UNIVERSIDADE DE LISBOA

FACULDADE DE CIÊNCIAS

DEPARTAMENTO DE ESTATÍSTICA E INVESTIGAÇÃO OPERACIONAL

THE COMBINATION OF NEURAL ESTIMATES IN PREDICTION AND DECISION PROBLEMS

Paulo Sérgio Abreu Freitas

DOUTORAMENTO EM ESTATÍSTICA E INVESTIGAÇÃO OPERACIONAL

(Especialidade de Análise de Sistemas)

2008

UNIVERSIDADE DE LISBOA

FACULDADE DE CIÊNCIAS

DEPARTAMENTO DE ESTATÍSTICA E INVESTIGAÇÃO OPERACIONAL

THE COMBINATION OF NEURAL ESTIMATES IN PREDICTION AND DECISION PROBLEMS

Paulo Sérgio Abreu Freitas

Tese orientada pelo Prof. Doutor António José Lopes Rodrigues

DOUTORAMENTO EM ESTATÍSTICA E INVESTIGAÇÃO OPERACIONAL

(Especialidade de Análise de Sistemas)

2008

The Combination of Neural Estimates

in Prediction and Decision Problems

Paulo Sérgio Abreu Freitas

Abstract

In this dissertation, different ways of combining neural predictive models or neural-based forecasts are discussed. The proposed approaches consider mostly Gaussian radial basis function networks, which can be efficiently identified and estimated through recursive/adaptive methods. Two different ways of combining are explored to get a final estimate – model mixing and model synthesis –, with the aim of obtaining improvements both in terms of efficiency and effectiveness. In the context of model mixing, the usual framework for linearly combining estimates from different models is extended, to deal with the case where the forecast errors from those models are correlated. In the context of model synthesis, and to address the problems raised by heavily nonstationary time series, we propose hybrid dynamic models for more advanced time series forecasting, composed of a dynamic trend regressive model (or, even, a dynamic harmonic regressive model), and a Gaussian radial basis function network. Additionally, using the model mixing procedure, two approaches for decision-making from forecasting models are discussed and compared: either inferring decisions from combined predictive estimates, or combining prescriptive solutions derived from different forecasting models. Finally, the application of some of the models and methods proposed previously is illustrated with two case studies, based on time series from finance and from tourism.

Keywords: Time series forecasting; neural networks; model combination; adaptive methods; optimal decision-making.

Combinação de Estimativas Neuronais

em Problemas de Previsão e de Decisão

Paulo Sérgio Abreu Freitas

Resumo

A previsão de séries temporais é um problema comum em muitas aplicações, e é habitual que as variáveis observadas apresentem comportamento não estacionário, isto é, distribuição e correlações não constantes no tempo. Mesmo em casos mais difíceis como esses, é desejável poder dispor de modelos relativamente simples e robustos, com boa capacidade preditiva, e estimados por métodos computacionalmente eficientes.

Assim, muitos autores – entre os quais Granger (1989), Prémio Nobel de Economia em 2003 – defendem que é conveniente afastarmo-nos da visão mais clássica, em que se supõe ser (sempre) razoável procurar identificar um modelo como sendo 'o melhor', e onde os processos de identificação e optimização desse modelo único requerem normalmente uma computação exigente e algo exaustiva. Nesses processos, muitas escolhas têm de ser feitas, que podem não ser as mais adequadas, nomeadamente a escolha do critério mais apropriado para a selecção da estrutura (ou tipo), da dimensão (ou grau de complexidade) e da parametrização do modelo.

Em alternativa ao paradigma clássico, múltiplos modelos – normalmente sub-optimais – podem ser explorados e combinados entre si, de forma a ultrapassar ou minimizar o risco de escolha e identificação de um único modelo não apropriado, mesmo que optimizado, que é limitado nas suas capacidades em relação às características dos próprios dados. Esta abordagem alternativa tem a potencial vantagem de melhorar os resultados, quer em termos de eficiência computacional, quer em termos de eficácia, ou capacidade preditiva. Os esquemas de combinação originalmente propostos na literatura sobre previsão de séries temporais têm sido estendidos no contexto da aplicação de redes neuronais supervisionadas.

No trabalho conducente a esta dissertação, pretendeu-se estudar diferentes formas de combinar estimativas de modelos preditivos obtidas, maioritariamente, a partir de redes

neuronais. Em todos os casos, foram considerados modelos paramétricos lineares, com a possibilidade de incluir parâmetros variáveis no tempo, de forma a poderem ser estimados recursivamente. Os modelos neuronais que considerámos no estudo foram as redes de funções de base radiais Gaussianas, e foram discutidas várias formas de as 'treinar' através de métodos de estimação recursiva e identificação adaptativa.

Foram exploradas, essencialmente, duas formas de combinar linearmente estimativas preditivas com o objectivo de melhorar a eficiência e eficácia dos resultados finais: a *mistura de modelos* e a *síntese de modelos*.

No contexto da *mistura de modelos*, foi proposta uma extensão da formulação mais conhecida para combinação linear de estimativas usando modelos distintos, de forma a poder lidar com correlações significativas possivelmente existentes entre as sequências de estimativas preditivas dos modelos individuais. Para o efeito, foi apenas considerado o caso de dois modelos, mas a ideia pode ser generalizada a mais modelos. Mais especificamente, foi incorporada na combinação linear usual um termo não linear, constituído pelo produto, ponto a ponto, das duas sequências de estimativas preditivas dos modelos individuais. Verificámos a existência de melhoria significativa na combinação estendida se (e só se) a sequência de estimativas adicional (associada ao termo não linear) e a sequência de erros resultante da combinação linear clássica estiverem fortemente correlacionadas.

No contexto da *síntese de modelos*, foi proposta uma abordagem para a identificação de modelos compostos aditivamente por um modelo de regressão dinâmica (nomeadamente, regressão de tendência dinâmica ou regressão harmónica dinâmica) e por uma rede neuronal (rede de funções de base radiais Gaussianas), com vista à modelação e previsão de séries não estacionárias. Para esse efeito, justificamos a opção pela realização inicial de uma pré-filtragem usando um modelo de regressão dinâmica, em alternativa aos procedimentos clássicos de pré-processamente para estacionarização da série, e antes da identificação da componente neuronal. Posteriormente, é proposta a estimação simultânea dos modelos componentes, fazendo uso de parte dos valores dos hiperparâmetros identificados na fase de pré-filtragem, principalmente os que dizem respeito ao modelo neuronal. A identificação do modelo de regressão dinâmica deve ser feita de uma forma criteriosa, já que tem por finalidade modelar exclusivamente grande parte, ou mesmo todos os efeitos de baixa frequência visíveis no periodograma (ou espectro empírico) da série. Recomenda-se que essa identificação deva ser feita por avaliação da capacidade preditiva desse submodelo para horizontes temporais relativamente longos. Caso contrário, corre-se

o risco de haver interferência entre os dois submodelos (o de regressão dinâmica e o neuronal) na modelação dos efeitos que dizem respeito às altas frequências.

A partir dos resultados das experiências computacionais efectuadas, com séries simuladas, verificou-se que as abordagens clássicas de pré-processamento podem ser bastante falíveis. Concluiu-se que a metodologia híbrida, constituída por um modelo de regressão dinâmica e uma rede neuronal, pode ser considerada como uma abordagem genérica na aplicação a problemas de previsão de séries temporais não estacionárias e com correlações não lineares, útil em especial sempre que não haja certeza sobre qual o 'verdadeiro' processo gerador da série.

Numa fase posterior, a metodologia de *mistura de modelos* foi estudada no contexto de problemas iterados de decisão optimal sob risco, onde as decisões são avaliadas por funções de custo mais realistas, assimétricas, em vez das habitualmente usadas para identificação e estimação de modelos preditivos, incluindo as baseadas no critério de mínimos quadrados.

Foram descritas e ilustradas duas abordagens: a inferência de uma decisão optimal a partir de uma previsão resultante da combinação de previsões; ou, a combinação de soluções prescritivas, cada uma obtida a partir de uma previsão diferente. Conjectura-se que é preferível combinar primeiro as previsões e, só depois, tomar as decisões finais com base nessas previsões combinadas. A ideia fundamental é de que o modelo preditivo pode ser mais convenientemente estimado, com base no critério dos mínimos quadrados, enquanto que o critério usualmente considerado para avaliação do modelo prescritivo se baseia em funções não diferenciáveis, o que dificulta o processo de estimação, e requer o uso de métodos de optimização não linear, mais difíceis de praticar. Em particular, para que os pesos da combinação linear de soluções prescritivas fossem estimados, com base em funções assimétricas de tipo idêntico àquelas consideradas neste estudo, tivemos de recorrer a métodos de programação linear mais complexos (nomeadamente, métodos de ponto interior), com a desvantagem de desconhecermos versões recursivas dos mesmos, tornando impraticável a actualização adaptativa desses pesos para novas observações futuras.

Embora seja conveniente realizar um estudo mais exaustivo e profundo, os resultados das experiências entretanto realizadas indicaram que nenhuma das abordagens (combinar as soluções prescritivas ou as soluções preditivas) domina a outra, dependendo o seu desempenho relativo da função de custos assimétricos considerada, do horizonte temporal que se considere para a previsão, e, ainda, das características da própria série temporal.

Finalmente, foi ilustrada a aplicação de alguns modelos e métodos propostos e discutidos nesta dissertação através do estudo de dois casos, relacionados com a análise do índice da Bolsa de Valores de Lisboa PSI20, e com a análise de uma série de turismo, relativa ao número de hóspedes entrados na Região Autónoma da Madeira. No primeiro caso, limitámo-nos a aplicar a metodologia de pré-filtragem, considerando um modelo híbrido de regressão de tendência dinâmica combinado com uma rede neuronal. No segundo caso, a aplicação de modelos híbridos de regressão harmónica dinâmica e redes neuronais é complementada com a ilustração da metodologia de *mistura de modelos*, no contexto da tomada de decisões baseadas em previsões.

Palavras-chave: Previsão de séries temporais; redes neuronais; combinação de modelos; métodos adaptativos; decisão optimal.

“Don’t limit yourself to a single model: More than one may be useful for understanding different aspects of the same phenomenon.”

— Solomon W. Golomb

“Don’t let reality keep you from your dreams.”

— Anonymous

To my parents

Acknowledgments

I want to show my appreciation to my Advisor, Professor António José Rodrigues, for his guidance, and valuable advice.

The other colleagues within the working group of Professor António Rodrigues have also earned my great appreciation and affection. The numerous discussions we have engaged into have been very stimulating and fruitful.

To the Department of Mathematics and Engineering, University of Madeira, I express my recognition for all the support provided to me.

I also acknowledge the financial support from Centro de Investigação Operacional (FCUL) that allowed me to present the work in progress at several conferences.

To Fundação Caridade Pestana and its Administrator, Dr. Dionísio Pestana, I express my gratitude for the financial support for travelling to the University of Lisbon.

I thank my parents, who have always illuminated my path, my brothers and sister, and João Carlos for the support they have so generously given me, thereby making it possible for me to find my way.

Last, but not the least, I would like to express thanks to all my friends that have contributed with advice and care to be more self-confident and to renew my strengths to keep going on.

Only thus could I bring this work to its completion.

