S}} = 1.773\%$ and $\mathrm{MAPE}_{\mathsf{G}} = 1.896\%$.

Once both models were estimated through the standard procedure, we applied a metaheuristic to fine-tune the parameters. After the results obtained in Section 6.2, where the GA tended to obtain the best results, we decided to use just the Genetic Algorithm in this step.

Using the GA to fine tune the parameters, left us with the following two mod-

els:

$$\begin{aligned} y_t = &-0.0062 - 0.1835y_{t-1} - 0.1320y_{t-2} - 0.1621y_{t-3} - 0.1747y_{t-4} \\ &- 0.0786y_{t-5} - 0.0123y_{t-6} + 0.1235y_{t-7} + \\ &(0.0250 - 0.1854y_{t-1} - 0.3541y_{t-2} - 0.2473y_{t-3} - 0.0632y_{t-4} \\ &- 0.2656y_{t-5} - 0.1106y_{t-6} + 0.1990y_{t-7}) \\ &\times \mu_{\mathsf{S}}(\mathbf{x}_t; \boldsymbol{\psi}_{\mathsf{S}}) + \varepsilon_t \end{aligned} \quad (6.14)$$

in the sigmoid case, with $\boldsymbol{\psi}_{\mathsf{S}} = (\gamma, \boldsymbol{\omega}, c) = (84.1999, [0.1172, 0.5883, 0.4497, 0.3892, -0.1272, -0.3945, 0.2620], -0.0033)$, and

$$\begin{aligned} y_t = &-0.0962 - 0.4302y_{t-1} - 0.4473y_{t-2} - 0.3029y_{t-3} - 0.1914y_{t-4} \\ &- 0.05022y_{t-5} - 0.0201y_{t-6} + 0.2923y_{t-7} + \\ &(1.4422 + 4.3428y_{t-1} + 4.2434y_{t-2} + 1.4012y_{t-3} + 0.4231y_{t-4} \\ &- 1.9322y_{t-5} + 1.1416y_{t-6} - 2.5633y_{t-7} +) \\ &\times \mu_{\mathsf{G}}(\mathbf{x}_t; \boldsymbol{\psi}_{\mathsf{G}}) + \varepsilon_t \end{aligned} \quad (6.15)$$

in the Gaussian case, where $\boldsymbol{\psi}_{\mathsf{G}} = (\gamma, \mathbf{c}) = (3.4230, [-0.4124, -0.4254, -0.4235, -0.4891, -0.4623, -0.4982, 0.8003])$.

For these models tuned with the GA, the obtained residual standard deviation was $\hat{\sigma}_\varepsilon = 0.0974$ for the sigmoid and $\hat{\sigma}_\varepsilon = 0.0983$ for the Gaussian membership function. The value obtained for the AIC were $\text{AIC}_{\mathsf{S}} = -6590$ and $\text{AIC}_{\mathsf{G}} = -6570$ respectively, while the median average percentage error was $\text{MAPE}_{\mathsf{S}} = 1.766\%$ and $\text{MAPE}_{\mathsf{G}} = 1.866\%$.

Once the models are built, we can check their correctness by using the misspecification tests described in Section 5.6. The results are shown in Table 6.8, and the tests clearly indicate that both models are properly capturing the inner behaviour of the series. There is no serial correlation amongst the residuals up to order 12, the parameters remain constant through regimes and the variance of the residuals remains constant through time.

Again, we further assess these results in Figure 6.15, where the ACF of the residual series shows a very low correlation. The PACF still shows some correlation around lag 7, which indicates that the model is still missing some information, but the value is not very high so in spite of this fact we accept the model.

Table 6.8.: Results of misspecification tests for the emergency call centre problem.

Test for q-order serial correlation			Test for parameter constancy	
	NCSTAR	NCGSTAR	NCSTAR	NCGSTAR
q	p-value	p-value	p-value	p-value
1	0.005	0.002	0.0008	0.001
2	0.000	0.003		
3	0.000	0.010		
4	0.000	0.001	Test for constant variance	
5	0.000	0.000	p-value	p-value
6	0.011	0.000	0.000	0.000
7	0.014	0.000		
8	0.000	0.001		
9	0.000	0.000	Test for an extra rule	
10	0.000	0.000	p-value	p-value
11	0.000	0.000	0.212	0.138
12	0.000	0.000		

Table 6.9.: Root mean squared error in the forecasts of the call centre problem.

	NCSTAR	NCGSTAR
BFGS	0.0046	0.0051
GA	0.0045	0.0047

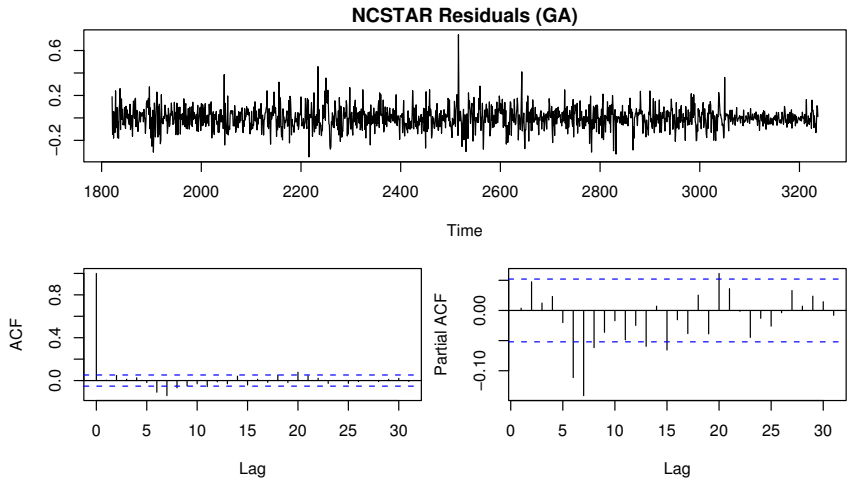

Figure 6.15.: NCSTAR residuals, ACF and PACF of the NCSTAR model for the transformed emergency call centre series.

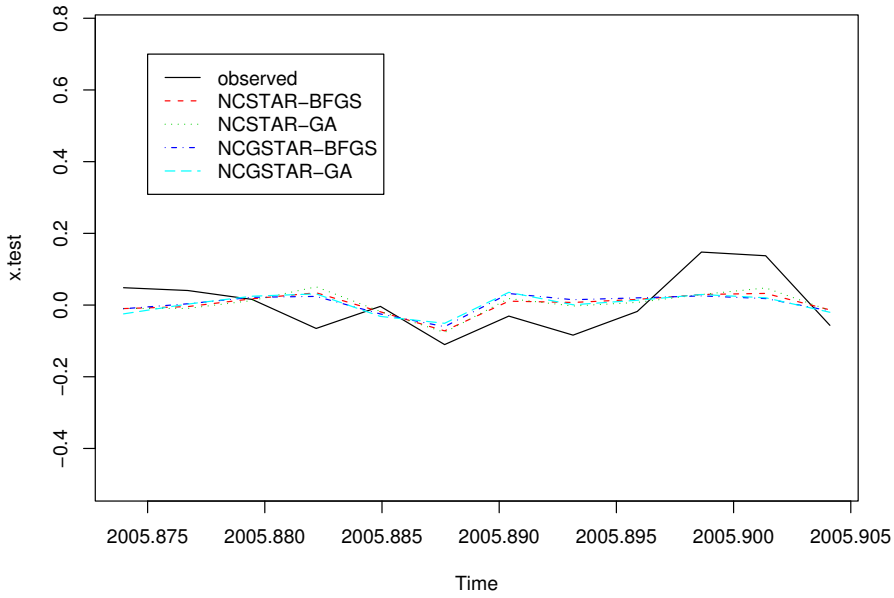

Figure 6.16.: Forecasting results for the transformed call centre problem series.

6.3.3. Airborne pollen series

This dataset was kindly provided by the Department of Botany of the University of Granada, and was object of a preliminary study published in the journal Expert Systems with Applications [59].

Forecasting future airborne pollen concentrations is undeniably of a high importance because of its medical, environmental and biological effects. The presence and amount of airborne pollen depends on a wide range of factors including meteorological (temperature, rain, humidity, wind etc.), biological (phenological and physiological state of plants, plant distribution etc.) and geological (topographic) issues. Actually, this is a highly chaotic and thus a hard to model problem.

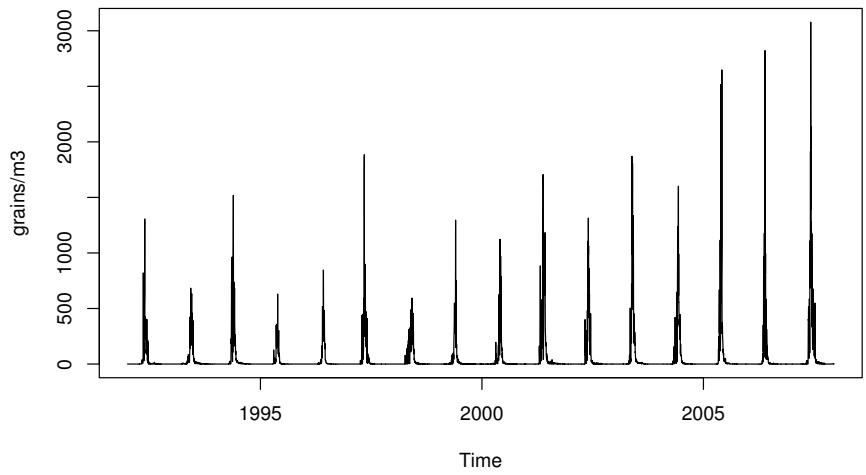

Figure 6.17.: Airborne pollen concentrations in the atmosphere of Granada from 1992 to 2007.

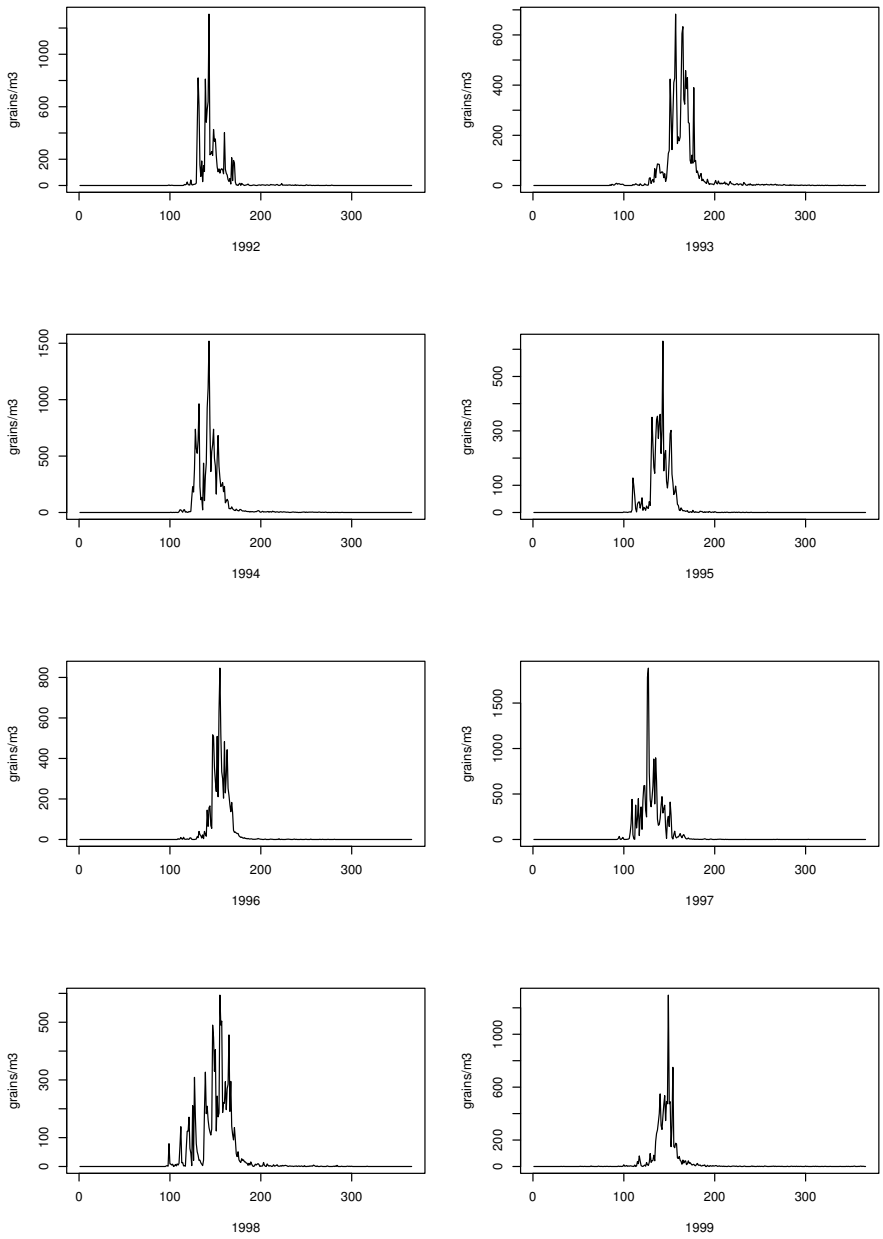

Figure 6.18.: 1992 to 1999 yearly pollen concentrations.

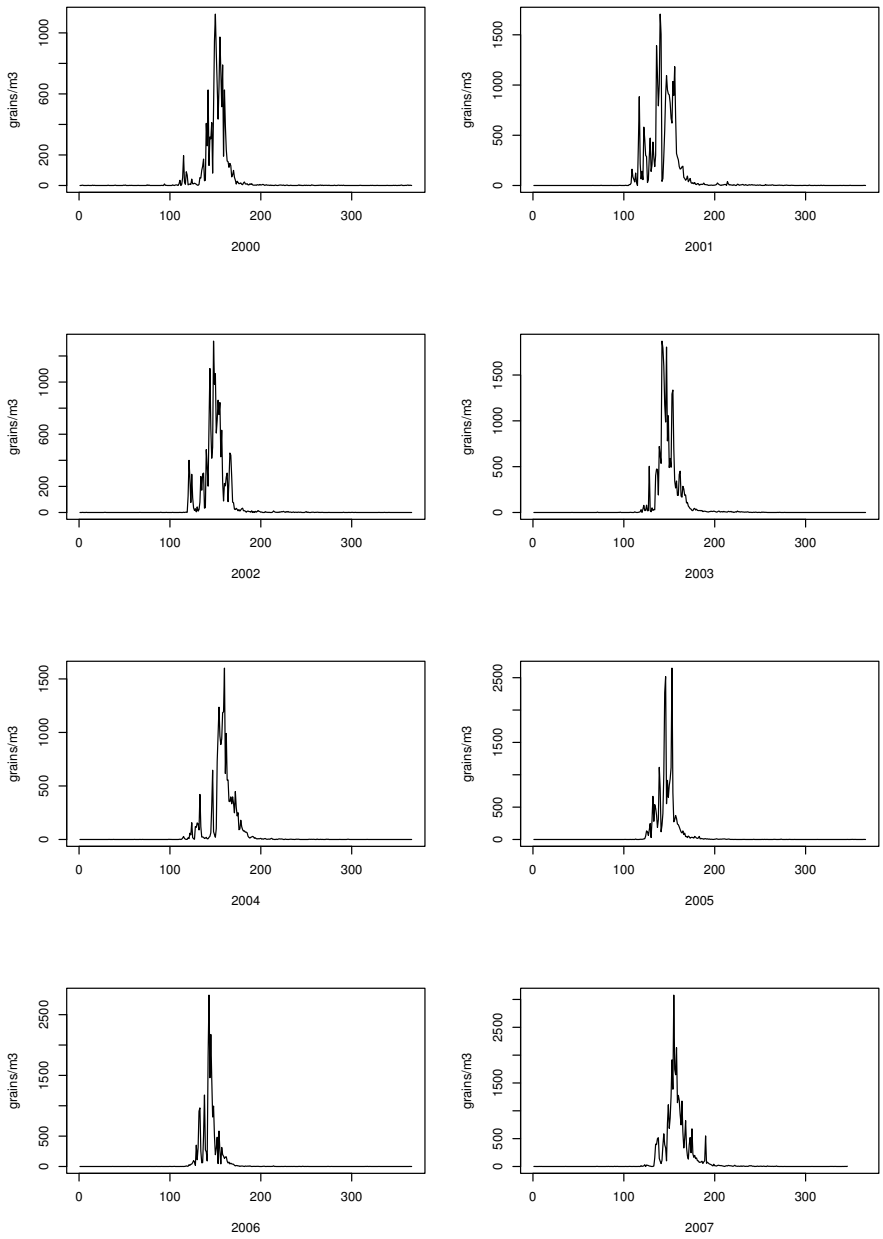

Figure 6.19.: 2000 to 2007 yearly pollen concentrations.

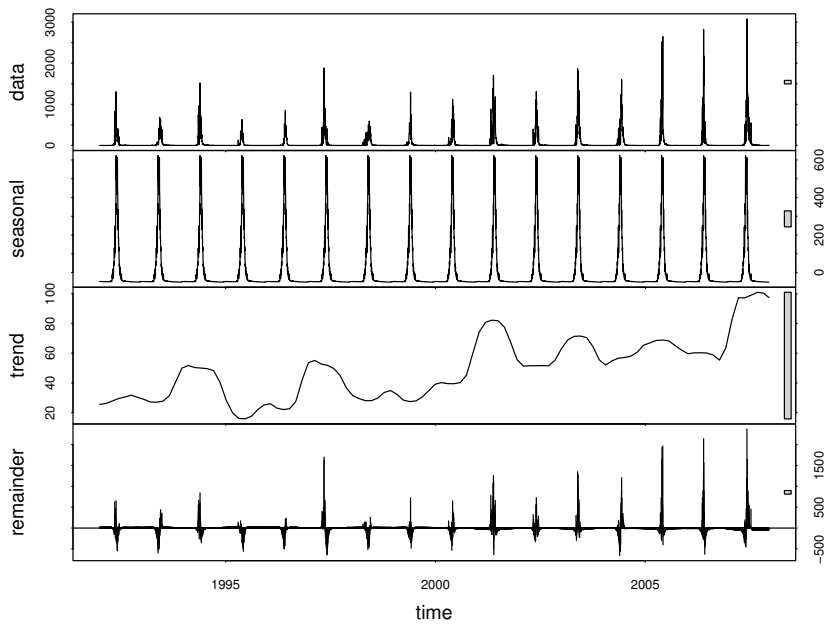

Figure 6.20.: Loess decomposition of the airborne pollen series.

Application of classical linear statistical methods to this problem has yielded results not entirely satisfactory [29, 20], whilst models based on Soft Computing techniques have proved successful [59]. Regarding other Aerobiology problems, some works have applied Neural Networks to pollen forecasting, reporting encouraging results [80, 71, 12].

In this case, the dataset used is a daily aerobiological log obtained over sixteen years, from 1992 to 2007 inclusive, in the city of Granada. Hence, 5821 data points were available. The complete series is shown in Figure 6.17, and the concentrations for each year is shown in Figures 6.18 and 6.19. The data were obtained following the standard methodology of the Spanish Aerobiological Network [22], and are measured in grains per cubic meter (grains/m^3) of air.

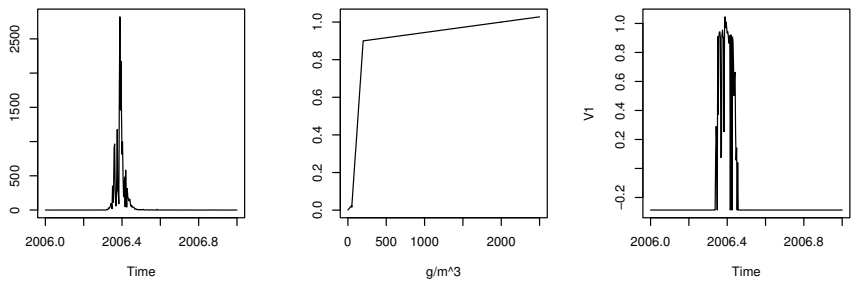

Figure 6.21.: Log-like transformation applied to the data.

From all the present phenotypes, only *Olea europaea* L. pollen values were considered, because this species is one of the most allergenic in the Iberian Peninsula, and has a very strong seasonality. Fortunately, it also has a very specific pollen morphology (it is monospecific) which allows biologists to perfectly identify it on the species level. This is important in order to effectively reduce the study to just one type of pollen hence producing a less noisy dataset with a consistent phenological behaviour. In addition, there exist other statistical studies about this pollen series, so more information was available for modelling [20, 2, 3].