Contents

List of Figures

List of Tables

List of Algorithms

List of Symbols and Abbreviations

AR(p)	Autoregressive model of order p
CDec	Combining methodology: first, decide from predictions, then combine
CLC	Classic linear combination
CPred	Combining methodology: first, combine predictions, then decide
CRLC	Classic restricted linear combination
DHR	Dynamic harmonic regressive (model)
DS	Difference-stationary (series)
DTR	Dynamic trend regressive (model)
DTR;RBF	GRBFN modelling, after a preliminary DTR filtering
DTR-RBF	Model composed of DTR and GRBFN submodels
EWLS	Exponentially weighted LS
fd;RBF	GRBFN modelling, after first-order differencing
GRBFN	Gaussian RBFN
IAR(1)	First-order integrated autoregressive (model)
IRW	Integrated random walk (model)
LC	Least-cost (criterion/loss function)
LS	Least-squares (criterion/loss function)
MAD	Mean absolute deviation
MAPE	Mean absolute percentage error
MSE	Mean squared error
NAR	Nonlinear autoregressive (model)
NVR	Noise variance ratio
OLS	Ordinary LS (method)
pd;RBF	GRBFN modelling, after pattern-differencing
ps;RBF	GRBFN modelling, after pattern-standardising
RBFN	Radial basis function network
RLS	Recursive LS (method)
RLS-CA	Covariance-addition version of RLS
RMSE	Root mean squared error
RW	Random walk (model)
SSE	Sum of squared errors
tr;RBF	GRBFN modelling, after deterministic trend removal

TS	Trend-stationary (series)
XLC	Extended linear combination
XRLC	Extended restricted linear combination

Notation

y_k	output/observation/measurement (at time k)
$\mathbf{u}_k$	regression input variables/observation vector (at time k)
ε_k	observation noise (at time k)
$J(\boldsymbol{\theta})$	loss function
$\hat{\boldsymbol{\theta}}_k$	estimate of a vector of parameters $\boldsymbol{\theta}$ or $\boldsymbol{\theta}_k$
$\hat{\boldsymbol{\theta}}_{k\|k-1}$	1-step-ahead predictive estimate of $\boldsymbol{\theta}_k$
$\hat{\boldsymbol{\theta}}_{k\|k}$	filtered estimate of $\boldsymbol{\theta}_k$
$\mathbf{P}_k$	variance-covariance matrix of $\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_k$
$\mathbf{P}_{k\|k-1}$	variance-covariance matrix of $\boldsymbol{\theta}_k - \hat{\boldsymbol{\theta}}_{k\|k-1}$
$\mathbf{P}_{k\|k}$	variance-covariance matrix of $\boldsymbol{\theta}_k - \hat{\boldsymbol{\theta}}_{k\|k}$
$\hat{y}_{k\|k-h}$	h-steps-ahead predictive estimate of y_k
$e_{k\|k-h}$	h-steps-ahead prediction error of y_k
$\boldsymbol{\eta}_k$	disturbance/process/system vector noise
$\mathbf{F}$	state transition matrix
$\mathbf{Q} \equiv \mathrm{Var}(\boldsymbol{\eta}_k)$	disturbance noise covariance matrix
$r \equiv \mathrm{Var}(\varepsilon_k)$	observation noise variance
$\mathbf{Q}_r \equiv \mathbf{Q}/r$	matrix of noise variance-covariance ratios
$d_k \equiv y_k - \hat{y}_{k\|k-1}$	recursive estimator innovation (at time k)
$\mathbf{k}_k$	recursive estimator gain (at time k)

CHAPTER 1

Introduction

Contents

Abstract

We start by introducing some basic ideas about the subject of the thesis – methodologies for combining solutions in prediction and decision problems –, and briefly present the main objectives of our work.

1.1 Context of the Problem

Numerical supervised learning refers to any computational process for which a mathematical parametric model, neural or of other type, is identified and estimated from a set or sequence of arrays of data (called 'patterns'), defined in an *input-output* Euclidean space – for instance, the mapping of 'symptomatic' information, defined in an input space, into diagnostic, prognostic, or prescriptive information, defined in an output space. Once estimated, the model can be used for *generalisation*, that is, the inductive inference of 'appropriate' output patterns, from a supplementary set or sequence of new input patterns.

Time series forecasting is a common goal in many data mining applications, where most often the recorded data are indexed in time, and the variables have distributional properties and correlation effects that are typically nonstationary in time (see [20] for a review). Without sacrificing predictive accuracy, the models to be used should not be too complex, should be both flexible and robust, and the methods to estimate those models should be efficient. Therefore, it is convenient to depart from the classical point of view of identifying a single, 'clearly best' model, which might require a high computational burden for its identification and optimisation.

Most references in the literature on neural-based forecasting follow that traditional paradigm, usually referring to the application of Multilayer Perceptrons (see [79] for a review). These are highly nonlinear models requiring the optimisation of parameters defined in a high-dimensional space under a nonlinear least-squares cost function, typically with many local optima (see [50] for a comprehensive introduction to the optimisation of neural networks). In those models, parameter updating, given new data, is cumbersome and, most importantly, their direct application to nonstationary data is inadequate (see [64, 66]).

In most of the literature about time series forecasting and neural supervised learning, that traditional paradigm is followed, with all the effort directed to the identification and estimation of a single model, in some sense optimal within a class of many possible models, different in structure, in size or in parameterisation. The rationale behind this paradigm is the assumption that a 'best' model can be conveniently identified for a given problem.

In real-world problems, the 'true' model is likely to be unknown, and the models usually considered are severe abstractions of a much more complex reality. Some choices and

assumptions have to be made so that the problem under study can be acceptably modelled and the underlying optimisation problem solved. During this process, there are some issues that may be hard to sort out, such as choosing appropriate model selection criteria. In particular, if the chosen model is too complex or over-parameterised, it can learn the noise intrinsic to the data, thus causing poor generalisation performance, i.e., producing poor results when applied to new data (see, e.g., [5]).

As alternatives to the traditional paradigm, several approaches have been proposed, where multiple models (normally, suboptimal ones) are explored and combined. This aims to minimise the implicit risk of taking into account only one model, even if it is optimised, and that is limited in its capabilities with respect to the characteristics of the data and to the problem itself. Moreover, it has potential advantages in terms of efficiency, effectiveness and robustness.

The combination paradigm has been defended for several reasons, not only to improve accuracy or computational efficiency, but also to achieve stabilisation in the process of model selection, i.e., selecting a specific model that optimises a given criterion from a set of candidate models [75].

The first proposals in this direction appeared in the literature on time series forecasting over 3 decades ago. Most of them pointed to the optimisation of weights, constant in time, of a convex linear combination of the forecasts produced by different models. The first studies go back to Bates and Granger [4], and possibly others, who considered the linear combination of two different forecasting models. This approach was later extended to more than two models in [30, 52, 73]. Surveys of the vast literature on forecast combination can be found in [17, 21, 29]. Many other contributions have emerged on the linear combination of supervised neural networks, such as [36, 58] for regression problems; [34, 71] for classification problems, and [40, 80] for forecasting problems. During the last decade these schemes have been adapted to the context of supervised neural networks, as descriptive or predictive models.

Model combination can be carried out in different ways, namely using either linear or nonlinear parametric metamodels – see, e.g., [3, 23, 33] on the usage of neural networks, and [55] on combining forecasts with genetic algorithms as nonlinear models. Furthermore, it can be done with constant or variable weighting parameters, or based on nonparametric kernel-regression weighting approaches (see [42], for instance).

In the context of forecasting problems, the combining methods have been essentially accomplished by considering individual models of similar nature. There has been some

criticism when combining similar models, since combining correlated information may not lead to significant improvement of the results (see, for instance, [18]). Nonetheless, we argue that combining individual similar-type models may also be important. This point of view may be supported by Clements and Hendry [81], where the problem of instability in model selection is addressed, which may cause unreasonably high variability in the final prediction. Instability in model selection may result in inappropriate identification of a unique 'best' model, since different ones – either different in the topology or in the learning methods used –, may perform equally well in prediction. Thus, the identified 'best' model may not necessarily be optimal in forecasting.

The literature on combining methods continues to grow, and many questions have emerged concurrently concerning the advantages and disadvantages of combining two or more models or their estimates. Theoretically, it can be proven that the combining procedure can always outperform or, at least, equals the effectiveness of the 'best' of the individual models, but one does not guarantee the same evidence empirically, mostly due to the presence of noise or the limited number of data available.

Another difficulty is related to the number of individual models in the combination. As the number grows, the resulting error decreases, at least in theory, but the assumptions that the errors are independent or uncorrelated become less valid (see [58]). Another important question that may emerge in the combination paradigm is related to the trade-off between efficiency and effectiveness. If the loss of accuracy is negligible compared to the savings in computational time, then, to some extent, combining is worthwhile. Finally, we may question in which circumstances combining is preferable to finding a single 'best' model. The answer to these questions may not be easy to produce or may be too difficult to verify in practice. It might depend on the specific problem on hand. Armstrong [1] provides methodological guidelines for combining forecasts, introducing some aspects related to either favouring or not the combination paradigm.

This thesis seeks to discuss some alternative approaches (initially introduced in [27]) using the combination paradigm, where, while still using supervised neural networks, one can achieve good, if not better forecasting performance, through efficient recursive estimation and adaptive identification methods. Most of parametric predictive models proposed here – neural or otherwise – are based on time-varying linear parameters that can be efficiently estimated in a recursive manner. Furthermore, we concur with the viewpoint that two or more suboptimal models, linearly composed or linearly combined, may, in general, constitute a better alternative to the optimisation of a single neural network in terms of predictive accuracy, efficiency and robustness. While most of the optimisation problems

reported here are related to time series forecasting, some may be adapted to other problems, such as classification or clustering.

The combination paradigm is explored in different ways, using throughout linear parametric models to yield a final estimate. In general, we can identify three fundamental ways of combining:

— *Model mixing*, where the chosen estimate is computed as a combination of estimates from different models;

— *Model synthesis*, where the chosen estimate is obtained from the (linear) combination of different, partial models, estimated in conjunction or in sequence;

— *Model switching*, where the chosen estimate is selected from the estimates of different models.

We may identify three fundamental contributions along this study using the combination paradigm, namely in the context of predictive model mixing, of predictive model synthesis, and of prescriptive model mixing.

We propose extending the usual framework for linearly combining estimates from different models to cope with the case where the forecast errors from different models are correlated. Only the case for two models is considered, but the idea can be extended to more models. Specifically, a nonlinear term, defined by a point-by-point multiplication of the sequences of individual predictive estimates, is included in the combining expression. We also derive generalisations of the analytical results known for the classic linear combination.

We pay special attention to model synthesis when dealing with heavily nonstationary time series, since we intend to study some methodological issues related to the identification and estimation of hybrid dynamic models. Each of the component models aims to account for particular characteristics detected on the spectra of the time series. Continuing the work started by Silva [66], the hybrid model – composed by a dynamic model and a neural network –, is defined as a particular case of the univariate Unobserved Components model [78]. Further methodological guidelines are also proposed for its identification, specifically optimisation processes based on forecast errors for large horizons.

The combination paradigm is subsequently considered in the context of decision-making or, rather, objective decision support. The identification and estimation of a prescriptive model should be based on more realistic utility functions, rather than on least-squares

based criteria. These utility functions are usually non-differentiable, adding difficulty to the overall process (see [19] for a direct application of neural networks using asymmetric cost functions). We discuss two different ways of introducing model combination in the context of predictive-prescriptive problems: either inferring decisions from combined predictive estimates, or combining prescriptive solutions derived from different forecasting models. We may argue that the former is preferable in most situations, since predictions are usually easier to obtain and optimise than decisions, but that might depend on some issues in the process, namely the time horizon and the degree of asymmetry between excess and default prediction errors. It is shown that either approach can produce the best results, depending on the data on hand and the problem to be solved. Since neither one always outperforms the other, in a given problem both can be tried and compared.

Most of the conclusions achieved in this thesis were assessed by computational experiences. To that end, we entirely developed a MATLAB toolbox encompassing all the models and methods used in this thesis.

1.2 Overview

In Chapter 2, we describe the models and methods to be used in the following Chapters. Our attention is focused mainly in linear stochastic parametric models for time series forecasting problems, estimated through least-squares based methods, namely the Recursive Least-Squares algorithm and its variants. Special attention is given to the state-space model formulation for time-varying parameters, and their estimation through the Kalman filter. We present a brief introduction to artificial neural networks and describe the neural network mainly used in our study: the Gaussian Radial Basis Function Network. An adaptive method is proposed to identify the centres of the basis functions, when working in nonstationary environments.

In Chapter 3, we emphasise the application of supervised neural networks in the context of advanced time series forecasting. Problems raised by heavily nonstationary time series are pointed out, as well as some drawbacks on classical detrending and differencing preprocessing approaches. Alternative preprocessing methods to render the training patterns of neural networks stationary are also mentioned. Namely, we propose a 'prefiltering' (preliminary filtering) methodology as a preferable alternative to classic (deterministic) detrending, by means of stochastic detrending. The objective is to consider a preliminary estimation and removal of a dynamic stochastic model to deal with low-frequency effects, and then using a supervised neural network to account for higher-frequency effects and possibly nonlinear effects.

In Chapter 4, we propose some new ideas to use the model mixing approach. The usual framework for linearly combining estimates from different models is extended to cope with the case where the forecast errors from different models are significantly correlated. Recursive and adaptive expressions are also considered for the cases where the strength relations between those errors are assumed to vary with time, as well as a procedure to revise the combining weight values when new information becomes available.

In Chapter 5, special attention is given to the application of the combining paradigm in the context of decision-making. We discuss two different ways of introducing combination in the context of predictive-prescriptive problems: either inferring decisions from combined predictive estimates, or combining prescriptive solutions derived from different forecasting models.

In Chapter 6, some methodologies proposed and discussed in the previous Chapters are illustrated with two case studies: the Lisbon stock exchange PSI20 index and a time series related to tourism in Madeira, Portugal. In the first case, we intend to apply the stochastic detrending methodology, with a dynamic trend regressive model and a Gaussian radial basis function network as individual components of the hybrid model. In the latter case, we aim to apply, in addition, the model mixing methodology in the context of decision-making.

Finally, in Chapter 7, we summarise the main ideas explored in this thesis, and suggest possible ways of continuing the research initiated here.

CHAPTER 2

On Models and Methods

Contents

Abstract

This Chapter briefly describes models and methods to which we will refer in the following Chapters. We review the basics of linear stochastic parametric models for the time series forecasting problem and discuss their estimation through least-squares based methods. Special attention is given to dynamic stochastic models in state-space representation, encompassing time-varying parameters. We present a brief introduction to artificial neural networks and describe the main neural-type model used in our study: the Gaussian Radial Basis Function Network.

2.1 Basic Concepts

Time series analysis constitutes an important tool in the study and better understanding of phenomena in several areas of practical interest, and, in most cases, its main ultimate goal is prediction (i.e., forecasting). Prediction is viewed as a means of supporting decision-making. Regardless of the depth of our understanding and the validity of our interpretation of the phenomenon in consideration, forecasting consists of extrapolating the identified patterns in the time series to predict future values, so that subsequent decisions are not made subjectively or arbitrarily.

The choice of the set of models and methods for computing forecasts should depend on the information and statistical characteristics of the observations – see, e.g., [15] for a brief catalogue of different time series forecasting methods. There is a generalised concern about the complexity and flexibility of models and methods due to the dynamical complexity of phenomena where time series forecasting is envisaged. Conversely, the need of producing 'quick' and 'good' answers within a short time requires that the models and methods be reliable and computationally efficient. The evolution in time of the system that describes the phenomenon under study needs some degree of monitoring and adaptation, that is, a way to obtain and improve forecasts as new information arrives and, even, the possible revision of the estimation process.

Time series result from observed (or simulated) values collected along time, at regular or irregular intervals, about one or more phenomena. These recorded observations are therefore time-dependent and their temporal order is very important. A regular (in time) single-variable time series can be represented by a vector of real values:

$$\mathbf{y} = \left[y_1, ..., y_k, ..., y_n \right]^T$$

with index k representing the time instant at which the corresponding numeric value was observed. If the data occur or is observed at irregular intervals, the time series is called irregularly spaced or intermittent and the sequence of time stamps is itself a time series. Throughout, we consider only regularly spaced time series.

Informally, we may define a time series to be stationary if its distributional properties, including the mean and variance are all constant over time (see, e.g., [7] for more precise mathematical definitions). Most time series in practice may exhibit trends, seasonal or

cyclical patterns, shifts, transient effects, etc. Figure 2.1 shows four times series exhibiting different nonstationarity patterns. In 2.1 (d), the time series exhibits a trend and seasonal effects as well as cyclical or quasi-cyclical patterns.

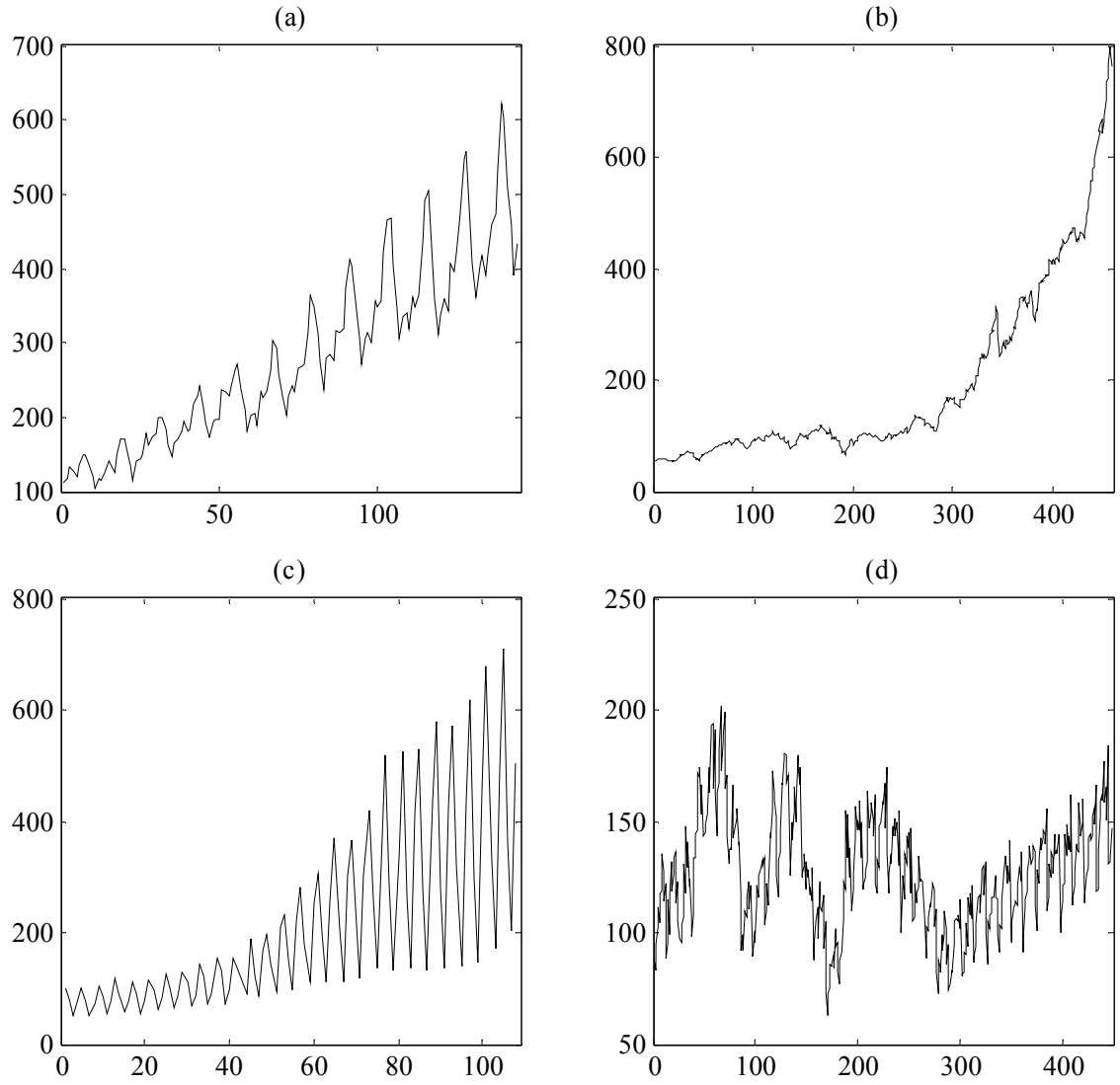

Figure 2.1: (a) Monthly totals (in thousands) of international airline passengers in the U.S.A. (1949-1960); (b) Monthly Index of Stock Prices in the U.S.A. (1959-01 to 1997-04); (c) Quarterly consumption of gas in the U.K. (1960-01 to 1986-04); (d) Monthly totals (in thousands) of houses built in the U.S.A. (1968-01 to 2005-04).

The statistical properties of a time series influence significantly the choice of the type and mathematical structure of the model and the method to be considered in each case. In order to produce forecasts about future events, some assumptions have to be made. We assume that the data consists of a systematic pattern, or signal, with added noise. Prediction is typically based on stochastic parametric models of the form:

$$y_k = f\left(\mathbf{x}_k, \boldsymbol{\theta}_k; \Psi\right) + \varepsilon_k \tag{2.1}$$

The noise or error component, ε_k, is a sequence of uncorrelated random variables from a certain probability distribution, independent of the functional form *f*. Normally, for analytical convenience, this error component is assumed to be white noise, i.e., to have zero mean value and constant variance σ_ε^2, and with no serial correlation, further denoted by $\left(0, \sigma_\varepsilon^2\right)$. Additionally, sometimes it is assumed to be a Gaussian stochastic process: $N\left(0, \sigma_\varepsilon^2\right)$.

The signal, defined by function *f*, is an estimable component and can be either deterministic or stochastic. It has input variables $\mathbf{x}_k$, in general defined by past observations or by exogenous variables, and depends on the unknown parameters $\boldsymbol{\theta}_k$, that should be estimated. Possibly, the signal depends also on a set of *hyperparameters* Ψ – for instance, special parameters that are used to define how parameters $\boldsymbol{\theta}_k$ vary in time.

The stochastic functional in (2.1) can be viewed as a regressive model, where function *f* relates the dependent variable y_k and the independent variable $\mathbf{x}_k$. Nonlinear expressions for *f* can be considered, either in terms of the inputs or in the parameters, with the advantage to introduce flexibility into the model. However, nonlinearity in the parameters may require using nonlinear optimisation procedures for the identification and estimation tasks, hence increasing the difficulty in solving the underlying problem. In the simpler case, the parameters are assumed constant, $\boldsymbol{\theta} \equiv \boldsymbol{\theta}_k$, and function *f* is linear both in the input variables and in the parameters. A class of such kind of models commonly used in practice are the *linear autoregressive* models of order *p*, AR(*p*), defined as

$$y_k = \theta_0 + \theta_1 y_{k-1} + \cdots + \theta_p y_{k-p} + \varepsilon_k \tag{2.2}$$

Nonlinear extensions can be considered, yet preserving linearity in the parameters. The following special cases of more general linear regressive models are particularly used in the context of model combining, as introduced in Chapter 4:

$$\text{(Model 1)} \qquad y_k = \theta_0 + \theta_1 x_{1k} + \theta_2 x_{2k} + \varepsilon_k \qquad \text{(linear model)}$$

$$\text{(Model 2)} \qquad y_k = \theta_0 + \theta_1 x_{1k} + \theta_2 x_{2k} + \theta_3 x_{1k} x_{2k} + \varepsilon_k \qquad \text{(nonlinear model)}$$

They are represented in Figure 2.2, with all parameters set to 2. Both models are linear in the parameters, and thus the estimation effort is much smaller, as discussed later in this chapter.

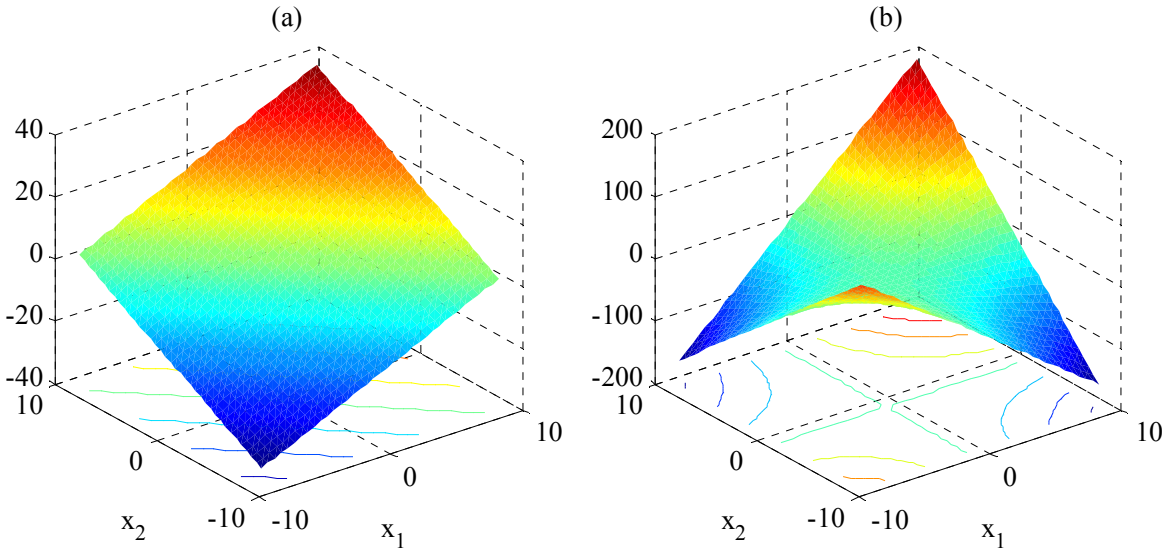

Figure 2.2: Graphical representation of linear parametric models: (a) Model 1; (b) Model 2.

Models that are linear in the parameters can be written as

$$y_k = \mathbf{u}_k^T \boldsymbol{\theta} + \varepsilon_k$$

where the column vector $\mathbf{u}_k = \begin{bmatrix} 1 & u_{k1} & \cdots & u_{km} \end{bmatrix}^T$ normally contains $m < n$ input variables, that can be interpreted as *regressors*, and the parameter vector $\boldsymbol{\theta} = \begin{bmatrix} \theta_0 & \theta_1 & \cdots & \theta_m \end{bmatrix}^T$ includes $m+1$ unknown parameters. For the AR(p) model given by (2.2) it is sufficient to define $m = p$ and $u_{kj} = y_{k-j}$. Radial Basis Function Networks, introduced in Section 2.5.2,

among other nonlinear models, can be seen as linear-in-the-parameters models, once the hyperparameters are defined, or assumed known.