A loess-based decomposition [17] into trend, seasonal and chaotic components was performed, and the result is shown in Figure 6.20. As we can see, the trend component is quite small compared to the other two components, a fact that suggests that removing the trend would not result in much benefit for the model, so we decided to keep the original series.

In order to reduce the high variability of the series, we applied an *ad hoc* preprocessing procedure, which is identical to the one used in [59]. Besides of rescaling the dataset into the interval $[0,1]$, special characteristics of the data suggested that further transformations could be in order. In particular, as we have seen, the presence of a high variance is normally tackled using a logarithmic transformation. Notwithstanding, in this case, the presence of a great amount of zero values in the series indicated that a linear log-like transformation could be used instead, considering 3 different intervals. The first interval,

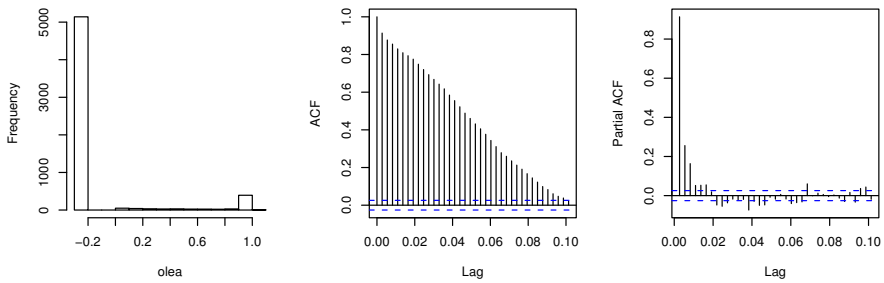

Figure 6.22.: Histogram, autocorrelation and partial autocorrelation functions for the transformed airborne pollen series.

which we shall call *low* interval, concerns all data below 50 grains/m^3 and was selected to try and separate the error of the broad regions of the series with values zero or close to zero, which are known to produce numerical instability in the models. The second or *medium* interval was fixed to contain values between 51 and 200 grains/m^3. This second threshold was proposed by the SEAIC (*Sociedad Española de Alergología e Inmunología Clínica*) for Olea pollen [25] as a general turning point between acceptable and risky concentrations, considering the alergological effects on the sensitive population. Figure 6.21 shows the transformation applied. The third interval (*high*) includes data from 201 grains/m^3 and above.

Once the preprocessing was done, we turned our attention to variable selection. We considered the autocorrelation function (acf) and the partial autocorrelation function (pacf) for the transformed dataset (Figure 6.22). These diagrams indicate that present values are influenced by previous days' values, decreasing its influence as the time lag increases. Concretely, the strongest partial autocorrelation is found in the previous six days, while the most recent three days are those showing a stronger ascendancy over the actual value. For this reason, and taking into account computational efficiency considerations, only three autocorrelation steps were considered here as inputs for the models.

The study of the autocorrelations, Figure 6.22, and of the regression functions

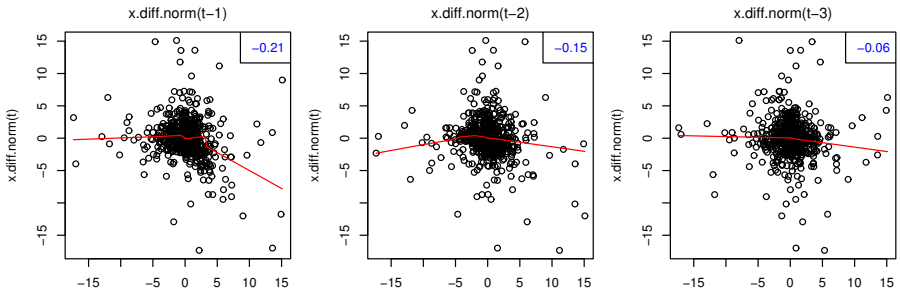

Figure 6.23.: Pollen series, nonparametric regression functions of the considered lags (1, 2 and 3).

shown in Figure 6.24 reveals the complexity of this dataset, as the relationship amongst the lags is not very strong nor clearly defined. Hence, this series poses a great challenge to the models, which will be taken here to their cutting edge.

At this point, the data was prepared for the rest of the steps of the modelling cycle. Not surprisingly, the linearity test threw a really low p-value (3.2391e−94 for the NCSTAR and 1.0764e−74 for the NCGSTAR), so the iterative procedure was applied to fix the number of required rules.

Table 6.10.: Error measures in the pollen problem.

	NCSTAR		NCGSTAR	
	BFGS	GAD	BFGS	GAD
σ_ε	0.1272	0.1219	0.1304	0.1203
AIC	-23929.72	-24357.45	-23695.03	-24516.47
MAPE	0.1503	0.1450	0.1543	0.1476

The number of rules was found to be quite high: 11 rules were estimated in both the NCSTAR and the NCGSTAR. This again is coherent with the expected complexity of the series, and, as said before, takes the models and the modelling cycle to an extreme situation.

The values for the standard deviation of the residuals and the AIC and MAPE,

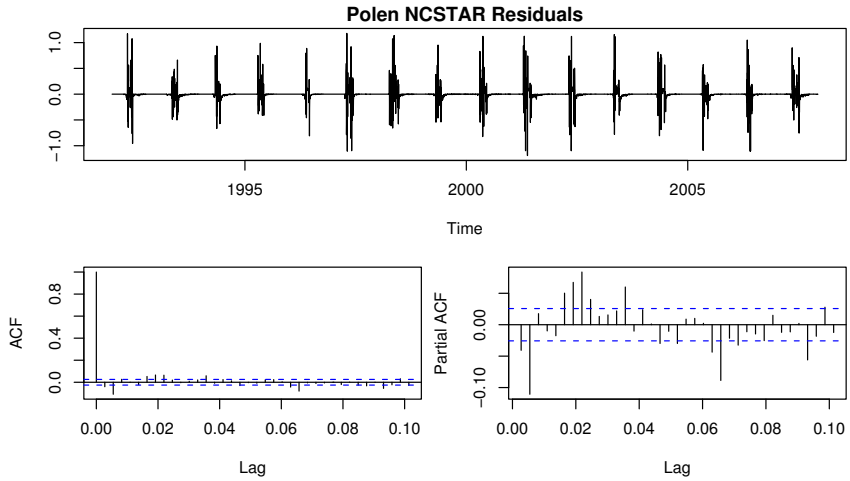

Figure 6.24.: Residual series for the NCSTAR model applied to the airborne pollen problem.

shown in Table 6.10, indicate that the best model was the NCGSTAR estimated with genetic algorithms, closely tied with the NCSTAR.

6.4. Discussion

In this Chapter we have explored the practical side of the results developed in previous chapters. On the one hand, we have tested the soundness and correctness of the statistical modelling cycle proposed in section 5.7 through its application to artificially generated time series. We have shown that the method works well already with a relatively small number of samples, and that when dealing with complex series there is a significant improvement in the use of metaheuristics.

On the other hand, we have applied the theoretical proposals to real world situations, in order to assess their practical utility. A well known series as the Canadian Lynx series was modelled with a statistically built FRBM, and such

Table 6.11.: Results of misspecification tests for the pollen problem.

Test for q-order serial correlation			Test for parameter constancy	
	NCSTAR	NCGSTAR	NCSTAR	NCGSTAR
q	p-value	p-value	p-value	p-value
1	0.232	0.241	0.912	0.879
2	0.132	0.663		
3	0.654	0.523		
4	0.238	0.233	Test for constant variance	
5	0.436	0.874	p-value	p-value
6	0.835	0.845	0.234	0.566
7	0.336	0.345		
8	0.228	0.334		
9	0.547	0.278	Test for an extra rule	
10	0.742	0.332	p-value	p-value
11	0.872	0.785	0.156	0.321
12	0.435	0.712		

model proved to be appropriate. We also studied the series of calls received by an emergency call centre during a period of five years. Despite the higher complexity of this series, successful models were also built. Also, the highly chaotic series of the airborne pollen concentrations was studied, and despite the extreme complexity of the series, good temptative models were also built.

7. Conclusions, main contributions and future research

In the framework of time series analysis, two approaches which come from different scientific disciplines and use different ideas to solve the same problems coexist. On the one hand, rooted in classic Statistics, the traditional autoregressive approach studies the existing relationships between the lagged values of a series in a linear manner. Recently, advanced models have been developed which allow for the description of nonlinear behaviours through locally distributed regimes.

On the other hand, during the last decades, the developments in Artificial Intelligence gave birth to a body of knowledge called Soft computing whose main aim was to solve complex problems which had a difficult analytical solution and that were affected by uncertainty and vagueness. Amongst the problems faced by the collection of techniques that form Soft Computing, time series analysis kept a prominent importance.

The objectives of this work were two: we wanted to explore the links between these two disciplines and we wanted to exploit these links in order to improve the available solutions for the problem of time series analysis. Globally, these objectives can be considered fulfilled: we found strong ties between models coming from both areas, and this fact helped us to transfer knowledge from one another.

More precisely, we proved that the main construct in which a whole family of Soft Computing models is based, the fuzzy rule, when applied to the problem of time series analysis, can be seen as a generalisation of the main linear statistical tool, the autoregressive model, which in turn is a basic piece for the advanced nonlinear statistical models.

This result encouraged us to study the so-called regime-switching models and we were able to prove close relations between them and a popular Soft Computing tool: the fuzzy rule-based model. Several recently developed regime-switching models were found to be particular cases of the fuzzy rule-based model. This is one of the main contributions included in this dissertation, and it successfully fulfils the first objective of our work.

Once the ties between fuzzy rule-based models and regime switching models were proven, we turned our attention to the benefits that could be obtained from this relation. There were two ways to work on this issue: extracting knowledge from regime-switching models to apply it to fuzzy rule-based models and using the Soft Computing techniques to improve the nonlinear statistical models.

Regarding Soft Computing models, a traditional criticism raised from other area's scientists was the alleged lack of a sound mathematical framework for the models. An engineering-oriented approach usually prevails in this area, stressing the effective resolution of problems over other considerations. After the equivalence result of this work, the possibility of developing a formal statistical framework for fuzzy rule-based models was open.

Translating the statistical theory from regime-switching models to fuzzy rule-based models is a second fundamental contribution of our work. We managed to effectively develop theoretical results like asymptotic stationarity or normality of the models, and this allowed us to obtain a new identification method for fuzzy rule-based models. Using this method, we can give a statistical justification for the number of fuzzy rules that are sufficient to properly model a given problem. Considering that determining the number of rules of a model is an open question that has received attention from researchers since the establishment of the fuzzy paradigm, this is one of the main results of this work.

But the exchange of knowledge, as stated above, can be twofold. The techniques of Soft Computing can be applied to the standard statistical approach, improving it. As an example, we decided to apply Genetic Algorithms to the model estimation phase of the autoregressive regime-switching models. In order to assess the benefits of such an exchange of knowledge, we developed an integrated modelling cycle which benefits from the strengths of both areas. This procedure gathers the statistical determination of the number of rules with the widely known estimation capabilities of Genetic Algorithms, and represents another original contribution included in this dissertation.

In order to assess the benefits of such a novel hybrid approach, a wide set of experiments was performed. This set was composed of a Montecarlo study, in which artificially generated series were faced to the model building and estimation procedure, and series coming from real-world problems. More precisely, we studied the widely known lynx population series and two original series: the daily received calls in an emergency call centre and the daily airborne pollen concentrations in the atmosphere of the city of Granada. These experiments shown the practical utility of the proposals, and they stood the comparison with previously existing alternatives.

The contributions put forward above allow us to conclude that the exchange of knowledge amongst the two areas is not only possible, but it also implies clear advantages in what should never be forgotten is the main goal of time series analysis: to understand the behaviour of data generating processes and to properly use that knowledge to solve the problems associated to them.

Future lines of research

There are still many paths to follow in the study of the benefits of the contributions of this work. These paths remain open for future developments, and can be summarised as follows.

- To extend the application of Soft Computing techniques to the existing statistical models, including feature selection techniques, other metaheuristics for optimisation (ant colonies, different evolutionary algorithms) etc. Improvements in both understandability and performance are expected through this line.
- To further study the implications of the newly stated statistical framework for fuzzy rule-based models, applying it to already studied problems and improving their results.
- To develop new hybrid models that exploit the links between both disciplines, as fuzzy rule-based models incorporating variance based statistical

models as GARCH. Another promising line is the trek for new links between other advanced statistical models different from multiple regime ones.

- The same vein of link between Statistics and Soft Computing techniques is expected to produce rich results in other kind of problems, e.g. classification and regression problems. Our intention is to develop results and hybrid modelling cycles in this area too.

Appendices

A. Software developed

The deployment of the tests and methods described in this document, as well as the tools required to perform the experimentation on them, required the development of a great quantity of software.

The choice of the language `R` was determined by a series of factors:

- Portability
- Flexibility
- Easy integration with existent code
- Object-oriented
- Easy prototyping

But, aside from these considerations, the possibility of including our code into an `R` extension highly influenced the decision. The work of a PhD Thesis should be immediately useful for society, and the inclusion of our models in the standard `R` distribution matches this objective.

A.1. `tsDyn`: an R extension

`tsDyn` is an R package for the estimation of a number of nonlinear time series models. The package is at an early stage, and may presumably change significantly in the near future. However, it is quite usable in the current version. Each function in the package has at least a minimal help page, with one or more working examples and detailed explanation of function arguments and returned values.

B. Resumen

En este apéndice se ofrece al lector en castellano[1] un resumen del contenido de la presente tesis. Para ello, primero se expone en qué consiste el problema general del análisis de series temporales, dando paso después a una revisión del estado de la cuestión. También se expondrán aquí los objetivos generales y la motivación que alienta este trabajo, cerrando el mismo una sucinta revisión de la bibliografía existente, una síntesis de los resultados obtenidos y el camino seguido hasta llegar a ellos y, por supuesto, las conclusiones finales.

B.1. Presentación

El análisis de series temporales es un área científica que forma parte de la Estadística, el Análisis de Datos, la Economía Estocástica y la Econometría. Durante los últimos años, ha sido un campo de estudio prolífico en cuanto a investigación básica y aplicaciones, habiéndose publicado numerosas innovaciones de métodos y herramientas. Sin embargo, no existe ninguna metodología que haya sido ampliamente aceptada y aun existen muchos asuntos por explorar.

De acuerdo con [90], tres son los objetivos fundamentales del análisis de series temporales: la *predicción*, el *modelado*, y la *caracterización*. El objetivo de la predicción consiste en el desarrollo de técnicas que permitan adivinar el comportamiento futuro a corto plazo de un sistema concreto, del que se conoce su

[1] Como resulta evidente a estas alturas, el resto de este documento está escrito en inglés. La obtención de la Mención Europea de Doctorado, y la evidencia de que es esa la lengua común en el ámbito de la investigación científica justifican esta decisión, que fue también avalada por la Comisión de Doctorado de la Universidad de Granada a solicitud de este doctorando. Para complacer los requerimientos de dicha Comisión en cuanto al bilingüismo de las tesis doctorales y, sobre todo, para que lo lean padres y amigas de la arriba firmante, he aquí este capítulo resumen en castellano. No se espere en éste ni la profundidad ni el rigor de los anteriores: sirve sólo a su doble propósito. Y vale.

comportamiento hasta el momento presente. Es, por supuesto, un problema interesante desde muchos puntos de vista, algunos de los cuales han estimulado durante años la imaginación de científicos, economistas y otras personas no necesariamente vinculadas con la Ciencia clásica.

El modelado, por otra parte, se ocupa de encontrar una descripción que capture con cierta exactitud algunas características relevantes del comportamiento a largo plazo del sistema. La relación entre estos dos primeros objetivos no es banal: encontrar ecuaciones que describan con propiedad el comportamiento a largo plazo de un sistema no es necesariamente la mejor manera de realizar predicciones sobre el futuro inmediato del mismo. También, suele ser el caso que un modelo válido para predicciones instantáneas o a corto plazo resulta incapaz de capturar un comportamiento a largo plazo.

El tercer objetivo, la caracterización del sistema, tiene que ver con la determinación, disponiendo de poco o ningún conocimiento *a priori*, de propiedades fundamentales del sistema como el número de grados de libertad o la cantidad de aleatoriedad. Éste puede confundirse con los dos objetivos anteriores, pero tiende, en realidad, a ser diferente: la complejidad de un modelo que sea útil para la predicción no tiene por qué estar relacionada con la complejidad real del sistema.

Pueden encontrarse aplicaciones del análisis de series temporales y de los métodos de predicción en una amplia variedad de áreas de conocimiento: desde el procesamiento de señales hasta la astronomía, pasando por el control de procesos industriales, la econometría, la meteorología, la física, la biología, la medicina, la oceanografía, la sismología, la psicología... Además, en la mayoría de estas disciplinas, obtener predicciones fidedignas de valores futuros de algunas características de los procesos estudiados es crucial, ya que esto permite controlar de forma óptima las condiciones futuras del proceso global en que están incluídos.

Pese a que la necesidad de predecir es tan antigua como las áreas de aplicación mencionadas (vale decir: tan vieja como la propia Ciencia), el análisis de series temporales es una disciplina que se estableció formalmente durante el siglo pasado. Antes de 1920, la predicción era llevada a cabo simplemente mediante la extrapolación de las series mediante un ajuste global en el dominio del tiempo. El comienzo de la predicción de series temporales "moderna" puede establecerse en 1927, que es cuando Yule inventó la técnica autorregresiva para predecir el número de manchas solares producidas cada año. Su modelo predecía el valor

siguiente de la serie a partir de una suma ponderada de las observaciones anteriores. Para obtener un comportamiento interesante a partir de un modelo lineal como este, es necesario asumir una intervención exterior, como son los llamados *choques externos* o aleatorios, ruido. Durante los cincuenta años posteriores al trabajo de Yule, el paradigma reinante en la predicción fue el de los modelos lineales dirigidos por ruido.

No obstante, el hecho de que series aparentemente complicadas pudieran ser generadas por ecuaciones muy sencillas apuntaba a la necesidad de un marco teórico más general en el ámbito del análisis y predicción de series temporales.

Alrededor de los años 1980, tuvieron lugar dos innovaciones cruciales. Ambas fueron debidas a la disponibilidad de poderosas computadoras que permitían registrar series temporales mucho más largas, aplicarles algoritmos más complejos y visualizar interactivamente los resultados de los mismos. La primera de estas innovaciones, la reconstrucción del espacio de estados mediante el uso de retardos 'incrustados', surgió a partir de ideas provenientes del estudio de la topología diferencial y de los sistemas dinámicos, y proporcionó una técnica para reconocer cuándo una serie temporal ha sido generada por medio de ecuaciones determinísticas y, de ser así, para entender la estructura geométrica subyacente al comportamiento observado. La segunda innovación fue el desarrollo del área del aprendizaje automático, representada por las redes neuronales artificiales, que pueden explorar de forma adaptativa un amplio espacio de modelos potencialmente válidos.

El giro que en aquellos años dió la Inteligencia Artificial, la cual comenzó a orientarse más hacia los métodos dirigidos por los datos, permitió su aplicación al campo de las series temporales. Pero, también, permitió que las series temporales, registradas ahora mediante volúmenes de datos varios órdenes de magnitud mayores, estuvieran listas para ser analizadas mediante técnicas de aprendizaje automático, las cuales requieren conjuntos de datos relativamente grandes.

B.2. Motivación

La construcción de modelos es uno de los pasos básicos en el análisis de series temporales. Como hemos dicho, estos modelos han sido históricamente lineales y de tipo regresivo, lo cual implica un enfoque que no siempre es realista. Recien-

temente, se ha incrementado el uso de modelos no lineales, cuyos parámetros han de ser estimados utilizando los datos disponibles, de forma paramétrica o no paramétrica. Antes de la fase de estimación de parámetros hay una fase de identificación, en la que se determina qué tipo de modelo debería ser usado y cuánta memoria (qué cantidad de valores pasados) debería utilizar. Finalmente, hay una fase de comprobación diagnóstica para establecer si el modelo funciona de forma satisfactoria o no.

El enfoque estadístico está fuertemente apoyado en la aplicación formal de estos tres pasos: *identificación*, *estimación* y *evaluación*. Ha desarrollado técnicas de examen que garantizan que el modelo resultante tiene buenas propiedades estadísticas, y proporciona demostraciones formales de ellas. Esto conduce a una sólida fundamentación matemática de las predicciones, y permite poseer conocimiento *a priori* sobre las capacidades de un modelo o método.

Sin embargo, la prueba formal de las propiedades de un método requiere usualmente el fijar restricciones que pueden ser, en ocasiones, poco realistas. Estas restricciones formales usualmente incluyen condiciones difíciles de probar como que el espacio de los parámetros sea compacto. Para obtener resultados aceptables en problemas reales, estos requerimientos son a veces ignorados o atenuados.

La Computación Flexible, por otro lado, se conduce por un enfoque más pragmático cuyo objetivo es la obtención de soluciones que funcionen. Debido principalmente a la complejidad de los modelos, no suelen derivarse demostraciones formales de convergencia, y por tanto no suelen establecerse restricciones. Sus defensores argumentan que el no tener (o tener menos) requerimientos *a priori* hace que los métodos de Computación Flexible sean aplicables en más casos y más fáciles de usar.

La obtención de un enfoque que conjugue las ventajas de ambos campos (análisis estadístico de series temporales y Computación Flexible) es la motivación fundamental de este trabajo. Una perspectiva de colaboración —que sustituya a la confrontación que ha regido históricamente las relaciones entre ambas áreas— podría resultar en importantes beneficios para la predicción de series temporales, y es el motivo que impulsa este trabajo.

B.3. Objetivos

Los objetivos principales de este trabajo son:

- Explorar relaciones formales entre modelos provenientes del área del análisis estadístico de series temporales y del área de la Computación Flexible.
- Explotar esas relaciones para obtener un fructífero intercambio de conocimiento entre ambas áreas.

Como objetivo secundario, será considerada asimismo la aplicación de los avances obtenidos a la resolución de problemas reales que involucren series temporales.

B.4. Revisión bibliográfica

La bibliografía clásica sobre series temporales abarca una lista bastante extensa, y cada día se publican nuevos volúmenes de referencia. Sin embargo, podemos citar [8, 70, 4] entre los libros más empleados en el ámbito del enfoque lineal para las series temporales. El libro de Tong sobre el modelado no lineal, [86], caracterizado por una perspectiva más moderna, es uno de los textos más influyentes de su rama.

La Computación Flexible o sus relaciones con los métodos estadísticos no ha sido estudiada por muchos investigadores del ámbito estadístico. Hay una excepción, sin embargo: la popularidad de las redes neuronales artificiales se ha extendido también entre los investigadores estadísticos, y aquellas han sido ampliamente utilizadas para el análisis de series temporales. En [103] (publicado en una revista del ámbito estadístico clásico), puede encontrarse una profunda revisión del estado de la cuestión, que incluye comparaciones entre modelos. Otra revisión interesante es [72].

Sin embargo, son menos los trabajos disponibles en este área acerca de otras ramas de la Computación Flexible. Concretamente, los modelos difusos o neurodifusos han sido mencionados en [21, 75, 27, 57].

En el campo de la investigación en Computación Flexible, los investigadores se han acercado al problema del análisis o la predicción de series temporales fundamentalmente de dos formas distintas. Por un lado, algunos investigadores

han visto en las series temporales una enorme colección de conjuntos de datos que podrían ser utilizados sin más para probar modelos nuevos o ya existentes. Este enfoque no es incorrecto en sí, pero generalmente conlleva el no considerar las características particulares que diferencian a una serie temporal de otro conjunto cualquiera de datos, descartando así todo el conocimiento científico acumulado durante años para este problema específico. Hay muchos ejemplos que ilustran esta situación, como las pruebas realizadas con modelos como ANFIS [39, 40, 1], EFuNN [43, 44] o ANNBFIS [56]. Otros ejemplos pueden encontrarse en [31, 46, 67, 60, 55].

Por otro lado, hay artículos que presentan modelos basados en Computación Flexible y específicamente diseñados para modelar y predecir series temporales concretas. La predicción de la demanda eléctrica es uno de los problemas más afrontados [42, 45, 82, 18] pero hay un gran número de otros ejemplos: predicción bursátil [51, 50, 88], biológica [29, 68] etc. Estos investigadores intentan modelar o predecir casos reales utilizando modelos genéricos y adaptándolos a algunas características observables de los datos, pero tampoco hacen uso de las herramientas proporcionadas por el análisis clásico de series temporales, como la inferencia estadística.

B.5. Síntesis de aportaciones

A continuación se recapitulan las principales aportaciones de esta tesis junto a los procedimientos llevados a cabo para su obtención. A fin de contribuir a la coherencia de este capítulo, se revisan previa y brevemente algunas de las herramientas empleadas.

B.5.1. Modelos basados en reglas difusas

La lógica aristotélica, de la que forman parte la Lógica Proposicional y la Lógica de Predicados, no consigue cubrir ciertas características inherentes al conocimiento humano como son la incertidumbre, la vaguedad y la información incompleta, por ejemplo. Para la resolución de estos problemas se han propuesto muchas variantes de la lógica clásica, aunque ninguno ha logrado resultados enteramente satisfactorios. Para hacer frente a los diversos inconvenientes de la lógica se han propuesto métodos de representación del conocimiento alter-

nativos. El más utilizado es la *regla de producción*. La forma de las reglas de producción es

> SI se da un conjunto de condiciones ENTONCES se puede inferir un conjunto de conclusiones.

La primera parte (antes de ENTONCES) es conocida como *antecedente*, condición o parte de la izquierda. La segunda parte suele ser llamada *consecuente*, conclusión o parte de la derecha. En el consecuente, también es habitual indicar un conjunto de acciones en lugar de un conjunto de consecuencias. Este modelo de representación del conocimiento ofrece diversas ventajas:

a. Constituye un modelo natural de representación del conocimiento. Su ámbito de aplicabilidad es muy amplio cubriendo múltiples dominios cuyo conocimiento se expresa en términos de relaciones causa-efecto.

b. Es un modelo de representación mixto, que tiene parte declarativa y parte procedural. La mezcla es tal que la unión de ventajas supera a la unión de inconvenientes de ambos enfoques.

c. Existe un modelo muy antiguo para representar la regla: la implicación lógica. Aunque el sentido de la regla no se corresponde exactamente con la implicación material de la lógica clásica.

Todas estas ventajas han propiciado un enorme auge de los sistemas basados en estas reglas, llamados *sistemas de producción*. La gran mayoría de los sistemas expertos desarrollados y, sobre todo, los explotados en la práctica, son sistemas de producción, por lo que constituyen el principal paradigma de sistema basado en el conocimiento. La aplicación de estos sistemas ha resultado un éxito contundente en multitud de campos. Sin embargo, no están exentos de desventajas, principalmente relacionadas con la forma (precisa) de manejar un conocimiento que, habitualmente, proviene de una fuente imprecisa (el experto humano).

El desarrollo de sistemas basados en el conocimiento se realiza en varias etapas, de las que la *Adquisición del Conocimiento* se ha convertido en un auténtico cuello de botella. Su objetivo es la extracción de reglas y heurísticas usadas por el experto para resolver problemas de su área, y representarlas en una forma

adecuada para su procesamiento empleando máquinas. Tradicionalmente se lleva a cabo usando técnicas manuales, consistentes en múltiples entrevistas entre el ingeniero del conocimiento y el experto. Sin embargo, estos procedimientos no siempre producen los resultados apetecidos por múltiples causas, entre ellas la dificultad de los expertos para verbalizar sus conocimientos.

Para mejorar el proceso de Adquisición del Conocimiento se busca su automatización. Desde hace bastante tiempo, muchos investigadores se han dedicado a ello obteniendo resultados dispares. Una de las principales vías por las que se ha atacado este problema es la del Aprendizaje Automático. La razón es muy simple: al experto puede que le cueste trabajo verbalizar y transmitir sus conocimientos, pero lo que seguro puede hacer es aplicarlos para resolver problemas de su ámbito. En esta situación, el ingeniero del conocimiento puede limitarse a observar y tomar nota de todo lo que hace el experto. Además, es frecuente que se disponga de extensos registros de la actuación del experto. Si podemos hacer que un sistema aprenda ese comportamiento del experto, habremos conseguido una representación del conocimiento en forma manejable por un ordenador.

En el Aprendizaje Automático tienen cabida multitud de técnicas y algoritmos, la mayoría de los cuales se han aplicado a esta tarea de Adquisición del Conocimiento. Dos de las técnicas empleadas con mayor profusión durante los últimos años son: Algoritmos Genéticos y Redes Neuronales Artificiales.

La mayoría de las técnicas de Aprendizaje Automático parten de un conjunto de ejemplos y devuelven el conocimiento representado mediante reglas de producción, ya que ésta es la representación más habitual en los sistemas basados en el conocimiento. A los métodos basados en estas técnicas se les denomina *métodos para extracción de reglas*.

El razonamiento que implementan estos SBC, basado en la Lógica Clásica, es preciso, no admite vaguedades de ningún tipo. Sin embargo, los seres humanos somos perfectamente capaces de obtener conclusiones a partir de afirmaciones o hechos vagos e imprecisos. Este tipo de razonamiento tiene un carácter más cualitativo que cuantitativo y se aleja del ámbito de la Lógica Clásica. En él se fundamentan las capacidades humanas para comprender el lenguaje natural, interpretar textos manuscritos, desarrollar tareas que requieren entrenamiento o habilidad mental y, en definitiva, tomar decisiones racionales en ambientes complejos y/o inciertos. Esta clase de razonamiento se conoce como *Razonamiento Aproximado*.

Los principios del Razonamiento Aproximado son claramente opuestos a la tradición cuantitativa y precisa de la Ciencia. Esto impidió que los investigadores le dedicasen la debida atención hasta los años setenta. La formalización del Razonamiento Aproximado mediante reglas similares a las de la lógica clásica ha captado una notoria atención en los últimos años, coincidiendo con el creciente interés que han despertado todos los intentos de conseguir mejores descripciones de los complejos procesos que subyacen en el razonamiento y la toma de decisiones.

Los conceptos fundamentales para la formalización del Razonamiento Aproximado se encuentran en la Teoría de Subconjuntos Difusos, y más concretamente en la Lógica Difusa, cuyo padre es L.A. Zadeh, quien merece por ello la consideración de pionero en tal formalización.

Reglas difusas

L.A. Zadeh [98] introdujo un nuevo punto de vista para la Ciencia al considerar que no debíamos huir sistemáticamente de la incertidumbre y la vaguedad, sino buscar medios para representarla que además permitiesen manejarla y controlarla. Zadeh se percató de que el concepto clásico de conjunto no representaba adecuadamente muchos de los conjuntos con que nos manejamos habitualmente. Considérese, por ejemplo, el conjunto de las personas jóvenes. Claramente, una persona con 18 años de edad es clasificada como joven. Con la misma claridad, clasificamos como no joven a alguien con 40 años. Alguien con 27 años es joven, pero menos que alguien con 25, y más que alguien con 30 años. Pero, ¿dónde está el límite entre el ser joven y el no serlo? No podemos establecer un punto de corte claro, sino que la propiedad "ser joven" es de carácter gradual. El concepto de joven no se puede definir de forma precisa, sino que está aquejado de incertidumbre porque es vago. La transición entre cumplir completamente la propiedad y el no cumplirla no es brusca, sino suave, continua. Esta naturaleza gradual de cumplimiento de una propiedad no se puede representar con un conjunto clásico, en el que dado un elemento sólo tiene dos opciones: o pertenece al conjunto o no pertenece.

Los métodos tradicionales de tratamiento de información, basados en la lógica clásica, actúan sobre datos precisos. No son válidos para procesos de razonamiento con conceptos vagos, inciertos o imprecisos. El tratamiento automático de este tipo de información requiere otro tipo de técnicas. La aportación de Za-

deh es un conjunto de tales técnicas cuya formulación algebraica se plasmó en la *Teoría de Conjuntos Difusos*.

El concepto fundamental en esta teoría es el de *conjunto difuso*, que recoge conjuntos cuyos bordes no están definidos con precisión. Formalmente, un conjunto difuso[2] A de un conjunto referencial o dominio U se define como un conjunto cuya función indicadora toma valores en $[0,1]$, en lugar de en $\{0,1\}$, cual es el caso de los conjuntos clásicos. De esta forma, se dice que un elemento pertenece a un conjunto con un determinado grado de pertenencia, cuyos posibles valores constituyen un rango continuo. La función indicadora o característica, más habitualmente llamada función de pertenencia, se representa por μ:

$$\mu_A : U \rightarrow [0,1].$$

Dado el elemento $x \in U$, su grado de pertenencia al subconjunto difuso A es $\mu_A(x)$. Es inmediato observar que un conjunto ordinario[3] constituye un caso particular y extremo de conjunto difuso.

Partiendo de esta definición de conjunto, Zadeh extendió otros conceptos relacionados con él como los de las operaciones de unión, intersección y complemento, producto cartesiano, relaciones, etc., hasta establecer una teoría completa sobre estos conjuntos. La extensión es tal que cuando los conjuntos involucrados son conjuntos clásicos, entonces las operaciones se reducen igualmente a las clásicas. Así mismo, Zadeh también estableció una lógica asociada a esta teoría de conjuntos, la Lógica Difusa, que ha supuesto una alternativa a la lógica clásica para el tratamiento de conocimiento aquejado de incertidumbre y vaguedad. Existen múltiples textos dedicados a una exposición detallada de la Teoría de los Conjuntos Difusos y sus principales aplicaciones [23, 48].

Los ordenadores actuales son muy poco efectivos en el tratamiento y reproducción de comportamientos y formas de razonar humanas. Zadeh interpreta esto como una manifestación de lo que denomina "Principio de Incompatibilidad" según el cual, precisión y complejidad son propiedades incompatibles. De este modo, las técnicas convencionales basadas en la manipulación precisa de datos numéricos resultan intrínsecamente insuficientes para modelizar el conocimiento humano y los complejos procesos de toma de decisiones. Sin embargo,

[2] El término correcto es el de *subconjunto difuso*, sin embargo en la literatura predomina el término *conjunto difuso* originado por el anglosajón *fuzzy set*.

[3] También llamado *conjunto crisp* en la literatura sobre conjuntos difusos. Por extensión, se aplica el término *crisp* a todos los conceptos clásicos para distinguirlos de sus extensiones difusas.

este tipo de información sí puede modelarse usando los conceptos contenidos en la Teoría de Conjuntos Difusos.

Las personas habituadas a trabajar con conjuntos, suelen captar y comprender bien las descripciones funcionales o gráficas de conjuntos. Pero esto no ocurre comúnmente con los seres humanos más acostumbrados, en general, a expresar e intercambiar información de modo lingüístico. Por esta razón y, aplicando nuevamente el "Principio de Incompatibilidad", parece adecuado el empleo de etiquetas o valoraciones lingüísticas para representar incertidumbre o información imprecisa, modelo con el que los decisores suelen trabajar más cómodamente.

Con estas observaciones como punto de partida, Zadeh introduce el concepto de variable lingüística [100, 101, 102]. De modo informal puede ser definida como una variable sobre un dominio de discurso que toma valores en un conjunto de etiquetas lingüísticas. A las etiquetas se asigna una interpretación semántica definida por un conjunto difuso sobre el dominio del discurso. Más formalmente:

Una *variable lingüística* es una quíntupla $(x, T(x), U, G, M)$, donde: x es el nombre de la variable; $T(x)$ es el conjunto de términos lingüísticos de x; U es el universo del discurso; G es la gramática con que se generan los términos de $T(x)$; y M es una regla semántica que asocia un significado, $M(t)$, a cada $t \in T(x)$, donde $M(t)$ es un subconjunto difuso de U.

En principio, el número de elementos de $T(x)$ puede ser cualquiera. Ahora bien, si el número de términos aumenta indefinidamente se llegará a la indistinguibilidad semántica de algunos de ellos como consecuencia de la naturaleza aproximada, vaga o imprecisa de la información que cada uno de estos términos contiene acerca de x.

En términos de la lógica clásica, cuando queremos representar que un objeto x cumple la propiedad P, o sea, x pertenece al conjunto de los objetos que cumplen P, escribimos "x es P". Si la propiedad es de carácter difuso, entonces el conjunto P es difuso, y la proposición "x es P" se denomina proposición difusa.

Para construir enunciados más complejos, en la lógica clásica se utilizan los conectivos lógicos: negación, conjunción, disyunción e implicación. Estos operadores actúan asociando un valor de verdad a la proposición compleja a partir de los valores de verdad de las proposiciones simples que la constituyen. Para conectar proposiciones difusas necesitamos extender la definición de estos operadores. Este ha sido un amplio campo de investigación en el ámbito de lógica difusa, produciendo múltiples definiciones alternativas para cada uno de los ope-

radores.

La negación lógica se ha ampliado a través de las funciones de negación, de entre las que $n(x) = 1 - \mu(x)$ es la más habitual.

Para la conjunción se emplean funciones conmutativas, asociativas y monótonas, no decrecientes en cada argumento, denominadas t-normas. Las funciones más comúnmente usadas son el mínimo y el producto. La disyunción se modela mediante una familia de funciones conocidas como t-conormas. Para cada t-norma existe una t-conorma dual. Así, las que se emplean con mayor frecuencia son el máximo y la suma acotada, duales del mínimo y del producto, respectivamente.

Las posibles elecciones para la función de implicación, que extiende la implicación lógica, son aún más numerosas que en los casos anteriores. Estas definiciones se agrupan en diversas familias s-implicaciones, r-implicaciones y ql-implicaciones.

Existen diversos trabajos en los que se recogen y analizan las distintas familias de operadores que modelan los conectivos lógicos [23, 48].

Si en una regla de producción clásica reemplazamos las proposiciones clásicas por difusas, resulta una *regla difusa*. En su versión más general, una regla difusa adopta la forma:

$$\begin{aligned} &\text{IF } f_1(x_1,\ldots,x_n) \text{ IS } A^1 \ \theta_1 \ \cdots \ \theta_{k-1} \ f_k(x_1,\ldots,x_n) \text{ IS } A^k \\ &\qquad\qquad\qquad\qquad\qquad\qquad \text{THEN} \\ &\qquad\qquad g_1(y_1,\ldots,y_m) \text{ IS } B^1 \ \psi_1 \ \cdots \ \psi_{j-1} \ g_j(y_1,\ldots,y_m) \text{ IS } B^j \end{aligned} \quad \text{(B.1)}$$

con f_i, g_l funciones, θ_i, η_l conectivos lógicos y A^i, B^l, etiquetas lingüísticas o conjuntos difusos.

Pero esta es una regla demasiado compleja para usarla de modo efectivo. Lo habitual es que las f_i sean las proyecciones y los A^i y B^j sean conjuntos difusos fijos, adoptando el siguiente aspecto:

$$\begin{aligned} &\text{IF } x_1 \text{ IS } A_1 \ \theta_1 \ \ldots \ \theta_{n-1} \ x_n \text{ IS } A_n \\ &\qquad\qquad\qquad\qquad \text{THEN } y_1 \text{ IS } B_1 \ \eta_1 \ \ldots \ \eta_{m-1} \ y_m \text{ IS } B_m \end{aligned} \quad \text{(B.2)}$$

con θ_i y ψ_j conectivos conjuntivos o disyuntivos. En particular, en el antecedente, θ_i suelen ser conjunciones.

Esta es la forma más frecuente cuando la regla procede de un experto humano o está destinada a ser manejada por éste. Cuando el objetivo es simplemente aproximar el sistema sin prestar mucha atención a su inteligibilidad, resulta más efectivo reemplazar la parte consecuente por una función de las entradas. Si se trata de una función lineal, entonces tenemos las reglas del tipo TSK (Takagi-Sugeno-Kang [78]), cuyos antecedentes se conectan mediante un operador conjuntivo modelado mediante el producto:

$$\text{IF } x_1 \text{ IS } A_1 \ \wedge \ x_2 \text{ IS } A_2 \ \wedge \ \ldots \ \wedge \ x_n \text{ IS } A_n \text{ THEN } y = p(x_1, x_2, \ldots, x_n) \quad \text{(B.3)}$$

con $p(x_1, x_2, \ldots, x_n)$ una función lineal.

Una proposición difusa "x es A" se puede ver también como una proposición lingüística, donde x es una variable lingüística y A es un conjunto difuso que constituye la asignación semántica a una etiqueta del conjunto de términos de x. Como consecuencia, un modelo más adecuado que la lógica clásica, para el manejo de la información lingüística es el de variable lingüística. La aplicación de este modelo da lugar a un nuevo punto de vista para tratar los problemas, que se conoce como *enfoque lingüístico*, bajo el cual las variables que intervienen se tratan como variables lingüísticas, es decir, admitiendo que sus valores vienen dados lingüísticamente.

Entre las propiedades más importantes de las reglas difusas destacan:

- Representación de la incertidumbre. Las reglas difusas permiten recoger conceptos vagos, faltos de precisión. Por este motivo son más cercanas al modo humano de manejar información, que las reglas de producción clásicas.