In order to allow for nonstationarity in the time series y_k, the (dynamic) parameter vector $\boldsymbol{\theta}_k$ can be characterised by stochastic time-varying parameters, with each time-varying parameter defined as a nonstationary stochastic variable, which in turn may be approximated by a linear time-varying parameter model (see Section 2.4).

2.2 Model Specification and Optimisation

To assess model accuracy, we must have some means of measuring or evaluating how good the forecast estimates, and thus the model itself are.

Model optimisation mainly depends on the degree of accuracy that is required, the computational efficiency or the cost for producing the forecasts, the degree of complexity of the model, the time horizon, and the available data. The degree of accuracy may not always be the most imperative goal, since it usually raises substantially the cost for producing the forecasts. Sometimes, simpler but less accurate models may be preferable to more accurate but also more complex ones if the loss in accuracy is not too critical. The forecasting horizon considered is essential too, since optimal short-term and optimal long-term forecasts usually require different optimisation criteria, under the same forecasting method. Furthermore, we assume that the data available is long enough so that recursive and adaptive methods can be employed.

For measuring model accuracy, we must firstly define what the prediction errors produced by the estimated model are. The estimate $\hat{y}_{k+h|k}$ is defined as the *h*-steps-ahead prediction, that is, the prediction of the value y_{k+h}, computed with origin at instant k – i.e., using only information up to, and including, k – and with time horizon $h \geq 1$. The value $\hat{y}_k$ is an estimate for the time instant k and sometimes is used as an abbreviation of $\hat{y}_{k|k-1}$. The 1-step-ahead forecast errors can be denoted as

$$e_k \equiv e_{k|k-1} = y_k - \hat{y}_k$$

and are the most commonly considered when optimising a model. But performance measures can also be based on prediction errors for larger time horizons.

Assuming that negative and positive forecast errors are equally undesirable and that large absolute errors are especially to be avoided, it makes sense to choose forecasts produced by a model with minimum (predictive) *mean squared error* (MSE), based on *h*-steps-ahead forecast errors:

$$MSE(h) = \frac{1}{n - n_0} \sum_{k=n_0+1}^{n} e_{k|k-h}^2$$

Here, n_0 is the number of initial errors to be ignored, likely to be biased due to initialisation tasks in the learning method, whenever recursive or adaptive estimation is used. Other common measures are the *mean absolute error* or *mean absolute deviation* (MAD):

$$MAD(h) = \frac{1}{n - n_0} \sum_{k=n_0+1}^{n} \left| e_{k|k-h} \right|,$$

the *root mean squared error* (RMSE):

$$RMSE(h) = \sqrt{MSE(h)}$$

and the *mean absolute percentage error* (MAPE):

$$MAPE(h) = \frac{100}{n - n_0} \sum_{k=n_0+1}^{n} \frac{\left| e_{k|k-h} \right|}{y_k}$$

The min RMSE criterion (equivalent to min MSE) is possibly the most widely used, namely in functional approximation and regression problems, because it leads to simpler mathematical solutions.

Any of the performance measures described above can be considered to compare different forecasting methods and models. In simpler cases, their minimisation can be accomplished by a grid search, where different values of predefined hyperparameters that appear in the model specification or learning method are compared.

2.3 Least-Squares Estimation

Almost everywhere in the thesis, we consider linear parametric models, characterised by either constant or time-varying parameters. Therefore, most of the estimation procedures are carried out through the (recursive) least-squares algorithm or by its variants, which are described below and in the following Section.

Let $\mathbf{y} = \left[y_k \right]$ be a sequence of n process observations, and $\left[\mathbf{u}_k \right]$ a sequence of regressor vectors of length $m+1$. We first assume that all the above are functionally related through the (*static*) linear parametric model:

$$y_k = \mathbf{u}_k^T \boldsymbol{\theta} + \varepsilon_k \tag{2.3}$$

The estimation of the unknown parameter vector $\boldsymbol{\theta}$ from historical data can be viewed as a functional approximation problem, where the fitted model is 'close' to the observations. Usually, this is formulated as a least-squares problem, originally described by Gauss, where one seeks to minimise the sum of squared errors (SSE), that is,

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} J\left(\boldsymbol{\theta}\right) \tag{2.4}$$

with

$$J\left(\boldsymbol{\theta}\right) = \sum_{k=1}^{n} \left(y_k - \mathbf{u}_k^T \boldsymbol{\theta} \right)^2 \tag{2.5}$$

The estimate $\hat{\boldsymbol{\theta}}$ can be computed straightforwardly using all available pairs $\left\{ \left(y_k, \mathbf{u}_k \right) \right\}$, after setting the system of normal equations:

$$\left(\sum_{k=1}^{n} \mathbf{u}_k \mathbf{u}_k^T \right) \hat{\boldsymbol{\theta}} = \sum_{k=1}^{n} y_k \mathbf{u}_k$$

The unique solution, if it exists, is then given by

$$\hat{\boldsymbol{\theta}}_n = \left(\sum_{k=1}^{n} \mathbf{u}_k \mathbf{u}_k^T \right)^{-1} \sum_{k=1}^{n} y_k \mathbf{u}_k \tag{2.6}$$

This includes the computation of a $\left((m+1)\times(m+1)\right)$ matrix inversion. Equation (2.6) defines the well-known *Ordinary Least-Squares* (OLS) algorithm.

Models that are not linear in all the parameters cannot be estimated using this closed form expression. In that case, the nonlinear parameters have to be estimated using either a grid search or a nonlinear optimisation search procedure towards the minimisation of (2.5).

We note that the solution given by (2.6) can be written as

$$\hat{\boldsymbol{\theta}} = \mathbf{U}^{+}\mathbf{y}$$

where $\mathbf{U}^{+} = \left(\mathbf{U}^T\mathbf{U}\right)^{-1}\mathbf{U}^T$ is the Moore-Penrose generalised matrix inverse, or *pseudoinverse* matrix of $\mathbf{U} = \left[\mathbf{u}_1 \mid \cdots \mid \mathbf{u}_n\right]^T$. This solution only exists provided $\mathbf{U}^T\mathbf{U}$ is non-singular [9].

The OLS algorithm assumes that the available data is sufficient to capture the evolution of the observed phenomenon. In cases where additional information is about to be on hand, one should be interested in revising (updating) the estimates without having to repeat the whole procedure for all the data. Therefore, to speed up significantly the computation of the overall process, online or recursive estimation procedures should be employed.

Denoting by $\hat{\boldsymbol{\theta}}_k$ the estimates based on the observation up to and including y_k, a general algorithm for a forward processing recursive estimation procedure can be defined as

- Specify initial values, $\hat{\boldsymbol{\theta}}_0$
- Specify values for hyperparameters, Ψ
- Repeat for $k = 1, 2, \ldots$:
  - Read new observation, y_k
  - Compute $\hat{\boldsymbol{\theta}}_k := f\left(y_k, \hat{\boldsymbol{\theta}}_{k-1}; \Psi\right)$

The *Recursive Least-Squares* (RLS) algorithm is an 'online' version of the OLS method. Assuming the unknown parameters are constant in time, the main objective of this method

is to efficiently produce a sequence of estimates of the parameter vector, $\hat{\boldsymbol{\theta}}_k$, asymptotically equivalent to the OLS estimates.

The RLS algorithm is defined by equations (see, e.g., [60])

$$\begin{cases} \hat{\boldsymbol{\theta}}_k = \hat{\boldsymbol{\theta}}_{k-1} + \mathbf{k}_k d_k \\ \mathbf{P}_k = \mathbf{P}_{k-1} - \mathbf{k}_k \mathbf{u}_k^T \mathbf{P}_{k-1} \end{cases}$$

where

$$\begin{cases} d_k = y_k - \mathbf{u}_k^T \hat{\boldsymbol{\theta}}_{k-1} \\ \mathbf{k}_k = \mathbf{P}_{k-1}\mathbf{u}_k \Big/ \left(1 + \mathbf{u}_k^T \mathbf{P}_{k-1}\mathbf{u}_k\right) \end{cases} \tag{2.7}$$

The values $\{d_k\}$ are known as the *innovations* and represent the 1-step-ahead prediction errors, $d_k \equiv e_{k|k-1}$. The vectors $\{\mathbf{k}_k\}$ are called the algorithm *gains* and determine how much weight should be given to the most recent forecast errors.

As much as the well-known error backpropagation algorithm – commonly used for training Multilayer Perceptrons and other neural models – is an online version of the steepest descent (negative gradient) method, RLS can be compared to the Gauss-Newton method, and other second order search algorithms, with matrices $\mathbf{P}_k$ playing a similar role to that of the inverse of the Hessian matrix [64]. Despite its computational simplicity, RLS can also be used to approach nonlinear least-squares problems efficiently. The estimates within matrices $\mathbf{P}_k$ weight the $m+1$ directions of the search space unequally, therefore providing second-order information about the search surface, and thus guiding the estimation procedure.

The basic RLS procedure can be elaborated into more complex variants, namely dynamic RLS procedures, which, in turn, can be seen as particular cases of the Kalman filter (see, e.g. [6, 45, 60, 77]). These alternatives are meant for cases where the parameters are assumed to vary in time.

2.4 Time-Varying Parameters

The linear parametric model in (2.3) is static, i.e., it assumes that the unknown parameters $\boldsymbol{\theta}$ are constant in time. This implies that each observation (recent or past) has the same weight in determining the estimates. While the assumption of fixed data distribution is occasionally appropriate, it is in most cases unrealistic and one would expect that the parameters change over time and are constant at most locally. For this reason, we want to consider methods capable of tracking possible variations in time of those parameters. The general model is now:

$$y_k = \mathbf{u}_k^T \boldsymbol{\theta}_k + \varepsilon_k$$

There are several extensions to RLS algorithm capable to deal with time-varying parameters. In any case, we need to modify the basic RLS algorithm so to prevent that the updates to the estimates become negligible due to $\mathbf{P}_k \to \mathbf{0}$ (the null matrix).

In order to catch time variations in the parameters, we can start by reducing the importance of past observations in deriving their estimates and use an exponential weighting approach on the data. In this approach, the Least-Squares loss function given by (2.5) is replaced by an *Exponentially Weighted Least-Squares* (EWLS) one. The estimates can then be chosen so to minimise

$$J_{EWLS}(\boldsymbol{\theta}) = \sum_{k=1}^{n} \lambda^{n-k} \left(y_k - \mathbf{u}_k^T \boldsymbol{\theta} \right)^2$$

where λ $\left(0 < \lambda \leq 1\right)$ is a forgetting coefficient. A minor modification of the RLS algorithm can be employed (see, e.g., [6, 69]), where the updating equation for $\mathbf{P}_k$ is now

$$\mathbf{P}_k = \left(\mathbf{P}_{k-1} - \mathbf{k}_k \mathbf{u}_k^T \mathbf{P}_{k-1} \right) \Big/ \lambda$$

and the respective gains $\mathbf{k}_k$ are also modified to

$$\mathbf{k}_k = \mathbf{P}_{k-1} \mathbf{u}_k \Big/ \left(\lambda + \mathbf{u}_k^T \mathbf{P}_{k-1} \mathbf{u}_k \right)$$

For $\lambda = 1$, we have the basic RLS algorithm. Other approaches can equally be considered, where the parameter variations are explicitly modelled, such as dynamic regression based models.

Although more complex stochastic dynamic processes can be used to model time-varying parameters, reasonable results can be achieved by simply considering a multivariate version of the *random walk* process:

$$\boldsymbol{\theta}_k = \boldsymbol{\theta}_{k-1} + \boldsymbol{\eta}_k, \quad \boldsymbol{\eta}_k \text{ iid } (0, \mathbf{Q})$$

where the noise vectors $\boldsymbol{\eta}_k$ are assumed uncorrelated with the observation noise scalars ε_k, with zero mean and covariance positive definite matrix $\mathbf{Q}$. This matrix is usually assumed to be diagonal, thus imposing zero correlation between the different elements of each noise vector $\boldsymbol{\eta}_k$. The presence of noise variance in the parameters prevents progressive insensibility of the RLS to new information.

The estimates of the error covariance matrices $\mathbf{P}_k$ of the corresponding time-varying parameters $\hat{\boldsymbol{\theta}}_k$ are now computed as

$$\mathbf{P}_k = \mathbf{P}_{k-1} - \mathbf{k}_k \mathbf{u}_k^T \mathbf{P}_{k-1} + \mathbf{Q}_r$$

where $\mathbf{Q}_r$ is a *noise covariance ratio matrix*, defined by

$$\mathbf{Q}_r = \frac{\mathbf{Q}}{r}$$

with $r = \mathrm{Var}(\varepsilon_k)$. The respective gains $\mathbf{k}_k$ are computed as in (2.7). The elements in $\mathbf{Q}_r$ measure the variations in the shocks that drive the parameters $\boldsymbol{\theta}_k$, as compared to the variance of the noise error ε_k.