- Modelo compacto de representación de información. Con una sola regla difusa se puede expresar toda la información contenida en un conjunto de reglas clásicas.

- Carácter local. La información que describe una regla difusa, suele atañer sólo a una zona muy localizada del dominio completo del problema. Sus interacciones con otros elementos descriptivos del problema, se restringe a los que se encuentran en su vecindad. Esto facilita su construcción y su interpretación.

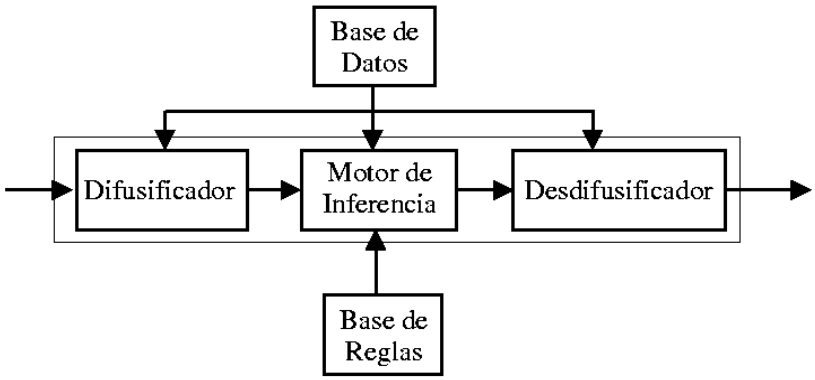

Figura B.1.: Diagrama de bloques de un modelo de inferencia difusa.

Proceso de Inferencia difusa

La Regla Composicional de Inferencia fue introducida por Zadeh en 1973 [99] como herramienta para traducir el "modus ponens" de la Lógica Clásica a la Lógica Difusa. Posteriormente ha sido formalizada y generalizada como método de inferencia, de tal manera que la formulación original surge como un caso particular.

El modus ponens, la regla básica de deducción del Cálculo de Predicados, es el método de inferencia mejor conocido y ha sido ampliamente empleado dentro del campo de la Inteligencia Artificial. De forma resumida puede describirse como sigue:

Supuesto que la implicación "Si P entonces Q" es cierta y que ocurre P (es decir, el hecho o proposición P es cierto), entonces se concluye que el hecho o proposición Q también es cierto.

$$\begin{array}{ll} P \rightarrow Q \\ P \\ \hline Q \end{array}$$

En muchos casos, P y Q contienen conocimiento acerca de variables. El caso

más simple es aquél en que P y Q son afirmaciones acerca de sendas variables, es decir, P es la proposición "x es A" y Q corresponde a "y es B", donde x e y son variables que toman valores en sendos universos U y V, no necesariamente diferentes, mientras que A y B siguen siendo propiedades sobre los valores de x e y. Ahora, a partir de la regla "Si x es A entonces y es B", y de "x es A", podemos deducir el hecho "y es B".

Desde el punto de vista del Razonamiento Aproximado, la situación que interesa es la deducción cuando la información disponible es imprecisa, incompleta o no totalmente fiable, es decir, cuando tratamos con predicados difusos. La lógica difusa proporciona un contexto apropiado para el tratamiento de la incertidumbre, porque en contraste con los sistemas lógicos tradicionales, su principal objetivo es la inferencia a partir de conocimientos, más que exactos, imprecisos. Para este caso, suele usarse el Modus Ponens Generalizado, que se establece en los siguientes términos:

$$\begin{array}{l} \text{IF } x \text{ IS } A \text{ THEN } y \text{ IS } B \\ x \text{ IS } A' \\ \hline \end{array}$$

donde, de nuevo, x e y son variables sobre U y V, pero ahora A,B y A' son conjuntos difusos (propiedades lingüísticas) de los respectivos universos de discurso, que también pueden considerarse como informaciones difusas o restricciones flexibles relativas a las mencionadas variables.

Puesto que en el caso del Modus Ponens clásico la conclusión postula que y es B, siendo B un conjunto ordinario de V, es razonable admitir que en el caso difuso la conclusión venga definida por un conjunto difuso sobre el universo de discurso de y, con lo que se deberá dar en la forma: "y es B'". Por tanto, el Modus Ponens Generalizado queda del siguiente modo:

$$\begin{array}{l} \text{IF } x \text{ IS } A \text{ THEN } y \text{ IS } B \\ x \text{ IS } A' \\ \hline \qquad\qquad y \text{ IS } B' \end{array}$$

El problema que se plantea es cómo obtener ese nuevo conjunto difuso B'. Tal como hemos comentado con anterioridad, en 1973 Zadeh, introdujo la denominada Regla Composicional de Inferencia que, intuitivamente, se puede describir y justificar del siguiente modo.

Una regla introduce una relación difusa R que liga los valores de los universos de las variables vinculadas en la regla, es decir, un conjunto difuso en el producto cartesiano de los universos de discurso $U \times V$, tal que

$$\mu_R(x,y) = F(\mu_A(x), \mu_B(y)),$$

donde μ_A y μ_B denotan las respectivas funciones de pertenencia de los conjuntos difusos A y B.

El conjunto difuso B' ha de estar engendrado, o inducido, por A' sobre y a través de R. Por tanto, puede escribirse $B' = A' \circ R$ y la cuestión, ahora es cómo construir F y $\circ$ para obtener B'. Para resolver estos problemas se han dado múltiples enfoques en la literatura. En todos ellos, no obstante, el punto de partida lo proporciona el Principio de Extensión de Zadeh que, en el contexto que tratamos, se traduce en:

$$\mu_{B'}(y) = \max_x(\mu_{A'}(x) * \mu_R(x,y)),$$

siendo $*$ una operación asociativa y monótona, no decreciente en cada argumento (una t-norma). La forma efectiva de realizar las inferencias, por tanto, descansa en la elección que se haga de F y de la t-norma $*$, obteniéndose consecuentemente, lo que podríamos denominar, distintos modos de razonar.

Modelos basados en reglas difusas

Un sistema que usa reglas difusas se denomina *Modelo Basado en Reglas Difusas* (MBRD). Estos modelos constituyen una extensión de los sistemas basados en reglas. Su principal campo de aplicación es el modelado difuso, es decir, se emplean para describir sistemas desconocidos o muy complejos. También en el campo del control han producido resultados excelentes, área en la que se conocen bajo el término de *controladores difusos*. Su aplicación a problemas de control con soluciones difíciles o imposibles para la matemática clásica, ha sido definitiva para la aceptación y rápida expansión de las técnicas basadas en la Teoría de los Conjuntos Difusos.

Los MBRD aplican espacios de dimensión n en espacios de dimensión m. Están formados por cuatro componentes, tal y como se muestra en la B.2: difusificador, base de conocimiento, motor de inferencia y desdifusificador.

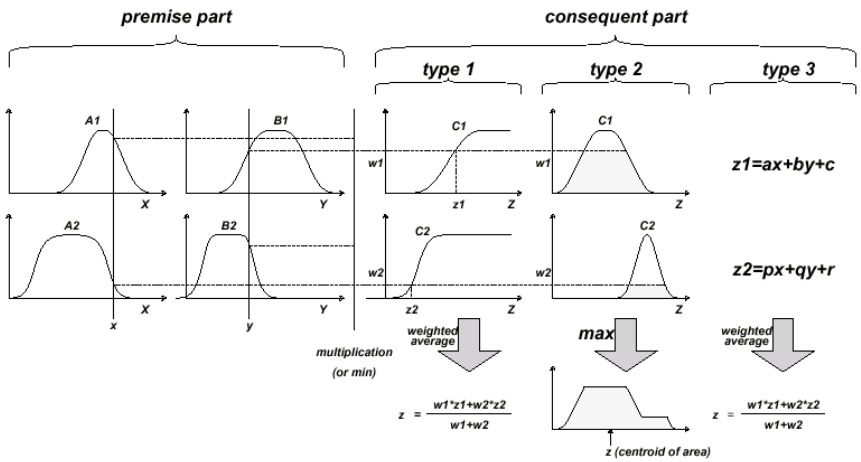

Figura B.2.: Tipos de razonamiento de los sistemas basado en reglas difusas

El difusificador convierte los valores reales de entrada en valores difusos, usualmente en conjuntos difusos univalorados o singulares (en inglés, *singleton*).

La base de conocimiento incluye la base de reglas y la base de datos. En la primera se incluyen todas las reglas que recogen el conocimiento inmerso en el sistema. Las definiciones de las funciones de pertenencia para las etiquetas de las reglas se recogen en la base de datos.

El motor de inferencia calcula la salida difusa a partir de las entradas difusas aplicando una función de implicación difusa. También agrega las salidas de las distintas reglas aplicables produciendo un único valor difuso de salida.

Finalmente, el desdifusificador condensa en un número real la salida difusa inferida.

De forma abreviada, el proceso de inferencia difuso es como sigue: Dada una entrada, se obtiene el *grado de disparo* de cada regla de la base. Este grado de disparo se obtiene a partir de los grados de emparejamiento de cada componente de la entrada con cada proposición en el antecedente de la regla difusa. Cuando las entradas son valores reales y la difusificación las transforma en conjuntos

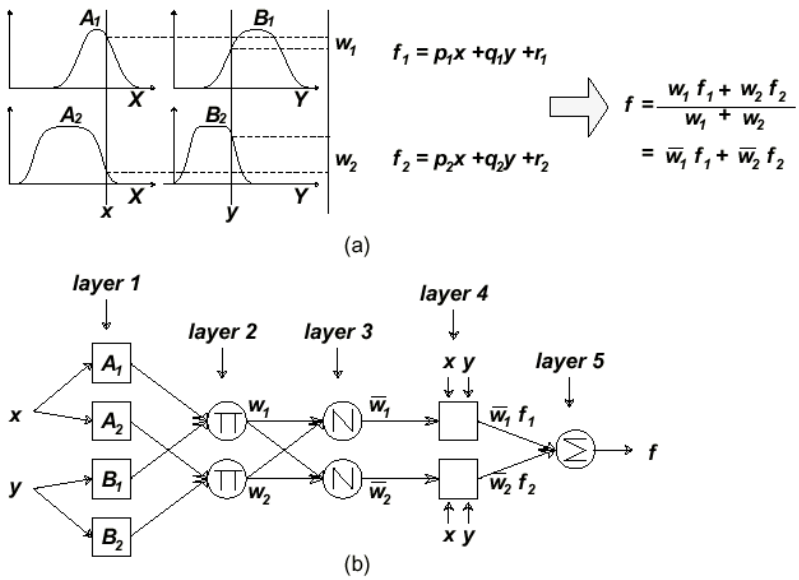

Figura B.3.: (a) A two-input TSK fuzzy model with two rules; (b) equivalent ANFIS architecture.

difusos singulares, este grado de emparejamiento no es más que el grado de pertenencia de cada valor real de la entrada al conjunto difuso definido por cada proposición de la regla. Después se agregan estos grados de emparejamiento, de acuerdo a los conectivos que enlacen las proposiciones. Lo habitual, es que sean conectivos conjuntivos, por lo que el grado de disparo de la regla será el valor de la función que modela la conjunción (mínimo, producto, etc.) aplicada a todos los grados de emparejamiento. La salida difusa de la regla será su consecuente, en el que los conjuntos difusos que aparecen tendrá un valor máximo de pertenencia igual al grado de disparo de la regla. El conjunto de reglas en la base de reglas representa conocimiento conectado según el operador *también*. La salida difusa de todas las reglas es agregada empleando la función que modela al operador *también*. Por último la salida difusa se transforma en valores reales aplicando el método de desdifusificación correspondiente.

En algunos sistemas, el orden de la agregación y desdifusificación es inverso. Es decir, se realiza una desdifusificación de las salidas difusas de cada regla y después se agregan los valores reales resultantes.

Los sistemas basados en reglas difusas poseen algunas propiedades muy interesantes, de entre las que destaca sobremanera su capacidad de "Aproximación Universal". En [13, 14, 49] se demuestra que amplias clases de controladores difusos son aproximadores universales. Este resultado habilita a los MBRDs como herramientas adecuadas para aproximar a otros sistemas aun cuando los datos asociados a los mismos no tengan carácter difuso.

Podemos distinguir dos usos fundamentales de los SBRD:

a. *Descripción lingüística* de sistemas en los que existe incertidumbre o vaguedad en las entradas y/o salidas. Se incluyen también los casos en los que no importa mucho la precisión. Lo fundamental es obtener una descripción de las relaciones entre entradas y salidas subyacentes a un sistema desconocido en términos de palabras y expresiones propias del lenguaje natural.

b. *Aproximación* de sistemas. Se explota su propiedad de Aproximadores Universales. Los MBRDs aparecen como alternativas a otros modelos matemáticos clásicos en la aproximación de sistemas desconocidos. Sus ventajas se ponen de manifiesto al tratar con sistemas complejos, donde su simplicidad y facilidad de manejo los hace preferibles a otros métodos, sobre todo en lo que respecta a eficiencia de funcionamiento.

MBRD de Mamdani

En 1975 E.H. Mamdani [61] dio la primera aplicación práctica de un MBRD. Se trataba de un MBRD simplificado dedicado a tareas de control. Este sistema dispone de difusificador, base de reglas, base de datos y desdifusificador.

Las reglas que emplea son simples, del tipo:

$$R_i: \text{IF } x_1 \text{ IS } A_1^i \text{ AND } x_2 \text{ IS } A_2^i \text{ AND } \ldots \text{AND } x_n \text{ IS } A_n^i \\ \text{THEN } y_1 \text{ IS } B_1^i \text{ AND } y_2 \text{ IS } B_2^i \text{ AND } \ldots \text{ AND } y_m \text{ IS } B_m^i. \quad \text{(B.4)}$$

Por su simplicidad y facilidad de interpretación se han convertido en el tipo estándar de regla difusa.

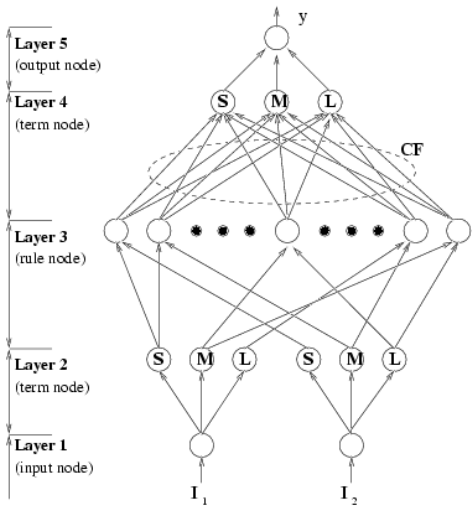

Figura B.4.: The structure of HyFIS.

Su procedimiento de inferencia se caracteriza por usar el mínimo tanto como operador de conjunción como función de implicación. Así dada la entrada $\mathbf{a} = (a_1, a_2, \ldots, a_n)$, el grado de disparo de la regla es:

$$\gamma_i = \text{mín}(A_1^i(a_1), A_2^i(a_2), \ldots, A_n^i(a_n)) \tag{B.5}$$

Las salidas difusas de la reglas son $B_1^i, B_2^i, \ldots, B_n^i$ cortadas a una altura γ_i. El operador *también* que agrega las salidas de distintas reglas se modela como una disyunción, habitualmente el operador *máximo*. Como interfaces de desdifusificación son habituales el centro de gravedad o la media de los máximos.

Por ser el más antiguo y por su simplicidad éste es el modelo de MBRD más difundido y aplicado. Además su simplicidad redunda en una implementación hardware muy fácil y barata, lo que ha favorecido su expansión a multitud de aplicaciones desde pequeños electrodomésticos a grandes dispositivos de ingeniería como grúas de contenedores o conductores automáticos de trenes.

MBRD de Takagi-Sugeno-Kang (TSK)

En 1985, Takagi, Sugeno y Kang [81, 78, 77] propusieron un modelo de MBRD que era mucho más eficaz que el de Mamdani para tareas aproximativas. Este sistema usa reglas del tipo:

$$R_i: \text{IF } x_1 \text{ IS } A_1^i \text{ AND } \ldots \text{ AND } x_n \text{ IS } A_n^i \text{ THEN } y = p_i(x_1, x_2, \ldots, x_n) \tag{B.6}$$

con $p_i(x_1, x_2, \ldots, x_n)$ una función lineal. Es decir, la salida adopta un carácter puramente funcional, que salvo en el caso base de que sea una constante, no tiene una interpretación lingüística simple.

El grado de activación de las reglas se obtiene casi igual que en los sistemas de Mamdani. Tan sólo varía en que la conjunción es modelada mediante el producto, en vez de usar el mínimo.

$$\gamma_i = \prod(A_1^i(a_1), A_2^i(a_2), \ldots, A_n^i(a_n)) \tag{B.7}$$

La salida final del sistema es:

$$y = \frac{\sum_{i=1}^{r} \gamma_i p_i(a_1, a_2, \ldots, a_n)}{\sum_{i=1}^{r} \gamma_i} \tag{B.8}$$

En general, los MBRD de Takagi-Sugeno-Kang son menos interpretables que los de Mamdani.

Modelos Difusos Aditivos

Los Modelos Difusos Aditivos (MAD) propuestos por Kosko [49] son otro tipo de MBRDs caracterizados por su forma peculiar de realizar la inferencia.

En general, un MBRD con un conjunto de reglas

$$R_i: \text{IF } x = A_i \text{ THEN } y = B_i,$$

dispara las reglas ante una entrada y cada regla produce una salida B_i'. La salida final se obtiene aplicando un operador de agregación, que habitualmente, se modela mediante el máximo. En el caso de los MADs esta agregación final se hace mediante una suma ponderada:

$$B = \sum_{i=1}^{r} w_i B_i', \tag{B.9}$$

donde w_i son pesos asociados a las reglas, no son el grado de disparo γ_i. La elección de estos pesos caracteriza el tipo de inferencia.

Cuando las reglas son del tipo TSK (B.6), la salida adopta la forma:

$$y = \sum_{i=1}^{r} w_i \gamma_i p_i(a_1, a_2, \ldots, a_n). \tag{B.10}$$

B.5.2. Modelos estadísticos para series temporales

El enfoque clásico estadístico para el análisis de series temporales tiene su referencia fundamental en los trabajos de Box y Jenkins [7], cuyo modelo autorregresivo de medias móviles (ARMA, por su nombre en inglés: *autoregressive moving average*) ha tenido una enorme popularidad desde que fue presentado. Los modelos ARMA han sido frecuentemente utilizados para modelar estructuras dinámicas lineales, para caracterizar relaciones lineales entre variables desplazadas temporalmente y para la predicción lineal.

La propuesta original de Box y Jenkins sugería la aplicación sucesiva de tres pasos, o fases: la fase de identificación, en la que se determina si la serie es estacionaria y si existe alguna componente estacional, y en la que se fija el orden de los términos autorregresivos y de medias móviles; la fase de estimación, que consiste en el ajuste del modelo previamente fijado; y la fase de validación, que, mediante el estudio de la serie de los residuos del modelo, permite determinar si el modelo construido es adecuado.

B.5.3. Modelo AR

El modelo *autorregresivo* de orden $p \geq 1$ es

$$X_t = b_1 X_{t-1} + \ldots + b_p X_{t-p} + \varepsilon_t \tag{B.11}$$

donde $\{\varepsilon_t\} \sim N(0, \sigma^2)$, conocido como *ruido blanco*. Suele escribirse $\{X_t\} \sim \mathrm{AR}(p)$ y se dice que la serie temporal $\{X_t\}$ generada mediante este modelo es un proceso $\mathrm{AR}(p)$.

El modelo (B.11) representa el estado actual de la serie, X_t, mediante los p valores inmediatamente anteriores $X_{t-1}, \ldots, X_{t-p}$ en una regresión lineal, especificando explícitamente la relación entre el valor actual y los valores pasados. Este es un modelo fácil de implementar y por eso es quizá el modelo de series temporales más utilizado.