This is sometimes known as the *Covariance-Addition* version of the RLS algorithm (RLS-CA) (see, e.g., [6]). The elements in the diagonal of matrix $\mathbf{Q}_r$ are called *noise variance ratios* (NVRs) and represent hyperparameters that may, or rather should be optimised.

Discrete-time linear dynamic systems can be conveniently described in state-space representation. Its applicability to forecasting problems is of great importance since it can

encompass time series models with quite different characteristics. State-space models are estimated through recursive methods, most notably the *Kalman filter* [38, 39], developed in the context of control systems engineering.

Modelling time series through state-space representation is based on the assumption that the Markov property holds for the observed stochastic process. This property means that the future of the process does not depend on its past states, but solely on the current (most recent) state, i.e., the current state encompasses all the information sufficient to predict the future.

A state-space representation consists of two main equations, the *observation* or *measurement equation* – which describes how the observations are generated from a given state vector of unobserved variables –, and the *system equation* – which describes the system dynamics:

$$\begin{cases} y_k = \mathbf{u}_k^T \boldsymbol{\theta}_k + \varepsilon_k \\ \boldsymbol{\theta}_k = \mathbf{F}\boldsymbol{\theta}_{k-1} + \boldsymbol{\eta}_k \end{cases}$$

In the above, y_k is an observation value, $\boldsymbol{\theta}_k$ is an unobservable *state vector* that describes the state of the system at time *k*, $\mathbf{u}_k$ is a known *input vector*, ε_k (known as the *measurement noise*) is a white noise process with variance $r = \sigma_\varepsilon^2$, and $\boldsymbol{\eta}_k$ (known as the *process* or *system noise*) is a vector white noise process with covariance matrix $\mathbf{Q}$. These two error terms – the measurement and the process noises – are assumed uncorrelated at all time lags. The *transition matrix* $\mathbf{F}$, assumed known, rules the updating process of the state vector from a previous time-step. In more complex versions, the state-space formulation can assume this matrix to be time-dependent, or it can include also a transition matrix for the system noise vector. Furthermore, the formulation can be extended to the multivariate case, as well as to nonlinear versions.

The Kalman filter essentially consists of two distinct phases: the prediction phase and the measurement update phase. The prediction phase can formally be described as finding $\hat{\boldsymbol{\theta}}_{k|k-1}$ from $\hat{\boldsymbol{\theta}}_{k-1|k-1}$ and the system equation:

$$\begin{cases} \hat{\boldsymbol{\theta}}_{k|k-1} = \mathbf{F}\hat{\boldsymbol{\theta}}_{k-1|k-1} \\ \mathbf{P}_{k|k-1} = \mathbf{F}\mathbf{P}_{k-1|k-1}\mathbf{F}^T + \mathbf{Q}_r \end{cases}$$

In the update phase, the measurement information from the previous phase is used to refine those predictions to obtain 'filtered estimates', that is, finding $\hat{\boldsymbol{\theta}}_{k|k}$ from $\hat{\boldsymbol{\theta}}_{k|k-1}$ and the measurement equation:

$$\begin{cases} \hat{\boldsymbol{\theta}}_{k|k} = \hat{\boldsymbol{\theta}}_{k|k-1} + \mathbf{k}_k d_k \\ \mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{k}_k \mathbf{u}_k^T \mathbf{P}_{k|k-1} \end{cases}$$

where

$$\begin{cases} d_k = y_k - \mathbf{u}_k^T \hat{\boldsymbol{\theta}}_{k|k-1} \\ \mathbf{k}_k = \mathbf{P}_{k|k-1}\mathbf{u}_k \Big/ \left(1 + \mathbf{u}_k^T \mathbf{P}_{k|k-1}\mathbf{u}_k\right) \end{cases}$$

An example of a state-space representation describing a simple dynamic linear stochastic system is the *dynamic trend regressive* (DTR) model, where the observations are modelled by a simple time-varying parameter with added measurement noise:

$$y_k = t_k + \varepsilon_k$$

Many different system equations can be devised to describe the trend parameter. For our purposes, we will consider the particular case of a versatile but yet simple model that has the ability to reproduce a wide range of patterns of variation in time (see Figure 2.3), namely the *first-order integrated autoregressive* (IAR(1)) process, defined by

$$\begin{cases} t_k = t_{k-1} + s_{k-1} \\ s_k = \alpha s_{k-1} + \eta_k \end{cases}$$

where the noise inputs $\{\varepsilon_k\}$ and $\{\eta_k\}$ are assumed to be uncorrelated. This model, similar to the *smoothed random walk* (SRW) model [77], encompasses, as limit or particular cases, the already mentioned linear regression model (corresponding to linear detrending) and the RW model (for which, first-order differencing indeed achieves whiteness in the residuals). The complete DTR model can thus be rendered into state-space form by setting:

$$\mathbf{u}_k = \begin{bmatrix} 1 & 0 \end{bmatrix}^T, \quad \boldsymbol{\theta}_k = \begin{bmatrix} t_k & s_k \end{bmatrix}^T, \quad \mathbf{F} = \begin{bmatrix} 1 & 1 \\ 0 & \alpha \end{bmatrix}, \quad \mathbf{Q} = \begin{bmatrix} 0 & 0 \\ 0 & \sigma_\eta^2 \end{bmatrix}, \quad r = \sigma_\varepsilon^2$$

The two sequences of parameters to be estimated, t_k and s_k, are usually named *levels* and *slopes*, respectively. The slope parameter is defined by a first-order autoregressive model and represents the first-differences of the level parameter. The *smoothing coefficient*, α, is a hyperparameter that defines the degree of damping in the level parameter, and normally one assumes that $0 < \alpha < 1$. Another hyperparameter that should also be identified is the variance σ_η^2. Figure 2.3 shows different patterns that are produced by a DTR-IAR(1) model by varying the values of its hyperparameters.

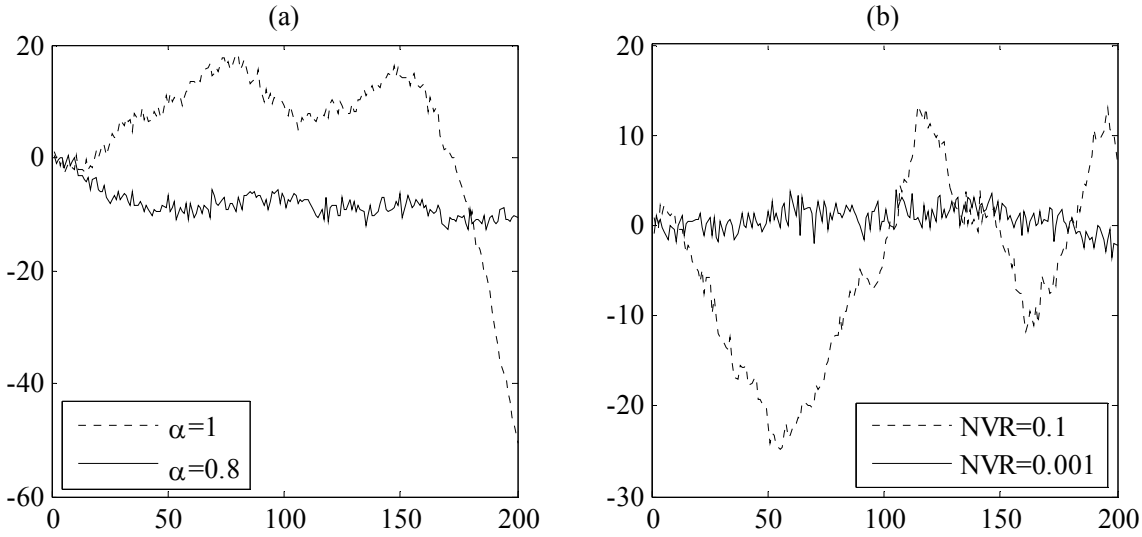

Figure 2.3: Different patterns produced by a DTR-IAR(1) model when: (a) varying α, while setting the NVR to 0.01; (b) varying the NVR, while setting α to 0.9.

By setting the smoothing coefficient to the limit case $\alpha = 0$ we get the simple random walk (RW) process, representing a locally constant level, while with $\alpha = 1$ we obtain the *integrated random walk* (IRW) process, characterised by a locally constant linear growth rate. Moreover, with $\alpha > 1$ the process is characterised by an exponential growth.

2.5 Artificial Neural Networks

2.5.1 Introduction

The literature on artificial neural networks is diverse, including several textbooks, such as [5, 25, 37, 62, 63], and a plethora of scientific papers, many of them biased towards the interests of a particular scientific community – computer science, physics, statistics, control engineering, etc.

Artificial neural networks are a sophisticated modelling paradigm with the potential to solve very complex problems. Mathematically speaking, a *supervised neural network* is essentially a nonlinear function defined in a multivariate input-output space. It is composed of a large number of highly interconnected processing elements or units (neurons) operating together to solve specific problems. The weights associated to the links between the network nodes define the parameters of the model.

Artificial neural networks learn from observed examples, through adaptation rules of the values of those weights. There are many types of learning rules but they can be classified in three broad categories: supervised learning, unsupervised learning and reinforcement (or graded) learning. In *supervised learning*, there must be input-output pattern examples (the *training set*). The input patterns (the *explanatory* or *independent variables*) are assumed to be related to the output patterns (the *dependent variables*) in a causal structure chain. In *unsupervised learning*, as opposed to supervised learning, the network is autonomous in the sense that it tries to reflect, in its output, the properties found in the input patterns. *Reinforcement learning* is similar to supervised learning, except that the correct input/output pairs are never presented, nor suboptimal actions explicitly corrected (see, e.g., [67]). The network performance is measured by means of grades, scores or rewards over some sequence of inputs.

Supervised neural networks can be viewed as multivariable and multidimensional nonlinear functions. The identification and estimation of such a model can be interpreted as an approximation mapping problem from a multidimensional input set to a multivariable output set. The generalisation consists of using that surface to interpolate or extrapolate the data set. They are mainly used in function approximation, time series prediction, classification, pattern recognition and control.

The two most commonly used types of feedforward supervised networks are the Multilayer Perceptron (MLP) and the Radial Basis Function Network (RBFN). Both models have been proven to be "universal function approximators" [56], which means that, given enough data, the underlying function can be approximated with arbitrary accuracy. MLPs and RBFNs have more in common than what is usually assumed [65], but the more essential differences led us to prefer using RBFNs in our work. In fact, their identification and estimation can be accomplished more efficiently than in the case of MLPs.

2.5.2 Gaussian Radial Basis Function Network

Radial basis function networks have their origins in techniques for performing exact interpolation in a multidimensional space. Earlier work on this subject can be found in the surveys of Powell [61] and Light [43]. Radial basis function models have appeared in the design of neural networks in the late 80's [8, 49] and then employed in nonlinear systems identification and time series prediction, as in [44, 68].

The structure of a RBFN consists of three separate layers: the input layer, the hidden layer and the output layer. The input layer is the set of source, or sensor nodes, where the data are just forwarded to the hidden layer nodes without any kind of processing. The transformation from the input space to the hidden-unit space is nonlinear while the transformation from the hidden-unit space to the output space is linear. This fact plays a fundamental role in the adopted learning strategy.

Although the structure of a RBFN (see Figure 2.4) is very similar to that of most MLPs considered in practice, the hidden-units act as basis functions to characterise the partitions of the input space.

Among other possible choices, the radial basis function most used is the Gaussian function. These models can be presented in two forms, normalised or unnormalised. The difference between them is that each unit of the former is normalised by the sum of all Gaussian units of the latter one. The outputs of the hidden layer depend on the distances between the network input and the 'centres' of the basis functions. As the input moves away from a given centre, the unit output goes rapidly to zero. Because the neurons in the network have localised receptive fields they don't create a global response, in contrast to the standard multilayer networks, where sigmoid functions are commonly used. The main neural networks presented in this thesis are single-output unnormalised Gaussian RBFNs. The extension to networks with multiple outputs is relatively straightforward.

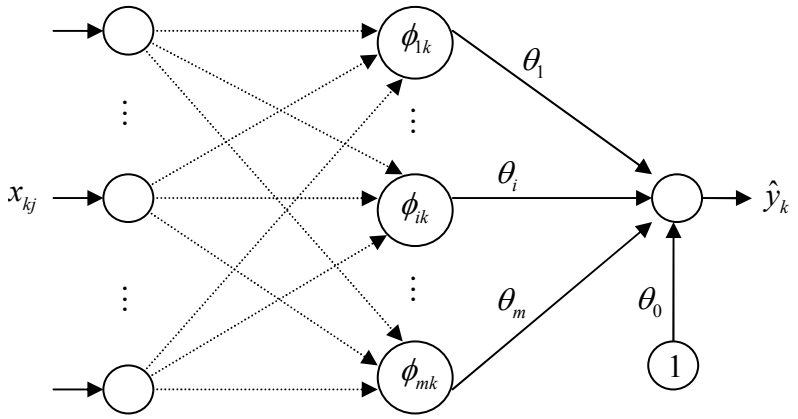

Figure 2.4: Structure of a single-output RBFN.