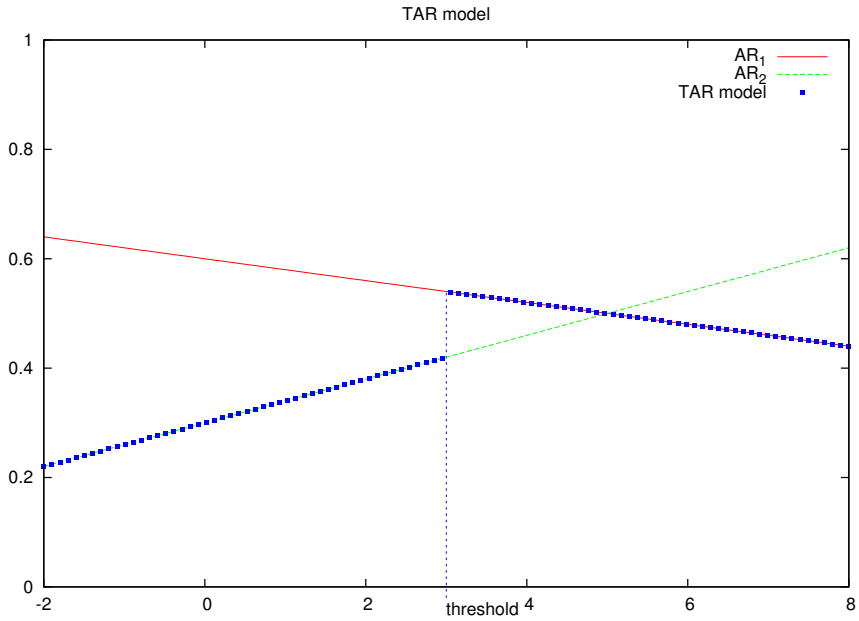

Figura B.5.: Un ejemplo del modelo TAR.

Otra alternativa clásica para el estudio de series temporales son los métodos de descomposición y suavizado. Entre ellos cabe citar el método de descomposición estacional y de tendencias basado en el suavizado *loess* [17] y el método de suavizado de Holt y Winters [94].

Modelo autorregresivo de umbral

Conocidas las limitaciones de los métodos disponibles hasta el momento, algunos investigadores propusieron modelos que permitieran capturar comportamientos no lineales de los datos. De entre estos, el primero y más conocido es la llamada autorregresión con umbral, *Threshold Autorregression* (TAR), establecida por H. Tong en 1978 [85]. Este modelo permite que una misma serie de datos sea

modelada mediante dos o más regímenes lineales adscritos a zonas disjuntas del espacio de la serie, produciéndose la transición entre uno y otro de forma súbita cuando una determinada variable (la variable de transición) supera un valor fijo (el umbral).

Un modelo TAR con k ($k \geq 2$) regímenes se define como

$$y_t = \sum_{i=1}^{k} \boldsymbol{\omega}_i \mathbf{x}_t I(s_t \in A_i) + \varepsilon_t = \sum_{i=1}^{k} \{\omega_{i,0} + \omega_{i,1} y_{t-1} + \omega_{i,p_i} y_{t-p_i} + \varepsilon_t\} I(s_t \in A_i) + \varepsilon_t, \quad \text{(B.12)}$$

donde s_t es la variable umbral, I es una función indicadora (o *función escalón*), $p_1, \ldots, p_k$ son números enteros positivos desconocidos, $\boldsymbol{\omega}_i$ son parámetros desconocidos y $\{A_i\}$ forma una partición de $(-\infty, \infty)$ con $\cup_{i=1}^{k} A_i = (-\infty, \infty)$ y $A_i \cap A_j = \emptyset, \forall i \neq j$.

Cuando la variable umbral es una de los valores desplazados de y_t, es decir, $s_t = y_{t-d}$, entonces el modelo se llama *auto-excitativo* (evite el lector connotaciones no matemáticas) y tiene el acrónimo SETAR.

Modelo autorregresivo de transiciones suaves

La característica fundamental de los modelos TAR es la forma discontínua de las relaciones autorregresivas que los componen, dada por el conjunto de umbrales. Considerando que la naturaleza es generalmente continua, Teräsvirta [83] propuso el modelo STAR (por su nombre en inglés *smooth transition autoregressive*), que se carateriza por utilizar transiciones suaves y continuas entre los distintos modelos AR, en lugar de un cambio brusco.

En los modelos STAR y sus variantes [87], se sustituye la función indicadora $I(\cdot)$ en (B.12) por una función suave con características sigmoides, como por ejemplo la función logística. Por tanto, el modelo STAR con k regímenes ($k > 2$) se define como

$$y_t = \sum_{i=1}^{k} \boldsymbol{\omega}_i \mathbf{x}_t \mathrm{F}_i(s_t; \gamma_i, c_i) + \varepsilon_t, \quad \text{(B.13)}$$

La función de transición, $\mathrm{F}(s_t; \gamma, c)$, es una función continua y acotada entre 0 y 1. En el modelo original, se asume que la variable de transición s_t es una

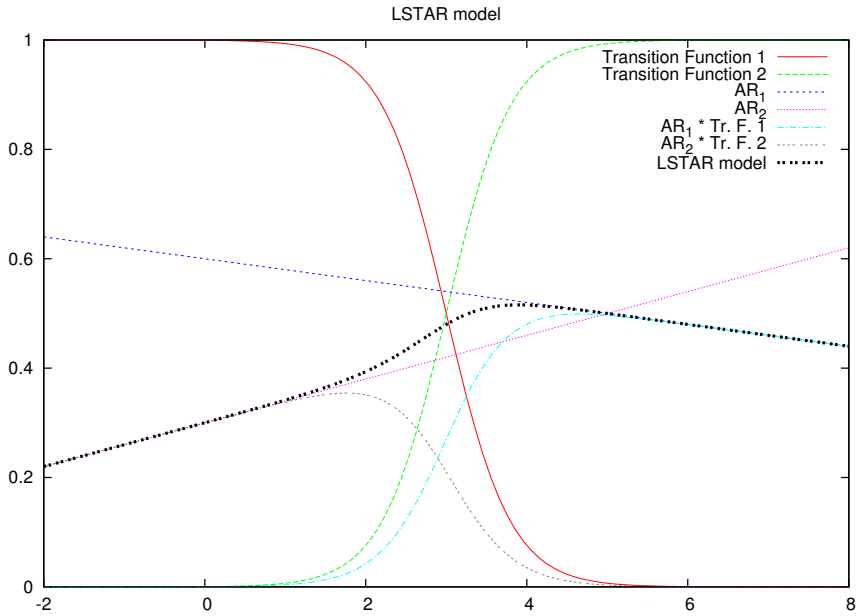

Figura B.6.: Un ejemplo del modelo STAR con 2 regímenes y función de transición logística.

variable endógena desplazada temporalmente, es decir, $s_t = y_{t-d}$ para algún número entero $d > 0$. Esto no tiene que ser así necesariamente, y la variable de transición puede ser también exógena ($s_t = z_t$), o incluso una función de variables endógenas $s_t = h(\tilde{\mathbf{x}}_t; \boldsymbol{\alpha})$ para alguna función h que depende del vector $(p \times 1)$ de parámetros $\boldsymbol{\alpha}$. Finalmente, esta variable también puede ser una tendencia temporal ($s_t = t$), lo que da lugar a un modelo con parámetros que cambian suavemente.

El régimen que prima en un momento t viene determinado por la variable s_t y el valor asociado de $\mathrm{F}(s_t; \gamma, c)$. Existen distintas alternativas para la función de transición las cuales dan lugar a distintos tipos de comportamiento. Una elección

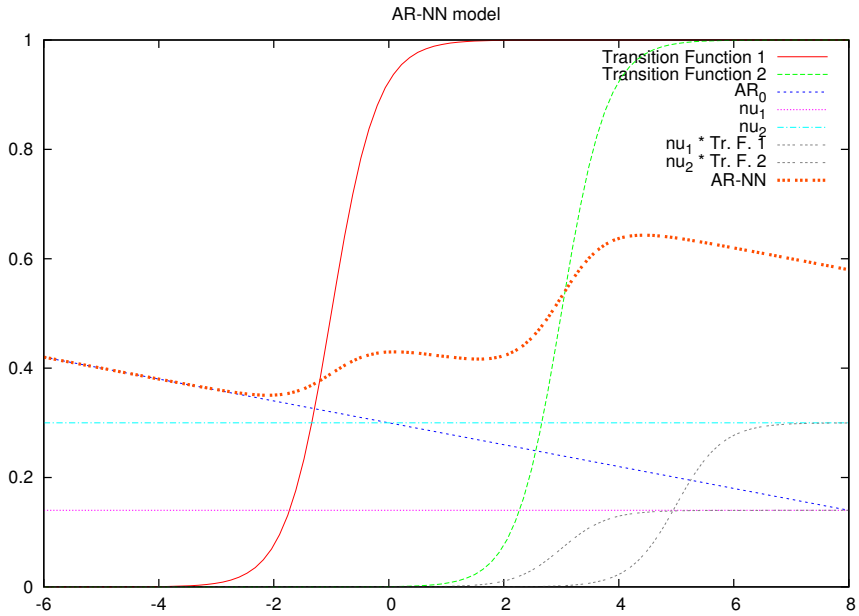

Figura B.7.: Un ejemplo del modelo AR-NN con dos neuronas en la capa oculta.

común para $F(s_t;\gamma,c)$ es la función logística de primer orden, equación (3.11), página 45, dando lugar a un modelo llamado STAR logístico (LSTAR).

Red neuronal autorregresiva

Después del éxito de las Redes Neuronales Artificiales en tantos campos distintos, incluyendo las series temporales, algunos investigadores [84] comenzaron a considerarlas como modelos estadísticos no lineales, y a aplicar la inferencia estadística al problema de la especificación del modelo. Desarrollaron una estrategia incremental que permite realizar correctamente la inferencia, además de la evaluación del modelo.

La red neuronal autorregresiva de una capa oculta se define como

$$y_t = \underbrace{\boldsymbol{\omega}_1 \mathbf{x}_t}_{AR} + \underbrace{\sum_{i=2}^{k} \nu_i f(\boldsymbol{\omega}_i \mathbf{x}_t)}_{NN} + \varepsilon_t \tag{B.14}$$

donde los ν_i son llamados "intensidades de conexión", la función $f(\cdot)$ "neurona oculta" y suele ser logística. Pese a que en el campo de la Computación Flexible se suele asumir que $\boldsymbol{\omega}_1 = 0$, el modelo AR-NN incluye esta "neurona lineal".

La interpretación geométrica de este modelo considera que la AR-NN divide el espacio euclidiano p-dimensional mediante k hiperplanos dados por $\boldsymbol{\omega}_i \mathbf{x}_t$, lo que resulta en varias regiones poliédricas. Calcula la salida como la suma de la contribución de cada hiper-región modulada por una función de suavizado $f(\cdot)$.

Red neuronal generalizada local-global

Otro enfoque estadístico a las redes neuronales es la red generalizada local-global (LGNN por sus siglas en inglés). La idea fundamental del modelo LGNN es utilizar una definición por partes para expresar el mapa de entrada-salida. La salida de la red está compuesta por una combinación de pares, cada uno de ellos constituído por una función de aproximación y por una función de activación. Las funciones de activación son equivalentes a las funciones de transición o neuronas ocultas mencionadas previamente, y definen el rol de la función de aproximación asociada a ellas para cada subconjunto del dominio. Se permite la superposición parcial de funciones de activación. De esta forma, el problema de la aproximación de una función se resuelve mediante la especialización de neuronas en cada uno de los sectores del dominio. En otras palabras, las neuronas están formadas por pares de funciones de activación y aproximación, que tratan de emular la función objetivo en zonas distintas del dominio.

El modelo LGNN se define por tanto como

$$y_t = \sum_{i=1}^{k} \mathrm{L}(\mathbf{z}_t; \boldsymbol{\psi}_{L_i}) \mathrm{B}(\mathbf{z}_t; \boldsymbol{\psi}_{B_i}) + \varepsilon_t \tag{B.15}$$

donde $\mathbf{z}_t$ es un vector de variables desplazadas de y_t y variables exógenas, y las funciones $\mathrm{L}(\mathbf{z}_t; \boldsymbol{\psi}_{L_i}) : \mathbb{R}^p \to \mathbb{R}$ y $\mathrm{B}(\mathbf{z}_t; \boldsymbol{\psi}_{B_i}) : \mathbb{R}^p \to \mathbb{R}$ son las funciones de aproximación y de activación respectivamente.

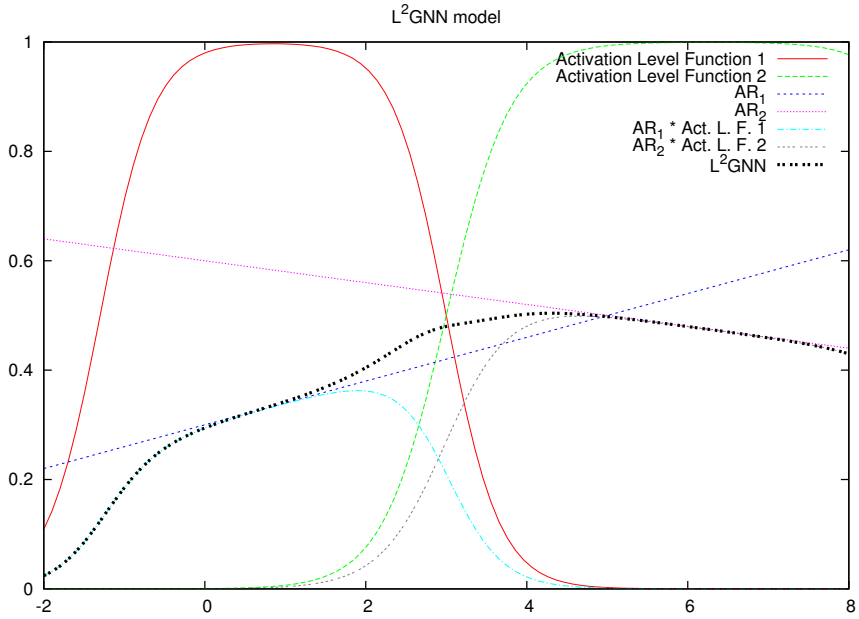

Figura B.8.: Un ejemplo del modelo L^{2}GNN con dos neuronas en la capa oculta.

En su formulación original, $\mathrm{B}(\mathbf{z}_t;\boldsymbol{\psi}_{B_i})$ se define

$$\mathrm{B}(\mathbf{z}_t;\boldsymbol{\psi}_{B_i}) = -\left[\frac{1}{1+\exp\left(\gamma(\boldsymbol{\varphi}\mathbf{z}_t-\beta^{(1)})\right)} - \frac{1}{1+\exp\left(\gamma(\boldsymbol{\varphi}\mathbf{z}_t-\beta^{(2)})\right)}\right] \quad \text{(B.16)}$$

donde $\boldsymbol{\psi}_{B_i} = \left(\gamma, \boldsymbol{\varphi}, \beta^{(1)}, \beta^{(2)}\right)$.

Un caso especial del modelo LGNN es la red neuronal local-global lineal, (L^{2}GNN) [76]. En este caso, las funciones de aproximación son lineales, es decir, $\mathrm{L}(\mathbf{z}_t;\boldsymbol{\psi}_{L_i}) = \boldsymbol{\omega}_i\mathbf{z}_t$, con $\boldsymbol{\omega}_i = (\omega_{i1},\omega_{i2},\ldots,\omega_{ip})' \in \mathbb{R}^p$. Por tanto, el L^{2}GNN puede

escribirse

$$y_t = \sum_{i=1}^{k} \boldsymbol{\omega}_i \mathbf{z}_t \mathrm{B}(\mathbf{z}_t; \boldsymbol{\psi}_{B_i}) + \varepsilon_t \tag{B.17}$$

y el proceso estocástico resultante consiste en una mezcla de procesos lineales.

Modelo autorregresivo de transiciones suaves con coeficientes neuronales

Uno de las últimas propuestas en el ámbito de los modelos basados en umbrales es el modelo STAR con coeficientes neuronales. Este modelo es una generalización de algunos de los expuestos en este capítulo, y puede incluír múltiples regímenes y múltiples variables de transición. Puede ser visto como un modelo lineal cuyos parámetros cambian en el tiempo y son dinámicamente determinados por una red neuronal de propagación hacia adelante con una única capa oculta.

Consideremos un modelo lineal cuyos coeficientes cambien en el tiempo:

$$y_t = \boldsymbol{\phi}_t' \mathbf{x}_t + \varepsilon_t, \tag{B.18}$$

donde $\boldsymbol{\phi}_t = (\phi_t^{(0)}, \phi_t^{(1)}, \ldots, \phi_t^{(p)})' \in \mathbb{R}^{p+1}$ es un vector que contiene los coeficientes del modelo. La evolución temporal de los $\phi_t^{(j)}$ de (B.18) viene dada por la salida de una red neuronal con k unidades ocultas:

$$\phi_t^{(j)} = \sum_{i=1}^{k} \nu_{ji} f(\boldsymbol{\omega}_i \mathbf{z}_t) - \nu_{j0}, \quad j = 0, \cdots, p, \tag{B.19}$$

en la que ν_{ji} y ν_{j0} son coeficientes reales.

Sustituyendo las p realizaciones de (B.19) en (B.18), obtenemos la forma general del modelo NCSTAR:

$$y_t = \boldsymbol{\nu}_1 \mathbf{x}_t + \sum_{i=2}^{k} \boldsymbol{\nu}_i \mathbf{x}_t f(\boldsymbol{\omega}_i \mathbf{z}_t) + \varepsilon_t, \tag{B.20}$$

en la que $\mathbf{z}_t$ es un vector $q \times 1$ de variables de transición y $\boldsymbol{\omega}_i = [\omega_{1i}, \ldots, \omega_{qi}]'$ son parámetros reales. A la norma de $\boldsymbol{\omega}_i$, llamada γ_i, se la conoce también como el parámetro de pendiente, y, en el caso límite en que la pendiente se acerca a infinito, la función logística se convierte en una función escalón.

B.5.4. Relaciones entre modelos

Una vez que los modelos considerados han sido ya expuestos, podemos dar paso al resumen de las principales aportaciones de esta tesis. La idea que da origen a todo el trabajo surge al observar la expresión del modelo AR, ecuación (B.11), y la de la regla difusa de tipo TSK, ecuación (B.6). Resulta fácil probar[4] que

Proposition B.5.1. [pág. 55] *Al ser empleada para modelar series temporales, una regla difusa de tipo TSK puede ser vista como un modelo AR local que se aplica en un subconjunto del espacio de estados definido por el antecedente de la regla.*

Un ejemplo de esta relación vendría dado por el siguiente modelo AR:

$$y_t = 2{,}1 + 0{,}01 y_{t-1} - 0{,}1 y_{t-2} + \varepsilon_t, \tag{B.21}$$

el cual puede ser visto como la definición de la relación entre la variable "de salida" y_t y las variables "de entrada" y_{t-1} y y_{t-2}. Esta relación puede ser vista gráficamente en la Figura B.9 (a).

Podríamos construir una regla difusa cuyo consecuente fuera igual al anterior modelo AR:

$$\text{IF } y_{t-2} \text{ IS } A_1 \text{ AND } y_{t-1} \text{ IS } A_2 \quad \text{THEN } y_t = 2{,}1 + 0{,}01 y_{t-1} - 0{,}1 y_{t-2} + \varepsilon_t, \tag{B.22}$$

en donde, ignorando constantes multiplicativas,

$$A_i(x) = \exp\left(\frac{-(x-\mu_i)^2}{2\sigma_i^2}\right) \quad i = 1,2 \tag{B.23}$$

son las funciones de pertenencia de las variables del antecedente de la regla. Una representación gráfica de esto puede verse en la Figura B.9 (b), en la que $\mu_1 = \mu_2 = 2{,}5$ y $\sigma_1 = \sigma_2 = 2{,}0$.

En dicha representación gráfica queda claro que la aplicación de la regla difusa se reduce a la aplicación del modelo AR en el subconjunto del espacio definido por la pertenencia de las variables "de entrada" a las funciones de pertenencia

[4] No se incluyen aquí las demostraciones de proposiciones y teoremas, asumiendo que el lector interesado lo estará hasta tal punto que no le resulte un problema leerlas en inglés.

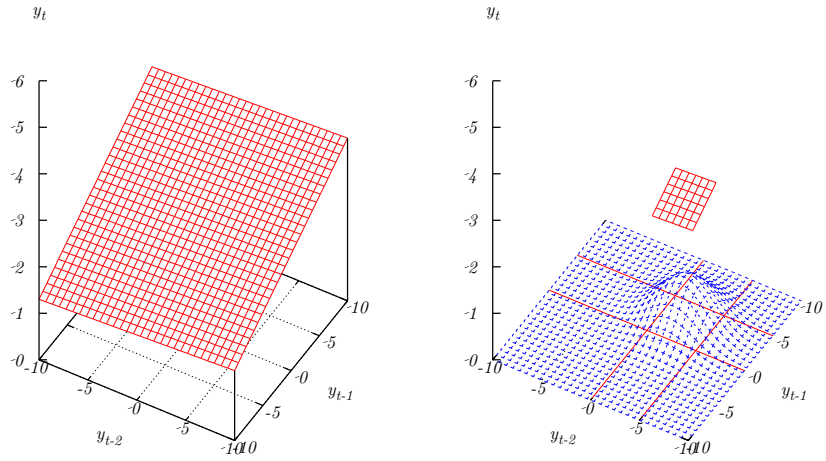

Figura B.9.: (a) Plano definido por el modelo AR definido en (4.1). (b) Representación gráfica de la regla difusa que contiene a dicho modelo AR.

definidas en el antecedente. Debe aclararse que dicho subconjunto es difuso, y por tanto sus límites no son abruptos.