Given a set of paired input-output patterns $\{(y_k, \mathbf{x}_k)\}$, $1 \le k \le n$, a single-output RBFN is viewed as a linear combination of the outputs produced by a number of radially symmetrical activation functions (the hidden-units), which are nonlinear in the inputs:

$$y_k = \theta_0 + \sum_{i=1}^{m} \theta_i \phi_{ik} + \varepsilon_k \tag{2.8}$$

where $\theta_0, \theta_1, ..., \theta_m$ are linear parameters (the connection weights), the sequence $\{\varepsilon_k\}$ is assumed to be a white noise process, and $\{\phi_{ik}\}$ are the basis functions. Gaussian basis functions (see Figure 2.5) are defined by

$$\phi_{ik} = \phi(\mathbf{x}_k; \mathbf{c}_i, \sigma_i) = \exp\left(-\frac{\|\mathbf{x}_k - \mathbf{c}_i\|^2}{2\sigma_i^2}\right), \quad 1 \le i \le m$$

The centres, $\mathbf{c}_i$, and the widths, σ_i, are model hyperparameters that have to be identified. This is usually accomplished through appropriate heuristics – several of them were compared in [12, 13] – as it is unpractical to approach it as a nonlinear optimisation problem in a very high-dimensional space.

Equation (2.8) can be rewritten using matrix notation similar to (2.3), where the regressors are now defined as $\mathbf{u}_k = \begin{bmatrix} 1 & \phi_{1k} & \cdots & \phi_{mk} \end{bmatrix}^T$.

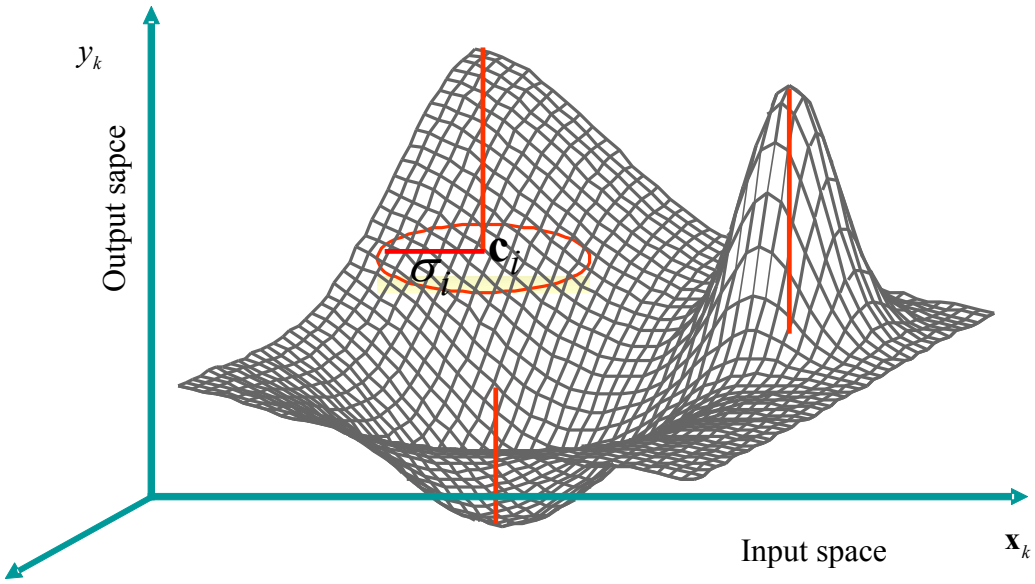

Figure 2.5: Graphical representation of a single-output Gaussian RBFN with 3 units.

Supervised neural networks can be used as nonlinear autoregressive models, with the input patterns, $\mathbf{x}_k$, built from sequences of observations of a time series:

$$\mathbf{x}_k = \begin{bmatrix} y_{k-h} & \cdots & y_{k-2} & y_{k-1} \end{bmatrix}^T$$

The network outputs are then viewed as predictive estimates, in particular, 1-step-ahead forecasts, $\hat{y}_k \equiv \hat{y}_{k|k-1}$. This scheme can be easily adapted to longer horizons or to hybrid – causal and autoregressive – models.

2.5.3 Identification and Estimation Methodologies

A key aspect of RBFNs is the distinction between the tasks performed in the hidden layer and the output layer, as mentioned previously. This leads to different learning strategies in the design of a RBFN, and that depends on how the centres are determined: either a supervised learning process is employed to obtain the weights, the centres and widths of the radial basis functions, or the identification and estimation of a RBFN is accomplished in two learning stages – hybrid training –, where the basis function centres and widths are first determined and then the weights optimised.

In supervised training, the linear parameters and the hyperparameters associated to the Gaussian basis functions are determined in only one stage in a supervised manner. Nonlinear optimisation methods have to be employed in order to train a RBFN with this scheme due to the nonlinear nature of the centres and widths. For instance, Poggio and Girosi [59] have suggested methods based on the gradient descent algorithm to implement supervised learning of the centre locations of the Gaussian basis functions.

According to Wettschereck and Dietterich [72], by employing supervised methods, RBFNs may be able to substantially exceed the generalisation performance of sigmoidal networks, including MLPs. However, 'fully-supervised' learning feedforward neural networks lack some ability to characterise nonstationary distributed data sets. Many real-world problems are based not only on highly nonlinear but also highly nonstationary data, particularly in time series prediction which is studied in the next chapter. For that reason, we discard this approach in favour of the next one in all of our study.

In hybrid training, firstly, the (nonlinear) hyperparameters (centres and widths) are identified through an unsupervised learning heuristic method and, in a second stage, the linear parameters are estimated in a supervised manner. The architecture of the network, namely the number of inputs, p, and the number of radial basis units, m, have also to be defined early in the identification stage. Usually, these values are chosen from a set of predefined values.

In the simplest case, the locations of the centres may be chosen randomly from the training data. Lowe [47] considers that this is a 'sensible' approach if the training data is distributed in a representative manner for the problem on hand. A common and more adequate procedure is to choose the location of the centres through the k-means clustering algorithm. In most cases reported in the literature, clustering is performed over the input patterns [46, 49]. However, the clustering procedure can be performed in the complete input-output space [10], giving a more precise description of the distribution of the data points. In this case, the final centres are given by the projection of the cluster centres into the input space. Bagirov *et al*. [2] have approached the clustering problem via nonsmooth and global optimisation methods.

The widths of Gaussian radial basis functions are frequently determined by heuristical formulae, and some of them were studied and compared by Carmo and Rodrigues [12]. One of the most commonly used is the k-nearest neighbours method, as proposed by Moody and Darken [49], where each width is computed as the mean distance between the respective centre and its k-nearest neighbours:

$$\sigma_i = \frac{1}{k\sqrt{2}} \sum_{j=1}^{k} \left\| \mathbf{c}_i - \mathbf{c}_{j(i)} \right\|, \quad 1 \le i \le m$$

where $\left\{ \mathbf{c}_{j(i)} \right\}$ are the k neighbour centres nearest to $\mathbf{c}_i$.

A fixed value, the same for all widths, $\sigma_i = \sigma$, can also be considered. This common value can be optimised, but some authors provide heuristical formulae: in Haykin [37] it is based on the maximum distance between the centres:

$$\sigma = \frac{1}{\sqrt{2m}} \max_{j,l=1,\ldots,m} \left\| \mathbf{c}_i - \mathbf{c}_l \right\|$$

and in Orr [54] it is based on the maximum distance between the input patterns:

$$\sigma = \frac{1}{2\sqrt{2}} \max_{j,l=1,\ldots,m} \left\| \mathbf{x}_i - \mathbf{x}_l \right\| \tag{2.9}$$

Both schemes help prevent the surface defined by the Gaussian basis functions over the input space to be, either too flat or too rough.

Once the centres and widths are identified and fixed, the linear parameters of the model can be efficiently estimated in a supervised manner.

Conversely, in real-world optimisation problems, where huge collections of data are available and continuously observed, we shall consider recursive estimation procedures so that the model parameters in radial basis function networks can be efficiently estimated, even in the limit case, where they are assumed to be constant in time.

Denoting by $\hat{\boldsymbol{\theta}}_k$ the estimate (computed at time k) of the unknown vector of parameters $\boldsymbol{\theta}_k$, based on the observations up to and including y_k, the general updating scheme for a forward processing recursive estimation procedure is

$$\hat{\boldsymbol{\theta}}_k := f\left(y_k ; \hat{\boldsymbol{\theta}}_{k-1} ; \Psi \right)$$

where Ψ is the set of model hyperparameters.

Most often, $\hat{\boldsymbol{\theta}}_k$ is updated on the basis of the one-step-ahead prediction error, e_k, and the gain vector, $\mathbf{k}_k$,

$$\hat{\boldsymbol{\theta}}_k = \hat{\boldsymbol{\theta}}_{k-1} + \mathbf{k}_k e_k = \mathbf{k}_k y_k + \left(\mathbf{I} - \mathbf{k}_k \mathbf{u}_k^T\right)\hat{\boldsymbol{\theta}}_{k-1} \tag{2.10}$$

seeking to minimise the Least-Squares loss function. The scheme given by (2.10) encompasses several well-known algorithms, including the RLS and its variants, particularly the 'dynamic RLS', or the RLS-CA, and the Kalman filter.

More general schemes can be considered, especially if the sequence of training patterns is nonstationary, and optimal estimation can then be accomplished via the Kalman filter. However, for heavily nonstationary data, it may be preferable to preprocess the data, or the training patters, as described in Chapter 3.

As an illustrative example of optimising and estimating a Gaussian RBFN, in terms of the least-squares criterion, we start by considering a simulated time series generated by the well-known logistic map (see Figure 2.6):

$$y_k = 4y_{k-1}\left(1 - y_{k-1}\right)$$

with $y_0 = 0.9$. The data, although strongly nonlinear, is noise-free and can be predicted rather well using a RBFN.

For that purpose, we have compared several RBFNs with different architectures and different localisations and respective widths for the centres of the Gaussian functions in terms of the RMSE measure based on one-step-ahead forecast errors. The parameters were estimated using the RLS algorithm. Specifically, we have compared 320 different models with the following predefined choices for the 3-tuple $\left(p, m, b\right)$:

$$p \in \{1, 2, \ldots, 8\}; \quad m \in \{1, 2, \ldots, 8\}; \quad b \in \{-1, -0.5, \ldots, 1\}$$

where $s = 10^b \sigma$ is the width, equal for all units, and σ is computed using Equation (2.9). The centres were obtained by the *k*-means clustering algorithm. The best results attained were $p^* = 1$, $m^* = 5$ and $s^* = 0.3535$ (corresponding to $b^* = 0$) for which $RMSE(1)^* = 0.0017$.

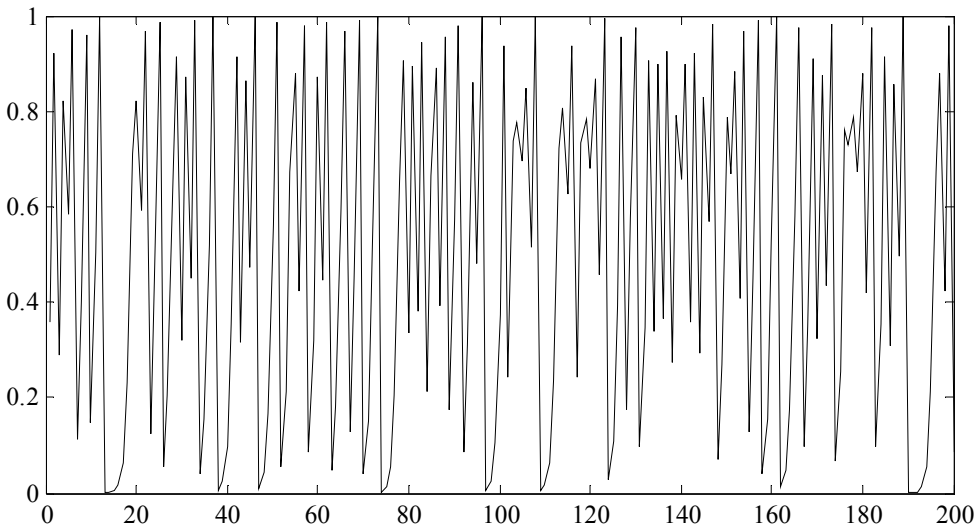

Figure 2.6: Logistic map.

From Figure 2.7 one can notice that different values for those hyperparameters can influence drastically the performance of the network, so optimisation of those variables is highly recommended.

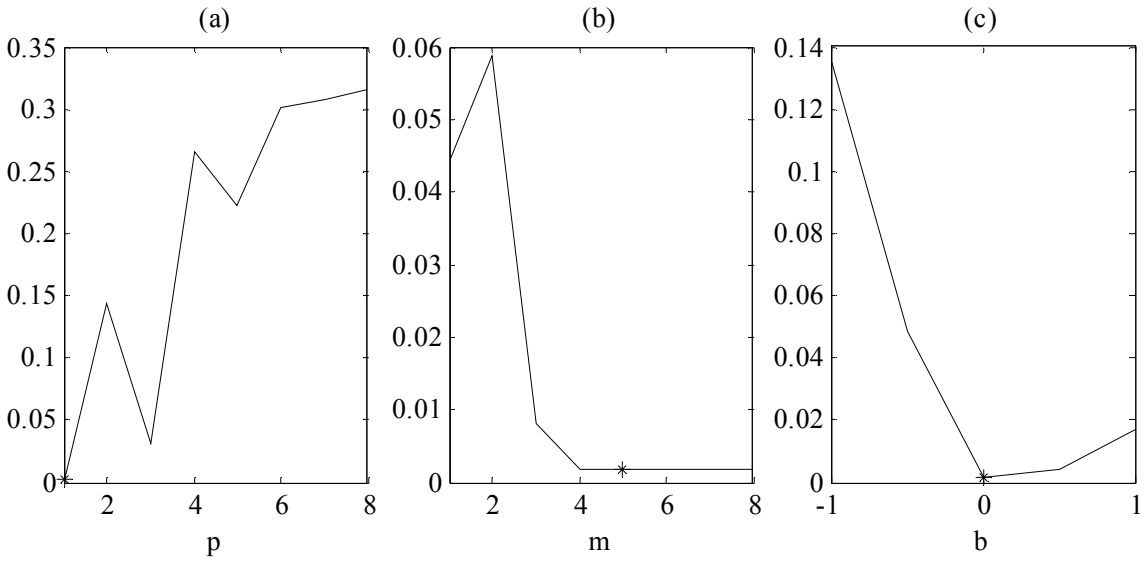

Figure 2.7: Performance of the RBFN with respect to: (a) the number of inputs; (b) the number of RBF units; (c) the magnitude of the widths.

In hybrid training, the centres and widths of the Gaussian basis functions are considered to be fixed during the estimation process. In order to circumvent this problem with nonstationary data, some heuristics can be devised to continuously adapt those hyperparameters in an online learning process, as an additional way of progressively revising the model by incorporating the information of newly observed data.

A general framework for adapting the centres, given a new input pattern, $\mathbf{x}_{k+1}$, is

$$\mathbf{c}_{i,k+1} = \mathbf{c}_{i,k} + \alpha_{i,k}\left(\mathbf{x}_{k+1} - \mathbf{c}_{i,k}\right) = \alpha_{i,k}\mathbf{x}_{k+1} + \left(1-\alpha_{i,k}\right)\mathbf{c}_{i,k}$$

where $\alpha_{i,k}$ is a learning rate defined in the interval $(0,1)$.

There are several versions of this scheme and suggestions for choosing values for $\alpha_{i,k}$. One of the simplest is the *sequential k*-means clustering, where only the nearest centre to $\mathbf{x}_{k+1}$ is adapted. Some proposals for the value of the learning rate were compared in [12].

With selective adaptation there is the risk that some centres may never have the chance to be adapted. Therefore, it seems sensible to extend the adaptation process to more than one unit or even to all units. In the literature, namely in the context of competitive learning, there are many clustering procedures that can be used to continuously adapt the centres of the Gaussian units in an online learning framework – see, for example, [74].

Here, we propose extending the application of the sequential *k*-means algorithm to all centres. Our idea is to allow that all centres follow the variations in the distribution of the input data. Let $\Delta\mathbf{c}_{i,k}$ be the adaptation step associated to the centre $\mathbf{c}_i$ given the new input pattern $\mathbf{x}_{k+1}$, that is,

$$\Delta\mathbf{c}_{i,k} = \alpha_{i,k}\left(\mathbf{x}_{k+1} - \mathbf{c}_{i,k}\right)$$

In a first step, one can limit this quantity to the minimum distance between the input pattern and the set of all centres, i.e.,

$$\left\|\Delta\mathbf{c}_{i,k}\right\| \leq \min_{1\leq j\leq m}\left\{d_{j,k}\right\}, \quad \forall i$$

where $d_{i,k} = \left\|\mathbf{x}_{k+1} - \mathbf{c}_{i,k}\right\|$.

By rationale, the closer the centre is to the new pattern, the higher should be the adaptation step. One way to achieve this is to define

$$\alpha_{i,k} = \frac{d_{i,k}^{-1}}{\sum_{j=1}^{m} d_{j,k}^{-1}}; \quad d_{i,k} > 0, \ \forall i$$

In an adaptation framework, the learning rates may be allowed to stabilise with time, that is, slowly decrease to zero. Thus, using the learning rate defined by Chen *et al*. [16], we can modify the previous expression to the following:

$$\alpha_{i,k} = u_{i,k}\beta_{i,k}, \quad \forall i$$

where

$$\begin{cases} u_{i,k} = \dfrac{d_{i,k}^{-1}}{\sum_{j=1}^{m} d_{j,k}^{-1}}, \quad d_{i,k} > 0 \\ \beta_{i,k} = \left(1 + \left\lfloor \dfrac{\#Q_{i,k}}{m} \right\rfloor\right)^{-1/2} \beta_{i,k-1} \\ Q_{i,k} = \left\{ \mathbf{x}_{j+1} : d_{\min,j} = \min\limits_{1 \le i \le m} \left\{ d_{i,j} \right\}; \ j = 0,1,...,k \right\} \end{cases}$$

$\#Q_{i,k}$ is the number of times that the centre $\mathbf{c}_{i,k}$ was chosen as being the closest to the new input pattern until the present iteration of the process. This procedure enables that a centre selected fewer times has a larger adaptation step so that it is positioned to a better location and hence to be chosen more times in the future.

Another possibility is to adapt all centres in a way inspired by particle swarm optimisation [24], drawing from the analogy between RBF centres and the particles in a 'swarm'. The advantage here is that the widths can also be considered in the adaptation process. Nevertheless, further research in this direction is required as potential advantages of this procedure come out.

CHAPTER 3

Hybrid Dynamic Models for Advanced Time Series Forecasting

Contents

Abstract

In this Chapter we aim to describe the application of supervised neural networks to time series prediction. We identify some problems raised by heavily nonstationary time, and some drawbacks of the classical detrending and differencing preprocessing approaches. As an alternative approach, a hybrid dynamic model, including a stochastic trend and a neural submodel, is proposed, with methodological guidelines for its identification.

3.1 Dealing with Nonstationarity

3.1.1 Outward Generalisation

A univariate time series with *n* observations is usually indexed between 1 and *n*, or, if those observations are irregularly spaced, between time stamps t_1 and t_n. When we compute an estimate for the series, the corresponding time stamp of that estimate can belong to the interior or the exterior of the interval $[t_1, t_n]$, and then we usually classify it as problem of interpolation or of extrapolation.

By analogy, for multivariate, input-output models, with *p* independent (or explanatory or casual) variables, all sampled in the same interval $[t_1, t_n]$, we may say that we have a problem of interpolation if the prediction is to be based on a point in the interior of $[t_1, t_n]^p$.

In general, we regard the interpolation problem as less risky than extrapolation, even when the interpolation estimate is further away from its chronologically closest observation than the extrapolation estimate is.

The above discussion, although elementary, has a great relevance in the context of pattern recognition and supervised learning for the purpose of *generalisation*. The set of input patterns to be used in the learning process, **X**, interpreted as a collection of points in a multidimensional space, has a convex linear hull, *H*. We may then distinguish two types of generalisation problems, interpolation or extrapolation, depending whether the new input pattern, **x**, for which an output pattern must be estimated, is located in the interior or the exterior of *H*.

If we build the training patterns from a stationary time series, it is common that a new input pattern, **x**, used for prediction (extrapolation in the time series sense) belongs to the interior of *H* (interpolation in the sense of pattern recognition).

However, if the original time series is nonstationary, the series of learning patterns will itself be nonstationary and the new input pattern **x** will be more often located outside *H* – we will describe this problem as one of *outward generalisation*. Naturally, this is regarded as more prone to generate poor results than *inward generalisation*. In general, the closer **x** is from **X** – say, in terms of the average distance between **x** and the points in **X** – the smaller is the expected generalisation error.

Many real time series are far from being stationary, presenting clear trend or periodic patterns, among other possible effects, such as sudden changes in the mean, or in the variance. Nonstationarity processes are more difficult to analyse and due to the variety of possible effects, there are no models or tools universally adequate.

From a practical point of view and mainly for inference purposes, several statistical forecasting procedures are based on the assumption that the approximate stationarity can be achieved through appropriate preprocessing methods. Therefore, it is very convenient to find means of adequately preprocessing the data so to render the training patterns moderately stationary, and thus reducing the likelihood of outward generalisation occurring [66].

3.1.2 Usual Preprocessing Techniques

There are no proven 'automatic' techniques to identify trend components, but common approaches in practice for achieving stationarity in the mean of a time series include differencing and deterministic detrending. Either approach has drawbacks.

Differencing is widely used to render a time series stationary in the mean; by repeating the procedure, thus creating second-order differences, one often achieves stationarity in the variance as well. Seasonal differencing is also common, especially in the context of the Box-Jenkins methodology [7]. First-differences are computed as $y_k - y_{k-1}$. If a time series becomes stationary by differencing it (once or more times) then the series is said to be *difference-stationary* (DS). A special case is the random walk model:

$$y_k = y_{k-1} + \varepsilon_k \tag{3.1}$$

where ε_k is assumed to be a white noise process.

Data-detrending is commonly understood as the removal of a deterministic growth curve, representing most of the low-frequency effects in the series. It is often easy to accept that the general growth pattern of a series is approximately linear, quadratic, logistic, etc., and, as an alternative to transforming or differencing the data, it may be appealing to fit a curve of one of these families, leaving a residual series usually easier to model. In the simplest case, detrending is accomplished through the linear fitting model

$$y_k = a + bk + \varepsilon_k$$

where ε_k is also assumed to be a white noise process. Such a series is said to be *trend-stationary* (TS).

First-order differencing and linear regression detrending, as well as higher-order extensions of these, are likely to produce different types of distortion in the spectral characteristics of the resulting data (see [14, 35]). Differencing typically exaggerates the concentration of power in the higher frequencies portion of those spectra. Such distortions can be noticed even in the residuals obtained from differencing a series consisting of a straight line with added white noise. Indeed, in that case, the differenced series is not white noise, as one might expect. On the other hand, the removal of a deterministic curve usually does not eliminate all the low-frequency power and typically leaves a spurious low-frequency peak in the spectrum of the residual series [51].

To illustrate the drawbacks of both approaches, let us first consider a simulated trend-stationary series, with 300 data points, using the model:

$$y_k = 5 + 0.05k + \varepsilon_k, \quad \varepsilon_k \sim N(0,1) \tag{3.2}$$

and then differentiate it. Figure 3.1 shows the trend-stationary time series, the resulting first-differenced series and the periodogram (or the sample spectrum) of the latter. In this case, differencing was inappropriate, because although it has removed the trend component, it has also introduced a complexity structure into the errors that is visible in the higher frequencies portion of the sample spectrum.

Conversely, let us now consider a simulated difference-stationary series using expression (3.1), with the same Gaussian white noise process as in (3.2), and detrend it. As it can be shown in Figure 3.2, the trend component was not completely removed and, additionally, spurious low-frequency peaks now seem to appear in the spectrum of the residuals.

We defend that the above techniques should be replaced by more elaborate alternative ones, where, for instance, stationary sequences of patterns may be created from nonstationary time series, a little different from the traditional differencing procedure. As we have discussed, the main problem lies in that, without any preprocessing, there would be no clear clusters of input patterns, or such clusters would be correlated with specific time intervals.