La conexión entre estos dos elementos, que son básicos en sus respectivos ámbitos, implica la posibilidad de estudiar las relaciones entre dos áreas distintas en el contexto del análisis de series temporales: la Computación Flexible y el enfoque estadístico tradicional.

Por un lado, hemos visto que los modelos AR son buenos modelos lineales aplicables a problemas de predicción. También, sabemos que un modelo STAR es básicamente un conjunto de modelos AR locales, y que la no linealidad tiene cabida en él. Por otro lado, hemos visto cómo una regla difusa se relaciona con un

modelo AR. Sabiendo que los modelos de inferencia difusa están compuestos por conjuntos de reglas difusas, podría interesarnos considerar la relación existente entre la familia de los modelos de umbral y la familia de modelos basados en reglas difusas.

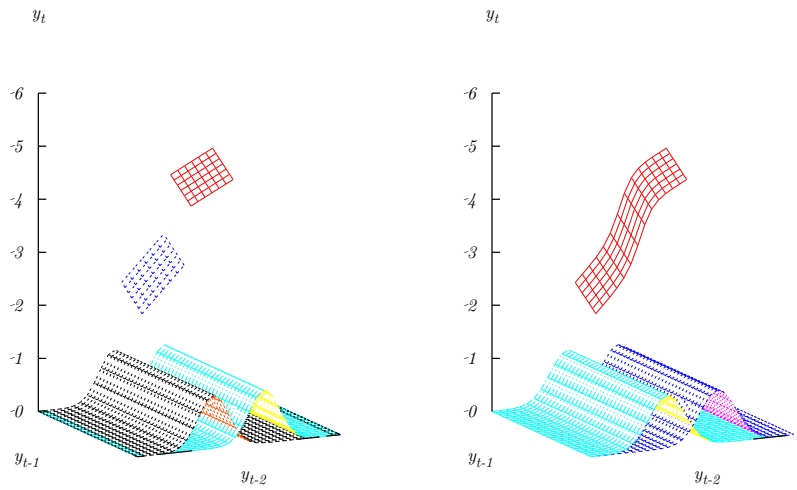

Figura B.10.: (a) Dos modelos AR locales (o dos reglas difusas) (b) El modelo STAR (o el modelo de inferencia difuso) derivado de los dos AR (o reglas) mostrados en (a).

Es algo obvio que existen algunos paralelismos entre ambas familias de modelos. De una forma abstracta, ambos modelos están compuestos por un conjunto de elementos (AR y reglas difusas) que han resultado estar estrechamente relacionados. Más concretamente, ambas familias de modelos encuentran su fundamento en la construcción de una hiper-superficie en el espacio de estados que

trata de modelar la relación que liga las variables de una serie temporal. Además, ambas definen esta hiper-superficie como composición de hiper-planos que son de aplicación únicamente en ciertas zonas del espacio. La Figura B.10 muestra esto con cierta claridad de forma gráfica.

De hecho, es fácil probar la siguiente

Proposition B.5.2. [pág. 59] *El modelo STAR es funcionalmente equivalente a un MBRD aditivo de tipo TSK con un único término en el antecedente de las reglas.*

De esta forma, es posible avanzar en el estudio de las relaciones entre los modelos basados en reglas difusas y algunos otros modelos autorregresivos de umbral expuestos previamente. Así, se demuestran las siguientes relaciones:

Proposition B.5.3. [pág. 60] *La red neuronal autorregresiva (AR-NN) es funcionalmente equivalente a un MBRD TSK con una regla por defecto (de aplicación universal).*

Theorem B.5.4. [pág. 61] *La red neuronal local-global es una generalización de los MBRD aditivos TSK.*

Proposition B.5.5. [pág. 62] *La red neuronal local-global lineal (L^2GNN) es funcionalmente equivalente a un MBRD aditivo de tipo TSK.*

Proposition B.5.6. [pág. 63] *El modelo STAR con coeficientes neuronales (NCSTAR) es funcionalmente equivalente a un MBRD aditivo de tipo TSK.*

Todo esto puede quedar sintetizado en el siguiente

Theorem B.5.7. *El MBRD TSK es una generalización de los modelos de umbral TAR, STAR, AR-NN, L^2GNN y NCSTAR.*

B.5.5. Enfoque estadístico para modelos basados en reglas difusas

Como fue expuesto anteriormente, una de las objecciones fundamentales que científicos con formación estadística clásica han formulado contra los modelos de la Computación Flexible en general, y los MBRD y las redes neuronales en

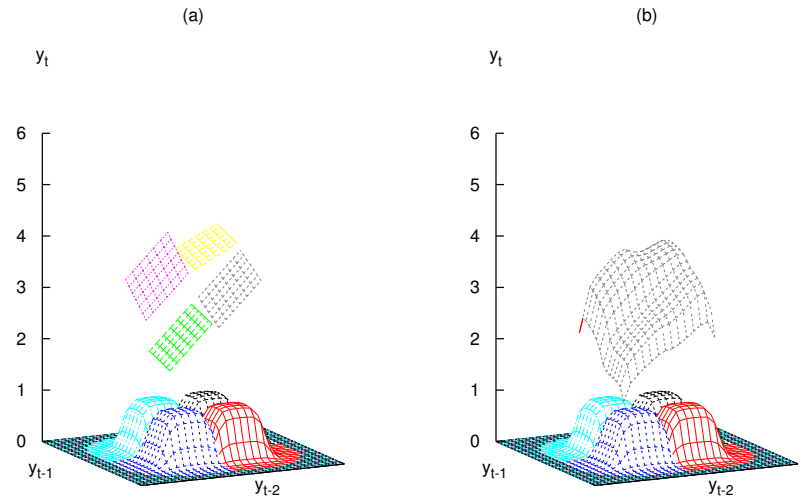

Figura B.11.: (a) Cuatro modelos AR locales (o reglas difusas) (b) El modelo L^2GNN model (o de inferencia difuso) derivado de ellas.

particular, es la falta de una teoría sólida que los soporte. El hecho de no existir medios para probar *a priori* si los modelos poseen o no buenas propiedades estadísticas (algo estrechamente relacionado con su famosa condición de *cajas negras*) impidió que fueran aceptados por amplias partes de la comunidad científica, a pesar de los buenos resultados que se obtenían en situaciones prácticas. La actitud de investigadores y usuarios de la Computación Flexible hacia esto ha sido usualmente la de trabajar desde un punto de vista ingenieril, y extender las aplicaciones prácticas de modelos y métodos en la esperanza de que los beneficios empíricos fueran en algún momento suficientemente buenos como para convencer finalmente a los escépticos.

Los resultados expuestos brevemente en la sección anterior (y con más profundidad en el Capítulo 4) tienen un impacto inmediato sobre esta cuestión, ya que permiten la derivación de un enfoque estadístico para una familia de modelos de Computación Flexible, los MBRD, al considerarlos modelos no lineales para series temporales.

Esto incluye pruebas *a priori* de sus propiedades estadísticas, como la estacionariedad o la identificabilidad, lo cual puede arrojar más luz sobre su comportamiento interno. La estacionariedad es una de las cuestiones centrales en la teoría de series temporales. Un modelo es estacionario si la estructura probabilística de la serie que genera es constante a lo largo del tiempo, o al menos asintóticamente constante (cuando no se inicia en equilibrio). Por otro lado, la identificabilidad, entendida como la unicidad y minimalidad de la especificación de un modelo, es también crucial para la inferencia estadística, ya que no es posible derivar ningún test estadístico si no se garantiza que el modelo es identificable.

Además, el uso de ajustes basados en máxima verosimilitud nos permite también garantizar la existencia, la convergencia, la consistencia y la normalidad asintótica de los estimadores. Esto es un requisito básico para que un modelo sea aceptado.

Más aun, los tests de linealidad nos proporcionan la posibilidad de decidir, basándonos en los datos, si una serie puede ser modelada con un único modelo autorregresivo lineal o si un MBRD podría ser más apropiado. Si entendemos los MBRD como modelos de regresión no lineal, los procedimientos estándar para comprobar la significancia de los parámetros, como los tests LM, deberían ser aplicables en principio. Para aplicar estos tests, sin embargo, es imprescindible conocer la distribución de los parámetros, de ahí la importancia del ajuste por máxima verosimilitud mencionado anteriormente. De cualquier manera, en la literatura difusa no se ha prestado hasta el momento mucha atención a los tests de hipótesis. En tanto que es obvio que una serie lineal debería ser modelada mediante un modelo lineal, es decir, una única regla, hasta donde sabemos no hay ningún procedimiento de tests que permita evitar el error de utilizar modelos complejos para resolver problemas simples.

Estos tests van a permitirnos por otro lado decidir iterativamente si un modelo tiene un número suficiente de reglas para capturar la dinámica de los datos o si, por el contrario, es necesario incluir más complejidad en el modelo en forma

de nuevas reglas. Esta decisión es crucial a la hora de desarrollar una estrategia incremental de construcción de MBRD que resulta en modelos más parsimoniosos, con una base de reglas cuya complejidad se adapta a la de los datos.

Finalmente, los tests estadísticos de diagnóstico sobre la serie de los residuos nos proporcionan la posibilidad de determinar si un modelo está capturando las propiedades intrínsecas de un conjunto de datos o no. Por ejemplo, es interesante obtener un conocimiento preciso acerca de la serie de los residuos del modelo, determinando si sus valores son independientes y normalmente distribuidos. Si los residuos no fueran independientes, eso implicaría que el modelo fracasa a la hora de capturar una parte importante del comportamiento de la serie, y por tanto debería de ser reespecificado.

Otra propiedad deseable que debería tener el modelo se refiere a la varianza de la serie de residuos. Si un modelo captura adecuadamente el comportamiento intrínseco de una serie, los residuos han de tener la misma varianza en cualquier punto de la serie. Si esto no se da, implica que la precisión del modelo depende del tiempo, y por tanto que hay partes del espacio de estados que no son modeladas adecuadamente.

Y hay una tercera alternativa para comprobar el buen funcionamiento de un modelo que se refiere a sus parámetros, que deberían ser constantes a lo largo del tiempo. Si un modelo está correctamente especificado, sus parámetros deben ser los mismos en cualquier punto de la serie, ya que parámetros cambiantes indicarían que el sistema entra en regímenes que no son considerados por el modelo.

Para terminar, decidimos reunir todas estas aportaciones en una propuesta única consistente en un ciclo de modelado híbrido para series temporales mediante modelos basados en reglas difusas. En el Algoritmo 3 se resume esta propuesta.

B.5.6. Experimentos y aplicaciones

Una vez que el ciclo híbrido de modelado para MBRD estuvo establecido, era de interés comprobar su validez empírica mediante su aplicación efectiva a algunos problemas. Esta comprobación fue realizada mediante dos tipos de experimentos: por un lado se emplearon conjuntos de datos artificialmente generados (ex-

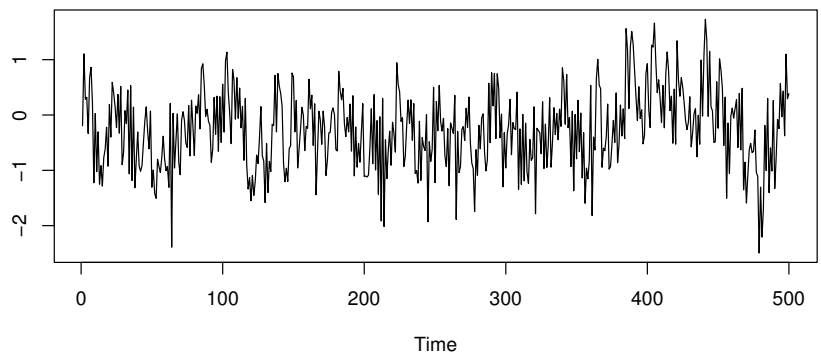

Figura B.12.: Generated series for Experiment 3, three regime STAR.

perimentos de Montecarlo) y por otro se abordó el análisis de series provenientes de situaciones reales.

Experimentos de Montecarlo

No ha sido hasta hace relativamente poco que el uso de conjuntos de datos sintéticos ha sido estudiado en el ámbito de la Computación Flexible [5]. Sin embargo, en el mundo estadístico es una práctica común el usar este tipo de experimentos para comprobar las capacidades de modelado de las propuestas.

La hipótesis básica es considerar que cualquier serie temporal proviene de un proceso generador de datos (PGD), usualmente desconocido, al que se añade una componente de ruido aleatorio. Inversamente, para generar una serie temporal artificial, necesitamos definir un PGD y una distribución aleatoria de ruido, cuya suma producirá los datos iterativamente. Así, esta serie temporal podría ser estudiada bajo el esquema de modelado elegido, identificando y ajustando un modelo para ella. Si los parámetros de este modelo resultan ser (o, al menos, tienden a ser) iguales a los parámetros originales del PGD, obtendríamos una evidencia clara de que el esquema de modelado es correcto.

Por ejemplo, la Figura B.12 muestra una realización de la serie temporal uti-

lizada en uno de los experimentos realizados (Experimento 3, para más detalles consultar la Sección 6.2.3, en la página 110) correspondiente a un modelo STAR de tres regímenes, dado por:

$$\begin{aligned} y_t = -0{,}1 + 0{,}3y_{t-1} + 0{,}2y_{t-2} + \\ (-1{,}2y_{t-1} + 0{,}5y_{t-2}) \times \mu_S(\mathbf{x}_t; \boldsymbol{\psi}_1) + \\ (1{,}8y_{t-1} - 1{,}2y_{t-2}) \times \mu_S(\mathbf{x}_t; \boldsymbol{\psi}_2) + \varepsilon_t, \\ \varepsilon_t \sim NID(0{,}0{,}5^2) \end{aligned} \quad \text{(B.24)}$$

donde los parámetros no lineales son $\boldsymbol{\psi}_1 = [\gamma_1, \boldsymbol{\omega}_1, c_1] = [20, (1,0), -0{,}6]$ y $\boldsymbol{\psi}_2 = [\gamma_2, \boldsymbol{\omega}_2, c_2] = [20, (1,0), 0{,}6]$.

Este modelo tiene tres regímenes límite, de los cuales el "inferior" corresponde a $\mu_S(\mathbf{x}_t; \boldsymbol{\psi}_1) = \mu_S(\mathbf{x}_t; \boldsymbol{\psi}_2) = 0$ y es estacionario, el régimen "medio" tiene $\mu_S(\mathbf{x}_t; \boldsymbol{\psi}_1) = 1$ y $\mu_S(\mathbf{x}_t; \boldsymbol{\psi}_2) = 0$ y es explosivo, mientras que el régimen "superior" está sucede cuando $\mu_S(\mathbf{x}_t; \boldsymbol{\psi}_1) = \mu_S(\mathbf{x}_t; \boldsymbol{\psi}_2) = 1$ y también es explosivo. En su comportamiento a largo plazo, este modelo tiene un ciclo límite con un periodo de 8 unidades.

Al aplicar el test de linealidad a 500 realizaciones de esta serie artificial, se encontró que el test funcionaba correctamente, determinando la no linealidad en la totalidad de las series. Una vez determinada dicha no linealidad, se aplicó el proceso iterativo de construcción, el cual paró en el número correcto de reglas (tres) en el 90 % de los casos.

El Cuadro B.1 muestra los resultados del ajuste doble, consistente en la aplicación de algunas metaheurísticas al modelo una vez que este ya había sido ajustado mediante el proceso iterativo de construcción. Sólo se muestran los valores para los parámetros no lineales, que son los que realmente se ajustan por el método de máxima verosimilitud.

De la observación de dicho cuadro queda claro que las metaheurísticas son capaces de mejorar significativamente los resultados obtenidos previamente por el algoritmo estándar. En concreto, éste tendió a fallar a la hora de encontrar los parámetros no lineales correspondientes al primer régimen no lineal, γ_1 y c_1. De esta forma, había margen para que las metaheurísticas intentaran mejorar, lo que a la luz de los datos consiguieron ampliamente.

El enfriamiento simulado (SANN) obtiene buenas estimaciones, pero su desviación media con respecto a la mediana es bastante alta si la comparamos con

el valor correspondiente de los algoritmos genéticos. Éstos consiguen resolver adecuadamente el problema, obteniendo valores relativamente bajos de dicha desviación media, lo cual justifica su inclusión en el ciclo de modelado.

Análisis de problemas reales

Una vez que los experimentos de Montecarlo demostraron que el ciclo de modelado estaba correctamente planteado, era muy importante comprobar su funcionamiento en situaciones reales, en lo que constituye la prueba de fuego de cualquier propuesta de este tipo.

Mostraremos aquí los resultados obtenidos en una de las series obtenidas, la que proviene de las antiguas hojas de cuentas de ciertas empresas peleteras del distrito del río Mackenzie, en el noroeste de Canadá. Esta serie contabiliza el número de linces capturados mensualmente en el periodo que va entre el año 1821 y el año 1934, y suele estudiarse su transformación logarítmica. La Figura B.13 muestra la serie original y la serie transformada.

A su vez, en la Figura B.14 podemos ver el histograma de la serie transfor-

Cuadro B.1.: Resultados del ajuste para el Experimento 3.

		Algoritmo			
Parámetro	Valor	BFGS	SANN	GA	GAD
γ_1	20.0	10.8616 (10.9488)	20.6136 (19.4400)	21.4366 (13.3229)	21.1886 (14.4303)
γ_2	20.0	17.0280 (18.2518)	28.4699 (12.9456)	20.3515 (7.7956)	21.3101 (9.8493)
c_1	-0.6	-0.2536 (0.5531)	-0.5770 (0.0731)	-0.5948 (0.0520)	-0.5973 (0.0540)
c_2	0.6	0.5819 (0.0703)	0.6001 (0.0355)	0.5972 (0.0365)	0.5981 (0.0361)
ω_{11}	1.0	0.9970 (0.0043)	0.9962 (0.0041)	0.9985 (0.0020)	0.9985 (0.0021)
ω_{12}	0.0	-0.0328 (0.1202)	-0.0864 (0.1215)	-0.0009 (0.0797)	-0.0033 (0.0789)
ω_{21}	1.0	0.9992 (0.0012)	0.9999 (0.0006)	0.9995 (0.0006)	0.9996 (0.0006)
ω_{22}	0.0	0.0167 (0.0635)	0.0011 (0.0044)	-0.00008 (0.0440)	0.0008 (0.0435)

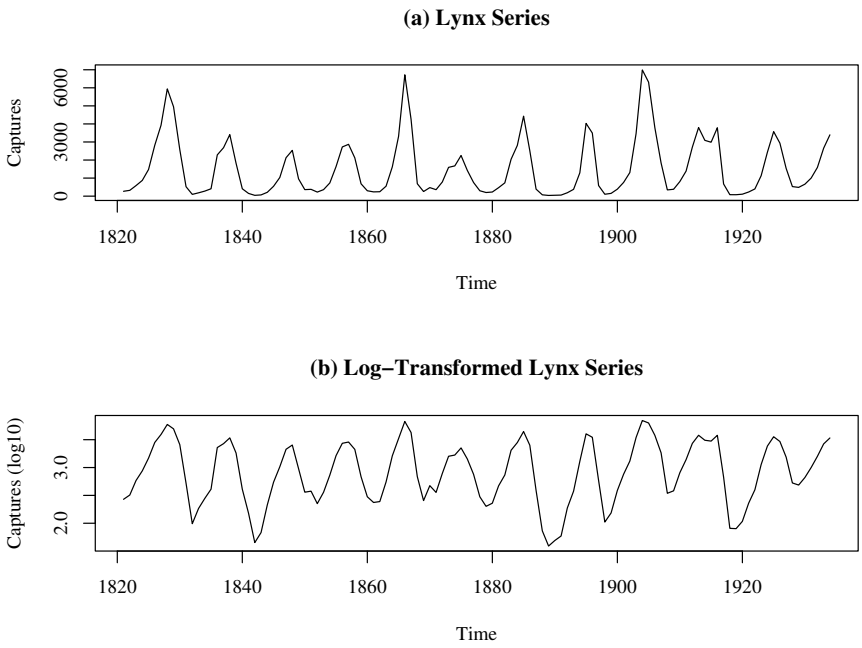

Figura B.13.: Número de linces capturados en el distrito del río Mackenzie, en el noroeste canadiense, entre los años 1821 y 1934.

mada, el cual presenta cierta bimodalidad. Calculamos su kurtosis, −0,773, y su sesgo, −0,357. En la misma figura se encuentran las funciones de autocorrelación (ACF) y de autocorrelación parcial (PACF). La función ACF muestra un comportamiento cíclico con un periodo de alrededor de 5 meses, mientras que la función PACF exhibe una autocorrelación significativa entre los dos primeros valores. Dado que este conjunto de datos es estándar en la literatura, dimos por terminado el análisis descriptivo del mismo.

Para la elección de la estructura del modelo, nos basamos también en estudios previos, que en su mayor parte sugieren la elección del orden 2 para los modelos lineales.

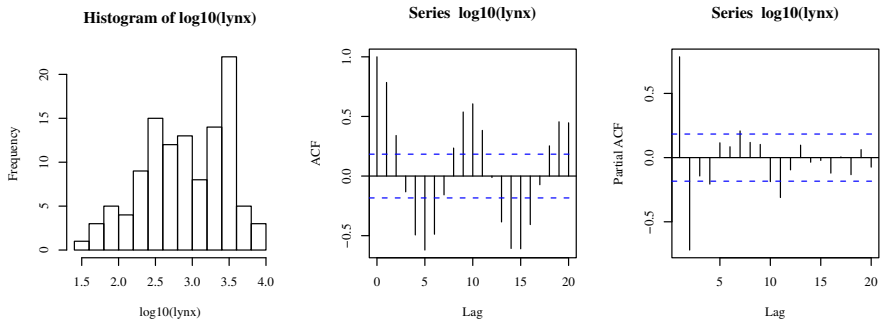

Figura B.14.: Histograma y autocorrelogramas de la serie de capturas de lince transformada.

El test de linealidad contra un modelo NCSTAR con función de transición sigmoide arrojó un valor p de 0,000259, mientras que el test contra un modelo basado en funciones gaussianas obtuvo un valor p de 0,000115. Ambos tests indican que la serie es no lineal y sugieren el uso de modelos avanzados.

En ambos casos, el ciclo de modelado terminó cuando el segundo régimen había sido añadido, con lo que ambos modelos contienen dos regímenes, dados por:

$$y_t = 0{,}9599 + 1{,}2514y_{t-1} - 0{,}3398y_{t-2} + (2{,}5466 + 0{,}3764y_{t-1} - 0{,}7973y_{t-2})\mu_{\mathsf{S}}(\mathbf{x}_t;\boldsymbol{\psi}_{\mathsf{S}}) + \varepsilon_t \quad \text{(B.25)}$$

en el caso de función de pertenencia sigmoide, con $\boldsymbol{\psi}_{\mathsf{S}} = (\gamma, \boldsymbol{\omega}, c) = (103{,}1266, [0{,}4630, 0{,}8863], 9{,}4274)$, y

$$y_t = 0{,}8749 + 1{,}2302y_{t-1} - 0{,}3074y_{t-2} + (2{,}0084 + 0{,}2961y_{t-1} - 0{,}6486y_{t-2})\mu_{\mathsf{G}}(\mathbf{x}_t;\boldsymbol{\psi}_{\mathsf{G}}) + \varepsilon_t \quad \text{(B.26)}$$

en el caso gaussiano, donde $\boldsymbol{\psi}_{\mathsf{G}} = (\gamma, \mathbf{c}) = (11{,}0129, [5{,}8417, 3{,}6653])$.

El primer modelo obtuvo una desviación estándar residual de $\hat{\sigma}_{\varepsilon,\mathsf{S}} = 0{,}196$, mientras que el segundo obtuvo un valor de $\hat{\sigma}_{\varepsilon,\mathsf{G}} = 0{,}207$. Los valores obtenidos para el criterio de información de Akaike (AIC) fueron $\text{AIC}_{\mathsf{S}} = -314$ y $\text{AIC}_{\mathsf{G}} = -306$ respectivamente, mientras que el porcentaje medio de la mediana del error (MAPE) fue del $\text{MAPE}_{\mathsf{S}} = 5{,}94\%$ y $\text{MAPE}_{\mathsf{G}} = 6{,}31\%$.

Una vez que ambos modelos fueron estimados mediante el procedimento estándar, aplicamos una metaheurística para reajustar los parámetros. Teniendo en cuenta los resultados obtenidos en la Sección 6.2, en la que los algoritmos bioinspirados tendieron a obtener los mejores resultados, decidimos utilizar tan sólo el algoritmo genético en esta ocasión.

Una vez que los parámetros fueron refinados, los modelos resultaron ser estos:

$$y_t = 0{,}3978 + 1{,}2560 y_{t-1} - 0{,}3359 y_{t-2} + (1{,}0193 + 0{,}3744 y_{t-1} - 0{,}7736 y_{t-2}) \mu_S(\mathbf{x}_t; \boldsymbol{\psi}_S) + \varepsilon_t \quad \text{(B.27)}$$

en el caso de función de pertenencia sigmoide, con $\boldsymbol{\psi}_S = (\gamma, \boldsymbol{\omega}, c) = (38{,}9935, [0{,}4969, 0{,}8678], 4{,}1306)$, y

$$y_t = 0{,}4023 + 1{,}2224 y_{t-1} - 0{,}3103 y_{t-2} + (0{,}8099 + 0{,}3751 y_{t-1} - 0{,}7074 y_{t-2}) \mu_G(\mathbf{x}_t; \boldsymbol{\psi}_G) + \varepsilon_t \quad \text{(B.28)}$$

en el caso gaussiano, donde $\boldsymbol{\psi}_G = (\gamma, \mathbf{c}) = (10{,}000, [2{,}576, 6{,}831])$.

Para estos modelos reajustados, la desviación estándar residual obtenida fue $\hat{\sigma}_\varepsilon = 0{,}191$ y $\hat{\sigma}_\varepsilon = 0{,}205$ respectivamente. Los valores obtenidos para el AIC fueron $\text{AIC}_S = -313$ y $\text{AIC}_G = -307$ respectivamente, mientras que el porcentaje medio de la mediana del error fue del $\text{MAPE}_S = 5{,}90\,\%$ y del $\text{MAPE}_G = 6{,}26\,\%$.

En este momento, pudimos considerar que la construcción de nuestros modelos había concluido, y dirigimos nuestra atención hacia los tests de especificación incorrecta. Aplicamos a ambos modelos los tres tests propuestos en la Sección 5.6, y obtuvimos los valores p que se muestran en el Cuadro B.2. Estos valores indican que ambos modelos están correctamente especificados, ya que no hay correlación entre los resíduos hasta el orden duodécimo, no hay cambio en los parámetros entre los regímenes y, por último, la varianza de los resíduos se mantiene constante a través del tiempo.

Podemos ver más evidencias de que nuestros modelos logran capturar el comportamiento dinámico de la serie en la Figura B.15, donde se muestra la serie residual y sus funciones ACF y PACF.

Finalmente, para comprobar las capacidades de predicción de los modelos, estos fueron reestimados utilizando únicamente los datos hasta el año 1924, dejando el resto de la serie, desde 1925 hasta el final para ser predicha mediante los modelos. Los resultados de tales predicciones son los mostrados en la Figura

Cuadro B.2.: Resultados de los tests de especificación errónea para el problema de las capturas de linces.

Test for q-order serial correlation			Test for parameter constancy	
	NCSTAR	NCGSTAR	NCSTAR	NCGSTAR
q	p-value	p-value	p-value	p-value
1	0.452	0.411	0.881	0.921
2	0.455	0.622		
3	0.354	0.587		
4	0.234	0.733	Test for constant variance	
5	0.834	0.234	p-value	p-value
6	0.236	0.834	0.179	0.645
7	0.716	0.532		
8	0.458	0.424		
9	0.347	0.672	Test for an extra rule	
10	0.673	0.562	p-value	p-value
11	0.702	0.623	0.212	0.328
12	0.422	0.789		

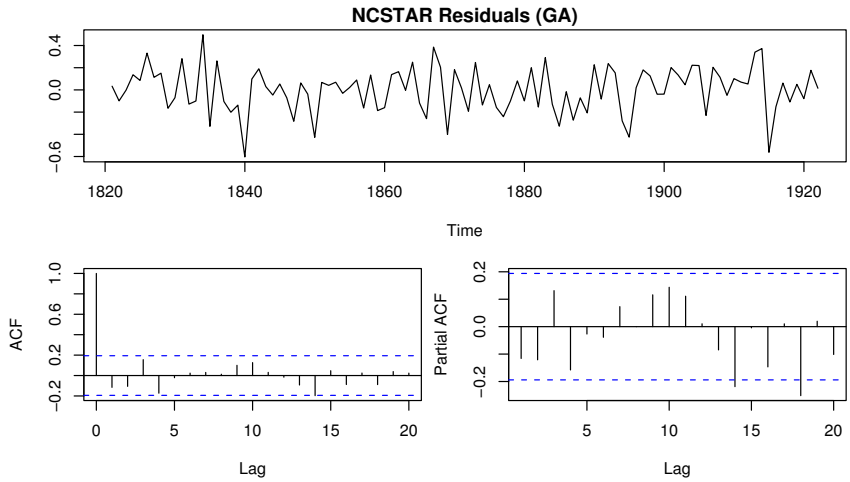

Figura B.15.: La serie de resíduos, junto a sus ACF y PACF, del modelo NCSTAR ajustado para la serie de las capturas de linces transformada.

B.16, mientras que el Cuadro 6.7 muestra el error cuadrático medio para cada modelo.

Cuadro B.3.: Error cuadrático medio en las predicciones del problema de las capturas de linces.

	NCSTAR	NCGSTAR
BFGS	0.0158	0.0392
GA	0.0156	0.0385

B.6. Conclusiones

Para enfrentar el problema del análisis de series temporales, existen dos enfoques provenientes de diferentes disciplinas científicas que utilizan ideas distintas para resolver los mismos problemas. Por un lado, el enfoque tradicional autorregresivo surge en el marco de la Estadística clásica y estudia las relaciones (usualmente lineales) que existen entre valores temporalmente desplazados de una serie. También, recientemente, se han desarrollado modelos avanzados que permiten describir comportamientos no lineales a través de regímenes locales.

Por otro lado, en las últimas décadas los desarrollos de la Inteligencia Artificial dieron lugar a un área de conocimiento llamada Computación Flexible cuyo objetivo fundamental es la resolución de problemas complejos con difícil solución analítica y que están aquejados también de incertidumbre o vaguedad. Las series temporales son uno de los problemas que han sido tratados mediante la colección de técnicas que forman la Computación Flexible.

Los objetivos principales de este trabajo eran dos: nos propusimos explorar las relaciones existentes entre ambas disciplinas y explotar dichas relaciones para mejorar las soluciones disponibles para el problema del análisis de series temporales. Globalmente, estos objetivos pueden considerarse cumplidos: encontramos fuertes conexiones entre modelos provenientes de las dos áreas y a partir de ellas iniciamos la transferencia de conocimiento de una a otra.

De una forma más precisa, demostramos que, al aplicarla al problema de las series temporales, la regla difusa, elemento básico de una amplia familia de

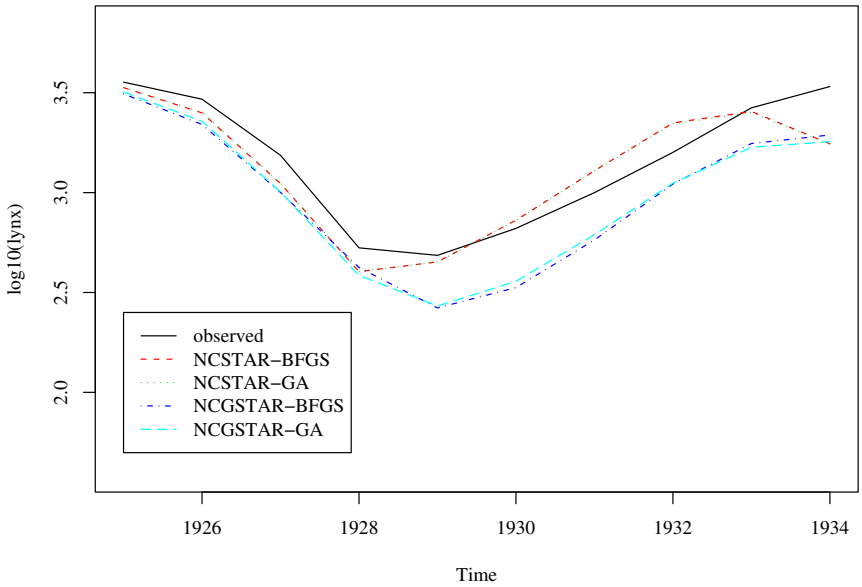

Figura B.16.: Resultados de predicción para la serie de las capturas de linces transformada.

modelos de Computación Flexible, puede ser vista como una generalización del modelo autorregresivo, que a su vez es una pieza fundamental para los modelos estadísticos avanzados no lineales.

Este resultado nos animó a estudiar los llamados modelos autorregresivos de regímenes cambiantes, y fuimos capaces de demostrar las estrechas relaciones que los unen con un modelo popular de la Computación Flexible: el modelo basado en reglas difusas. Encontramos que varios modelos de regímenes cambiantes recientemente desarrollados eran casos particulares del modelo basado en reglas difusas, y esta es una de las aportaciones principales de esta tesis. A la vez, supone la exitosa consecución del primero de los objetivos de nuestro trabajo.

Una vez que los lazos entre los modelos basados en reglas difusas y los modelos de regímenes cambiantes fueron establecidos, dirigimos nuestra atención a los beneficios que podrían ser derivados de este hecho. Se abrieron entonces dos caminos para trabajar esta idea: la extracción de conocimiento desde los modelos de regímenes cambiantes y su aplicación a los modelos basados en reglas difusas, y el uso de técnicas de Computación Flexible para mejorar dichos modelos estadísticos nolineales.

Respecto a los modelos de Computación Flexible, existe un escepticismo tradicional por parte de científicos de otras áreas más teóricas relacionado con la presunta falta de un sólido marco de trabajo matemático que les sirva de base. Lo cierto es que, usualmente, en este área prevalece un enfoque de ingeniería que asigna más importancia a la resolución efectiva de problemas que a otras consideraciones. Sin embargo, a partir del resultado de equivalencia expuesto en esta tesis, se abrió la prometedora posibilidad de desarrollar un marco formal estadístico para los modelos basados en reglas difusas.

La segunda de las aportaciones importantes de este trabajo consiste en la adaptación de la teoría estadística de los modelos basados en regímenes cambiantes al marco de los modelos basados en reglas difusas. Conseguimos desarrollar efectivamente resultados teóricos como la estacionariedad asintótica o la normalidad de los modelos, y esto nos permitió obtener un nuevo método de identificación para modelos basados en reglas difusas. Mediante este método, es posible justificar estadísticamente el número de reglas suficientes para modelar adecuadamente un problema dado. Considerando que la determinación del número de reglas es una cuestión abierta que ha sido ampliamente estudiada en la literatura desde el establecimiento del paradigma difuso, puede decirse que éste es otro resultado fundamental de esta tesis.

Sin embargo, el intercambio de conocimiento mencionado arriba puede ir en dos direcciones. Es posible también aplicar las técnicas de Computación Flexible al enfoque clásico estadístico para mejorarlo. Como ejemplo, decidimos aplicar Algoritmos Genéticos a la fase de estimación de los modelos autorregresivos basados en regímenes cambiantes. Para comprobar los beneficios de tal intercambio de conocimiento, propusimos un ciclo de modelado integrado que se beneficia de las ventajas de ambas áreas. Este procedimiento conjuga la determinación estadística del número de reglas con las ampliamente conocidas capacidades de aproximación de los Algoritmos Genéticos, y representa otra contribución origi-

nal más que se desprende de nuestro trabajo.

Los beneficios de este novedoso enfoque híbrido quedan claros tras el profundo estudio experimental realizado. Este estudio estuvo compuesto de, por un lado, un conjunto de experimentos de Montecarlo, en los que el ciclo de modelado fue aplicado a series temporales generadas artificialmente, y por otro de la aplicación de las propuestas a series provenientes de situaciones y estudios científicos reales. Más concretamente, estudiamos la conocida serie de la población del lince canadiense y una serie original: la compuesta por el número de llamadas recibidas por un centro de recepción de llamadas de emergencia. Estos experimentos muestran la utilidad práctica de las propuestas, las cuales resultan además favorablemente comparables con alternativas preexistentes.

Las aportaciones y resultados expuestos hasta ahora nos permiten concluir que el intercambio de conocimiento entre ambas áreas no es sólo posible sino que de hecho implica claras ventajas para lo que, no olvidemos, es el objetivo principal del análisis de series temporales: entender el comportamiento de procesos generadores de datos y utilizar convenientemente ese conocimiento para resolver los problemas asociados a ellos.

B.6.1. Líneas futuras de investigación

Quedan aún muchas vías por explorar en el estudio de los beneficios que conllevan las aportaciones de este trabajo. Entre otras, cabe citar las siguientes.

- La extensión del uso de técnicas de Computación Flexible a otros modelos estadísticos existentes. De entre ellas, por ejemplo, resultaría interesante aplicar técnicas inteligentes para la selección de características, distintas metaheurísticas para optimización (colonias de hormigas, otros algoritmos evolucionarios) etc.

- El estudio de otras consecuencias derivadas del marco estadístico propuesto para modelos basados en reglas difusas, y su aplicación a problemas ya estudiados para mejorar sus resultados.

- El desarrollo de nuevos modelos híbridos que exploten las conexiones entre las dos disciplinas mencionadas anteriormente, como podrían ser modelos basados en reglas difusas que incorporaran información acerca de la varianza de los datos como hacen los modelos GARCH, por ejemplo.

Algorithm 3 Ciclo de modelado híbrido para MBRD.

[1.] Análisis exploratorio: estudiar las propiedades estadísticas de la serie: media, varianza, kurtosis, sesgo, función de autocorrelación (parcial)...
if es necesario **then**
  Identificar (y eliminar) factores de estacionalidad y tendencias.
  Aplicar otras transformaciones a la serie (normalización, diferenciación, ...).
end if
[2.] Definir la estructura del modelo (variables de entrada, variables de pertenencia).
[3.] Aplicar el test de linealidad.
if se acepta la hipótesis nula **then**
  La serie es lineal: el ciclo de modelado termina.
else
  while Se rechace la hipótesis nula **do**
    [4.] Añadir una regla nueva.
    Determinar un buen conjunto de parámetros iniciales.
    Ajustar los parámetros del modelo mediante máxima verosimilitud.
    Aplicar el test para añadir una nueva regla.
  end while
end if
[5.] Reajustar el modelo mediante una metaheurística (enfriamiento simulado, algoritmos genéticos, ...).
[6.] Aplicar los tests de especificación errónea (constancia de parámetros, independencia lineal y homoscedasticidad de los residuos).
if La hipótesis nula es rechazada en alguno de los tests **then**
  Considerar la posibilidad de recomenzar todo el proceso de nuevo, o la de utilizar un modelo distinto.
else
  El modelo está correctamente construido: el ciclo de modelado termina.
end if

Bibliography

[1] E.M. Abdelrahim and T. Yahagi. A new transformed input-domain ANFIS for highly nonlinear system modelling and prediction. *IEICE Trans. Fundamentals*, E84-A(8):1981–1985, August 2001. 5, 156

[2] F. Alba and C. Díaz de la Guardia. The effect of air temperature on the starting dates of ulmus, platanus and olea polen seasons in the SE iberian peninsula. *Aerobiologia*, 14:191–194, 1998. 139

[3] F. Alba, C. Díaz de la Guardia, F. Ocaña, and M. Valderrama. Modelos de regresión lineal dinámica aplicados al polen de olea y cupressaceae en la ciudad de granada. In *XIV Simposio de Palinología de la Asociación de Palinólogos de lengua española (APLE)*, Salamanca, Spain, 2002. 139

[4] T.W. Anderson. *The statistical analysis of time series*. John Wiley and Sons, New York, USA, 1971. 4, 155

[5] M. Basu and T.K. Ho, editors. *Data Complexity in Pattern Recognition*. Springer, 2006. 102, 187

[6] J.M. Benítez, J.L. Castro, and I. Requena. Are artificial neural networks black boxes? *IEEE Transactions on Neural Networks*, 8:1156–1164, 1997. 60

[7] G. E. P. Box and G. M. Jenkins. *Time Series Analysis: Forecasting and Control*. Holden–Day, San Francisco, 1970. 39, 172

[8] G. E. P. Box, G. M. Jenkins, and G. C. Reinsel. *Time Series Analysis: Forcasting and Control*. Prentice Hall, third edition, 1994. 4, 155

[9] C. G. Broyden. The convergence of a class of double-rank minimization algorithms. *Journal of the Institute of Mathematics and Its Applications*, 6:76–90, 1970. 105

[10] M.G. Bulmer. A statistical analysis of the 10-year cycle in Canada. *J. Anim. Ecol.*, 43:701 – 715, 1974. 119

[11] M.J. Campbell and A.M. Walker. A survey of statistical work on the McKenzie river series of annual Canadian lynx trappings for the years 1821 - 1934, and a new analysis. *J. Roy. Statist. Soc. A*, 140:411 – 431, 1977. 121

[12] M. Castellano-Méndez, M.J.Aira, I. Iglesias, V. Jato, and W. González-Manteiga. Artificial neural networks as a useful tool to predict the risk level of betula pollen in the air. *International Journal of Biometeorology*, 49:310 – 316, 2005. 138

[13] J.L. Castro. Fuzzy logic controllers are universal approximators. *IEEE Trans. Systems, Man and Cybernetics*, 25(4), 1995. 18, 169

[14] J.L. Castro and M. Delgado. Fuzzy systems with defuzzification are universal approximators. *IEEE Trans. Systems, Man and Cybernetics*, 1996. 18, 169

[15] V. Cerný. Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of Optimization Theory and Applications*, 45(1):41–51, 1985. 103

[16] R. Chen and R.S. Tsay. Functional-coefficient autoregressive models. *Journal of the American Statistical Association*, 88:298–308, Mar 1993. 54

[17] Robert B. Cleveland, William S. Cleveland, Jean E. Mcrae, and Irma Terpenning. STL: A seasonal-trend decomposition procedure based on loess. *Journal of Official Statistics*, 6(1):3–73, 1990. 41, 139, 173

[18] P. K. Dash, A. C. Liew, S. Rahman, and S. Dash. Fuzzy and neuro-fuzzy computing models for electric load forecasting. *Engineering Applications of Artificial Intelligence*, 8(4), 1995. 5, 156

[19] Robert B. Davies. Hypothesis testing when a nuisance parameter is present only under the alternatives. *Biometrika*, 74(1):33–43, 1987. 77

[20] C. Díaz de la Guardia, F. Alba, M.M. Trigo, C. Galán, L. Ruiz, and S. Sabariego. Aerobiological analysis of Olea europaea l. pollen in different localities of southern spain. *Grana*, 42:234–243, 2003. 138, 139

[21] G. Deco, C. Schittenkopf, and B. Schürmann. Dynamical analysis of time series by statistical tests. *International Journal of Bifurcation and Chaos*, 7(12):2629–2652, 1997. 4, 155

[22] E. Dominguez, C. Galán, F. Villamandos, and F. Infante. Manejo y evaluación de los datos obtenidos en los muestreos aerobiológicos. *Monografías REA / EAN*, 1:1–18, 1991. 138

[23] D. Dubois and H.Prade. *Fuzzy Sets and Systems*. Academic Press, 1980. 12, 160, 162

[24] C. Elton and M. Nicholson. The ten-year cycle in numbers of the lynx in Canada. *J. Anim. Ecol.*, 11, 1942. 119

[25] SEAIC (*Sociedad Española de Alergología e Inmunología Clínica*). Grasses, how to interpret the pollen counts. http://polenes.com/en/interpretacion.html, october 2005. 140

[26] M. Fariñas and C. E. Pedreira. Mixture of experts and Local-Global Neural Networks. In *ESANN'2003 proceedings - European Symposium on Artificial Neural Networks*, pages 331–336, 2003. 52

[27] Antonio Fiordaliso. A nonlinear forecasts combination method based on Takagi-Sugeno fuzzy systems. *International Journal of Forecasting*, 14(3):367–379, 1998. 4, 155

[28] R. Fletcher. A new approach to variable metric algorithms. *Computer Journal*, 13:317–322, 1970. 105

[29] C. Galán, P. Cariñanos, H. García-Mozo, P. Alcázar, and E. Domínguez-Vílchez. Model for forecasting Olea europaea L. airborne pollen in South-West Andalusia, Spain. *Int J Biometeorol*, 45:59–63, 2001. 5, 138, 156

[30] M-T. Gan, M. Hanmandlu, and A.H. Tan. From a Gaussian mixture model to additive fuzzy systems. *IEEE Transactions on Fuzzy Systems*, 13(3):303–316, 2005. 75

[31] Adam Gaweda and Jacek Zurada. Data-driven linguistic modelling using relational fuzzy rules. *IEEE Transactions on Fuzzy Systems*, 11(1), february 2003. 5, 156

[32] D. Goldfarb. A family of variable metric updates derived by variational means. *Mathematics of Computation*, 24:23–26, 1970. 105

[33] Belén Gopegui. *El padre de Blancanieves*. Anagrama, 2007. 211

[34] J.W. Grzymala-Busse. *Managing uncertainty in expert systems*. Kluwer Academic, Dordrecht, 1991. 86

[35] Hans Hellendoorn and Dimiter Driankov, editors. *Fuzzy model identification: selected approaches*. Springer-Verlag, London, UK, 1997. 86

[36] J. T. Gene Hwang and A. Adam Ding. Prediction intervals for artificial neural networks. *Journal of the American Statistical Association*, 92(438):748–757, 1997. 71, 72

[37] J.P. Ignizio. *Introduction to Expert Systems*. McGraw-Hill, Inc., New York, NY, USA, 1991. 86

[38] R.A. Jacobs, M.I. Jordan, S.J. Nowlan, and G.E. Hinton. Adaptive mixtures of local experts. *Neural computation*, 3:79–87, 1991. 52

[39] J.-S.R. Jang. ANFIS: Adaptive-network-based fuzzy inference system. *IEEE Transactions on systems, man and cibernetics*, 23(3), May-June 1993. 5, 26, 32, 156

[40] J.-S.R. Jang and C.-T. Sun. Predicting chaotic time series with fuzzy if-then rules. *Fuzzy Systems, 1993., Second IEEE International Conference on*, pages 1079–1084 vol.2, 1993. 5, 156

[41] Robert I. Jennrich. Asymptotic properties of non-linear least squares estimators. *The Annal of Mathematical Statistics*, 40:633–643, 1969. 83

[42] K. Kalaitzakis, G.S. Stavrakakis, and E.M. Anagnostakis. Short-term load forecasting based on artificial neural networks parallels implementation. *Electric Power Systems Research*, 63:185–196, 2002. 5, 156

[43] Nicola Kasabov. Evolving fuzzy neural networks for supervised/unsupervised online knowledge-based learning. *IEEE Transactions on Systems, Man and Cybernetics - part B*, 31(6):902–918, December 2001. 5, 156

[44] Nikola K. Kasabov. On-line learning, reasoning, rule extraction and aggregation in locally optimized evolving fuzzy neural networks. *Neurocomputing*, 41:25–45, 2001. 5, 156

[45] C. Kim, I. Yu, and Y.H. Song. Kohonen neural network and wavelet transform based approach to short-term load forecasting. *Electric Power Systems Research*, 63:169–176, 2002. 5, 156

[46] J. Kim and N. Kasabov. HyFIS: adaptive neuro-fuzzy inference systems and their application to nonlinear dynamical systems. *Neural Networks*, 12:1301–1319, 1999. 5, 32, 34, 37, 38, 156

[47] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science, Number 4598, 13 May 1983*, 220, 4598:671–680, 1983. 103

[48] G.J. Klir and B. Yuan. *Fuzzy Sets and Fuzzy Logic*. Prentice Hall, 1995. 12, 160, 162

[49] B. Kosko. Fuzzy systems as universal approximators. *IEEE Trans. Computers*, 43(11):1324–1333, 1994. 18, 20, 169, 171

[50] R.J. Kuo. A sales forecasting system based on fuzzy neural network with initial weights generated by genetic algorithm. *European journal of operational research*, 129, 2001. 5, 156

[51] R.J. Kuo and K.C. Xue. A decission support system for sales forecasting through fuzzy neural networks with asymmetric fuzzy weights. *Decission support systems*, 24, 1998. 5, 156

[52] C.C. Lee. Fuzzy logic in control systems: fuzzy logic controller. i. *Systems, Man and Cybernetics, IEEE Transactions on*, 20(2):404–418, Mar/Apr 1990. 21, 25

[53] C.C. Lee. Fuzzy logic in control systems: fuzzy logic controller. ii. *Systems, Man and Cybernetics, IEEE Transactions on*, 20(2):419–435, Mar/Apr 1990. 21, 25

[54] Friedrich Leisch, Adrian Trapletti, and Kurt Hornik. Stationarity and stability of autoregressive neural network processes. In *Proceedings of the 1998 conference on Advances in neural information processing systems II*, pages 267–273, Cambridge, MA, USA, 1999. MIT Press. 69

[55] A. Lendasse, M. Verleysen, E. de Bodt, M. Cottrell, and P. Grégoire. Forecasting time-series by Kohonen classification. In D-Facto Publications, editor, *European Symposium on Artificial Neural Networks*, Brussels, April 1998. 5, 156

[56] J. Leski and E. Czogala. A new artificial neural network based fuzzy inference system with moving consequents in If-Then rules and selected applications. *Fuzzy Sets and Systems*, 108:289–297, 1999. 5, 156

[57] Xiang Li, Cheng-Leong Ang, and Robert Gray. An intelligent business forecaster for strategic business planning. *Journal of Forecasting*, 18(3):181 – 204, 1999. 4, 155

[58] R. Luukkonen, P. Saikkonen, and T. Teräsvirta. Testing linearity against smooth transition autoregressive models. *Biometrika*, 75:491–499, 1988. 77, 78, 80

[59] José Luis Aznarte M., José Manuel Benítez Sánchez, Diego Nieto Lugilde, Concepción de Linares Fernández, Consuelo Díaz de la Guardia, and Francisca Alba Sánchez. Forecasting airborne pollen concentration time series with neural and neuro-fuzzy models. *Expert Syst. Appl.*, 32(4):1218–1225, 2007. 134, 138, 139

[60] L.P. Maguire, B. Roche, T.M. McGuinnity, and L.J. McDaid. Predicting a chaotic time series using a fuzzy neural network. *Information Sciences*, 112, 1998. 5, 156

[61] E.H. Mamdani and S. Assilian. An experiment in linguistic synthesis with a fuzzy logic controller. *Intern. J. of Man-Machines Studies*, 1975. 18, 32, 169

[62] M.C. Medeiros and T. Teräsvirta. Statistical methods for modelling neural networks. *Engineering Intelligent Systems*, 9:227–235, 2001. 63

[63] M.C. Medeiros and A. Veiga. Diagnostic checking in a flexible nonlinear time series model. *Journal of Time Series Analysis*, 24:461–482, 2003. 64, 89, 91, 92, 93

[64] M.C. Medeiros and A. Veiga. A flexible coefficient smooth transition time series model. *IEEE Transactions on Neural Networks*, 16(1):97–113, January 2005. 52, 54, 67, 68, 70, 71, 72, 77, 79, 82, 84, 97, 102, 121

[65] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953. 104

[66] P.A.P. Moran. The statistical analysis of the Canadian lynx cycle. i: structure and prediction. *Aust. J. Zool.*, 1:163 – 173, 1953. 120, 121

[67] J. Nie. Nonlinear time-series forecasting: a fuzzy-neural approach. *Neurocomputing*, 16, 1997. 5, 156

[68] D. Nieto, J.L. Aznarte, F. Alba, J.M. Benítez, C. Díaz de la Guardia, and C. De Linares. Modelling and forecasting *Olea Europaea* L. airborne pollen concentrations in Granada (Southern Spain) using Soft Computing. In *Polen*, volume 14, pages 372–373, 2004. 5, 156

[69] Benedikt M. Potscher and Ingmar R. Prucha. A class of partially adaptive one-step m-estimators for the non-linear regression model with dependent observations. *Journal of Econometrics*, 32(2):219–251, July 1986. 83, 85

[70] M.B. Priestley. *Spectral analysis and time series*, volume I and II. Academic Press, San Diego, USA, 1981. 4, 155

[71] A. Ranzi, P. Lauriola, V. Marletto, and F. Zinoni. Forecasting airborne pollen concentrations: Development of local models. *Aerobiologia*, 19:39–45, 2003. 138

[72] Gianluigi Rech. Forecasting with artificial neural network models. Working Paper Series in Economics and Finance 491, Stockholm School of Economics, February 2002. available at http://ideas.repec.org/p/hhs/hastef/0491.html. 4, 155

[73] Gianluigi Rech, Timo Teräsvirta, and Rolf Tschernig. A simple variable selection technique for nonlinear models. *Communications in Statistics - Theory and Methods*, 30(6):1227 – 1241, may 2001. 96

[74] D. F. Shanno. Conditioning of quasi-newton methods for function minimization. *Mathematics of Computation*, 24:647–656, 1970. 105

[75] Chad Stroomer and David E.A. Giles. Income convergence and trade openness: Fuzzy clustering and time series evidence. Technical Report 0304, Department of Economics, University of Victoria, May 2003. available at http://ideas.repec.org/p/vic/vicewp/0304.html. 4, 155

[76] Mayte Suarez-Farinas, Carlos E. Pedreira, and Marcelo C. Medeiros. Local global neural networks: A new approach for nonlinear time series modeling. *Journal of the American Statistical Association*, 99:1092–1107, December 2004. 52, 64, 68, 70, 72, 77, 83, 178

[77] M. Sugeno and G.T. Kang. Structure identification of fuzzy model. *Fuzzy Sets and Systems*, 28:15–33, 1988. 19, 171

[78] M. Sugeno and M. Nishida. Fuzzy control of model car. *Fuzzy Sets and Systems*, 16:103–113, 1985. 19, 163, 171

[79] Hectór J. Sussmann. Uniqueness of the weights for minimal feedforward nets with a given input-output map. *Neural Networks*, 2:589–593, 1992. 71, 72

[80] J.A. Sánchez-Mesa, C. Galan, J.A. Martínez-Heras, and C. Hervás-Martínez. The use of a neural network to forecast daily grass pollen concentration in a mediterranean regions: the southern part of the iberian peninsula. *Clinical and Experimental Allergy*, 32:1606 – 1612, 2002. 138

[81] T. Takagi and M. Sugeno. Fuzzy identification of systems and its applications to modeling and control. *IEEE Trans. Systems, Man and Cybernetics*, 15:116–132, 1985. 19, 171

[82] M. Tamimi and R. Egbert. Short term electric load forecasting via fuzzy neural collaboration. *Electric Power Systems Research*, 56, 2000. 5, 156

[83] T. Teräsvirta. Specification, estimation and evaluation of smooth transition autoregresive models. *J. Am. Stat. Assoc.*, 89:208–218, 1994. 47, 107, 174

[84] Timo Teräsvirta, Marcelo C. Medeiros, and Gianluigi Rech. Building neural network models for time series: a statistical approach. *Journal of Forecasting*, 25(1):49–75, 2006. 49, 50, 51, 70, 77, 82, 176

[85] H. Tong. On a threshold model. *Pattern Recognition and Signal Processing*, 1978. 46, 173

[86] H. Tong. *Non-linear time series: a dynamical system approach*. Oxford University Press, 1990. 4, 43, 155

[87] Dick van Dijk, Timo Teräsvirta, and Philip Hans Franses. Smooth transition autoregresive models — a survey of recent developments. *Econometric Reviews*, 21(1):1–47, 2002. 47, 174

[88] J. Vázquez-Abad, F. Fdez-Riverola, and J.M. Corchado. Forecasting economic cycles with conectionist models. In *I International Meeting on Economic Cycles*, 2000. 5, 156

[89] L. Wang and J. M. Mendel. Generating fuzzy rules by learning from examples. *IEEE Transactions on Systems, Man and Cybernetics*, 22(6):1414–1427, November/December 1992. 36, 37, 86

[90] Andreas S. Weigend and Neil A. Gershenfeld, editors. *Time Series Prediction: Forecasting the Future and Understanding the Past*. Addison-Wesley, 1994. 151

[91] H. White. *Estimation, inference and specification analysis*. Cambridge University Press, 1994. 84, 85

[92] Halbert White. Consequences and detection of misspecified nonlinear regression models. *Journal of the American Statistical Association*, 76(374):419–433, 1981. 84

[93] Halbert White and Ian Domowitz. Nonlinear regression with dependent observations. *Econometrica*, 52(1):143–161, 1984. 84, 86

[94] P.R. Winters. Forecasting sales by exponentially weighted moving averages. *Management Science*, 6:324–342, 1960. 42, 173

[95] Jeffrey M. Wooldridge. Estimation and inference for dependent processes. In R. F. Engle and D. McFadden, editors, *Handbook of Econometrics*, volume 4 of *Handbook of Econometrics*, chapter 45, pages 2639–2738. Elsevier, 1986. 84

[96] Y. Xia and W.K. Li. On single-index coefficient regression models. *Journal of the American Statistical Association*, 94:1275–1285, 1999. 54

[97] S.J. Yakowitz and J.D. Spragins. On the identifiability of finite mixtures. *The Annals of Mathematical Statistics*, 38(1):209–214, 1968. 73, 75

[98] L.A. Zadeh. Fuzzy sets. *Information and control*, 3(8):338–353, 1965. 11, 159

[99] L.A. Zadeh. Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Trans. Systems, Man and Cybernetics*, 1(1):28–44, 1973. 15, 164

[100] L.A. Zadeh. The concept of a linguistic variable and its applications to approximate reasoning. part i. *Information Sciences*, 8:199–249, 1975. 13, 161

[101] L.A. Zadeh. The concept of a linguistic variable and its applications to approximate reasoning. part ii. *Information Sciences*, 8:301–357, 1975. 13, 161

[102] L.A. Zadeh. The concept of a linguistic variable and its applications to approximate reasoning. part iii. *Information Sciences*, 9:43–80, 1975. 13, 161

[103] G. Zhang, B.E. Patuwo, and M.Y. Hu. Forecasting with artificial neural networks: the state of the art. *International journal of forecasting*, 14:35–62, 1998. 4, 155

"DETRÁS de los teléfonos móviles, el zumo de naranja, las vigas de hormigón de los hospitales y el té de jazmín, las costas sucias y el suelo cementado y los montes vivos todavía, los libros que se estudian en los colegios, el trabajo diario, los jóvenes en paro y los adultos en paro, los clientes de la business class y los clientes del autobús inseguro y los del club de fútbol y los de las bicicletas inmóviles y los que compran medicinas, detrás de un paisaje observado a través de las ventanas de un hotel o pisando tierra, detrás de las natillas, del calor. Del salario usado como abono, soborno o recompensa, los departamentos de las universidades, los despidos y los gritos, la docilidad y los pimientos rojos y las televisiones y los pájaros, la lámpara encendida, el sillón de orejas, los directores de recursos humanos, la composición del detergente, la depresión, el agua, las fichas del parchís, los cementerios, las latas de mejillones, los preservativos y los dientes, los animales y las gasolineras, las películas y los muertos, los créditos y la imaginación, detrás de la clase de vida por la que discurrimos hay, siempre, propietarios que calculan sus beneficios."[5]

Belén Gopegui, [33].

[5] *Behind cellular phones, orange juice, hospitals' concrete beams and jasmine tea, dirty coasts and paved floors and mountains that still thrive, books studied at schools, daily work, unemployed youngsters and unemployed adults, business class clients and unreliable buses clients and those from the football club and those of the stationary bicycles and those who buy medicines, behind a scenery observed through a hotel's windows or stepping on land, behind custard, heat. Behind salary used as fertilizer, bribe or reward, university departments, dismissals and shouting, docility and red pepper and televisions and birds, the turned on lamp, the arm chair, human resource managers, washing powder's chemical composition, depression, water, ludo counters, cemeteries, tins of mussels, condoms and teeth, animals and petrol stations, films and dead people, credits and imagination, behind the type of life through which we wander there are, always, owners calculating their profits.*

Se terminó de imprimir
en Granada, el día 11 del mes de
septiembre de dos mil ocho, día de
San Orlando, Proto y Jacinto
y séptimo aniversario de
los atentados del
World Trade
Center en
Nueva
York.