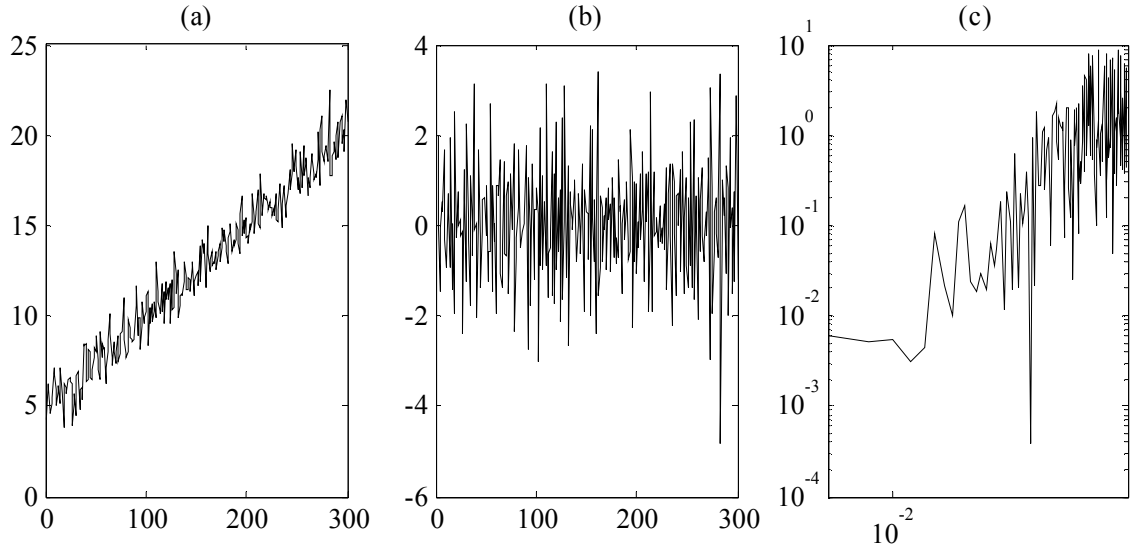

Figure 3.1: (a) A simulated trend-stationary series; (b) time series after first-order differencing; (c) the periodogram of the differenced series b) (in log-log scales).

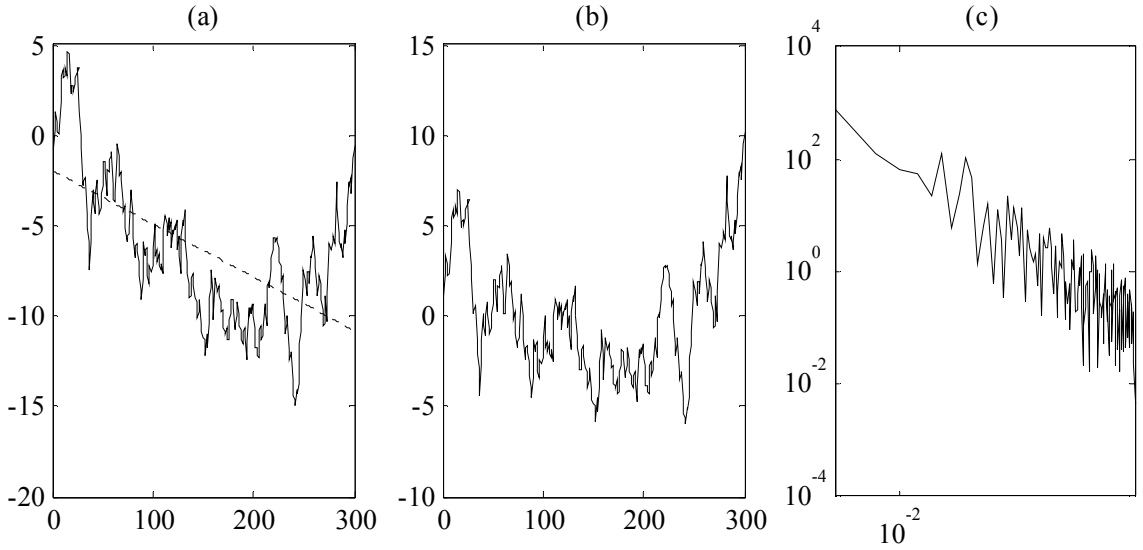

Figure 3.2: (a) A simulated difference-stationary series (solid line) and the regressive trend line (dotted line); (b) the detrended; (c) the periodogram detrended series (in log-log scales).

The alternative preprocessing approaches that we exemplify next include pattern-differencing and pattern-standardising, as firstly described Silva [66].

3.1.3 Alternative Preprocessing Methods

If the time series has a locally linear trend, but is stationary in the variance, the pattern-differencing approach consists of differencing the multivariate series made of the input patterns, instead of differencing the original time series.

In the *pattern-differencing* approach, each input pattern, say

$$\mathbf{x}_k = \begin{bmatrix} y_{k-p} & y_{k-p+1} & \cdots & y_{k-1} \end{bmatrix}^T$$

is shifted, originating a new input pattern:

$$\mathbf{x}'_k = \begin{bmatrix} y_{k-p+1} - y_{k-p} & \cdots & y_{k-1} - y_{k-p} \end{bmatrix}^T$$

The dimension of the input space is reduced by one. The corresponding output pattern, say y_k, is shifted accordingly:

$$y'_k = y_k - y_{k-p}$$

and the new input-output training patterns is thus $(\mathbf{x}'_k, y'_k)$, for $k = p+1,...,n$.

Another approach consists of the input patterns rescaling through a linear transformation, for instance, standardising each input pattern such that it has zero mean and variance one ('*pattern-standardising*'):

$$\mathbf{x}'_k = \begin{bmatrix} \dfrac{y_{k-p} - \overline{\mathbf{x}}_k}{\sigma_k} & \dfrac{y_{k-p+1} - \overline{\mathbf{x}}_k}{\sigma_k} & \cdots & \dfrac{y_{k-1} - \overline{\mathbf{x}}_k}{\sigma_k} \end{bmatrix}^T,$$

where $\overline{\mathbf{x}}_k$ and $\sigma_{\mathbf{x}_k}$ are, respectively, the mean and the sample standard deviation of the elements in $\mathbf{x}_k$.

The corresponding output pattern is shifted and rescaled accordingly:

$$y'_k = \frac{y_k - \overline{\mathbf{x}}_k}{\sigma_k}$$

For the sake of illustration, let us consider the difference-stationary series given in Figure 3.2. Assuming $p = 12$, we show in Figure 3.3 the first 30 preprocessed input patterns, for each of the proposed approaches.

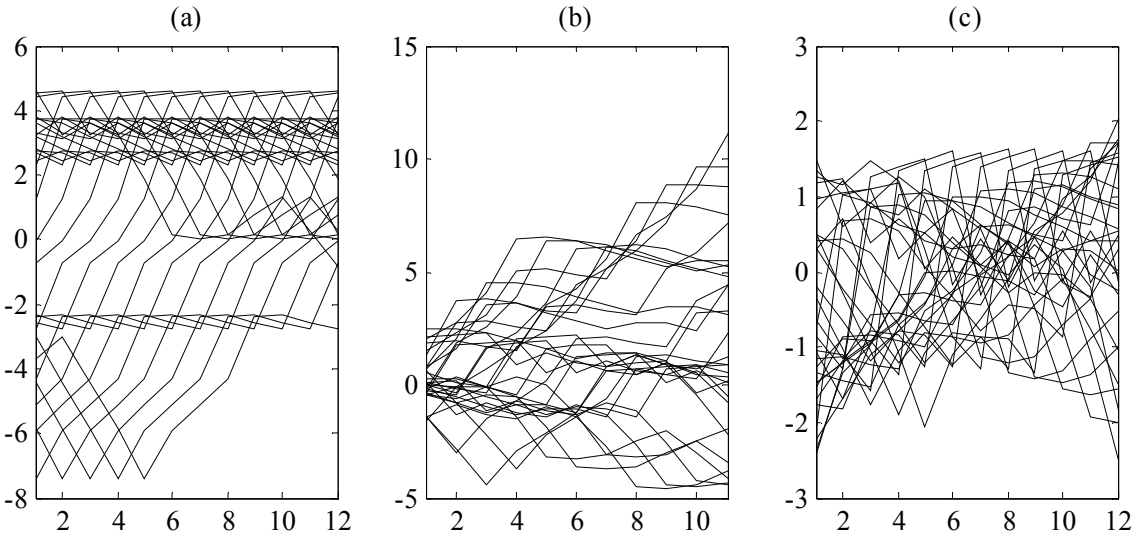

Figure 3.3: (a) Original input vectors of observations; (b) input patterns after differencing; (c) input patterns after standardising.

The main criticism against first-differencing a time series is that the procedure amplifies peaks in the spectrum located at higher frequencies. However, by differencing or rescaling (standardising) the input patterns, those high-frequency effects can be preserved provided the order of the autoregression is large enough.

3.2 Stochastic Detrending

As it was mentioned before, the traditional preprocessing techniques are inadequate, as they tend to significantly distort the spectral characteristics of the time series. However, we can expect that such distortion is negligible if an adequate trend model – possibly, IAR(1) plus noise – is identified and estimated, and removed from the series. This requires that the power in the left-hand side of the spectrum, that is imputable to the trend, does not overlap other spectral peaks.

Stochastic detrending can be described as a kind of prefiltering (or preliminary filtering), meant to facilitate the analysis and modelling of the high-frequency effects in the data. It consists of the preliminary estimation and removal of a dynamic stochastic model, to account for the trend or other low-frequency effects [64]. Two popular methods aimed at removing low-frequency variations from a time series are the Hodrick-Prescott and the Baxter-King filters, but they have been the subject of several criticisms [31, 57].

We thus propose a methodology in two phases: in the first, we make use of a dynamic stochastic model estimated through the Kalman filter to deal with low-frequency effects and, in the second phase, a supervised neural network is used to account for the high-frequency, and possibly nonlinear effects. A similar approach, Sequential Spectral Decomposition, was presented by Ng and Young [53], consisting of the decomposition of the original series in quasi-orthogonal components. Usually, dynamic trend (or periodic) and AR component models are used. The latter is a purely linear model, whereas a nonlinear one is considered here.

For this purpose, we may start by considering the simple, yet versatile, DTR model with IAR(1) parameters, as introduced in Section 2.4:

$$\begin{cases} y_k = t_k + \varepsilon_k, \quad \varepsilon_k \ \text{iid} \ \left(0, \sigma_\varepsilon^2\right) \\ t_k = t_{k-1} + s_{k-1} \\ s_k = \alpha\, s_{k-1} + \eta_k, \quad \eta_k \ \text{iid} \ \left(0, \sigma_\eta^2\right) \end{cases}$$

Its main aim is to describe trend components. A series detrended by using this model might not yet be stationary, either in the mean – for instance, due to the presence of other low-frequency peaks in the spectrum – or in the variance. In the latter case, the application of a nonlinear transformation to the data, which is commonly advocated, could not be able to

render the series stationary in the variance, and might create by itself important distortions in the spectral characteristics of the data.

The obvious way out of these problems is to identify and estimate not only a trend model, but a more complex model for the data, for instance using the Dynamic Harmonic Regressive (DHR) model [78] – which is composed by a trend component, one or more periodic components and an error component:

$$y_k = t_k + \sum_{i=1}^{m} \left(a_{ik} \cos\left(2\pi f_i k\right) + b_{ik} \sin\left(2\pi f_i k\right) \right) + \varepsilon_k$$

where f_i $(i = 1,...,m)$ are the frequencies of interest. We would allow the residuals of the DHR model to still carry colour or structure that would hopefully be handled by the neural network.

To facilitate the identification of a DHR model, we may consider a IAR(1) process for the trend component and RW processes for the parameters of the periodic components, a_{ik} and b_{ik}. The components of the DHR model, with mixed IAR(1) and RW processes, can be written as:

$$\begin{cases} t_k = t_{k-1} + s_{k-1} \\ s_k = \alpha s_{k-1} + \eta_k, \quad \eta_k \ \text{iid} \ \left(0, \sigma_\eta^2\right) \\ a_{ik} = a_{i,k-1} + \xi_k^{(a)} \\ b_{ik} = b_{i,k-1} + \xi_k^{(b)}, \quad \xi_k^{(a)}, \xi_k^{(b)} \ \text{iid} \ \left(0, \sigma_\xi^2\right) \end{cases}$$

with all noise input sequences assumed to be uncorrelated. The risk of producing distortions in the residuals spectrum is then much attenuated or negligible provided suitable values for the hyperparameters are used in those models, that is, α, $NVR_\eta = \sigma_\eta^2 / \sigma_\varepsilon^2$ and $NVR_\xi = \sigma_\xi^2 / \sigma_\varepsilon^2$. Once we have identified a suitable model for the low-frequency effects in the data, we can identify a supervised neural network to account for higher frequencies and nonlinear autocorrelations.

The identification of the hyperparameters in the first phase should be carried out with caution. Although more elaborated methods would be advisable in most cases, such as optimisation methods in the frequency domain [78], we propose a simple procedure for the identification of the preliminary dynamic model. Since the aim of the preliminary stage is

the identification of a long-run term, the optimisation process should be based on forecast errors for large forecasting horizons rather than smaller ones. Too low time horizons will likely yield irregular fluctuations of the trend component. This can be illustrated using the difference-stationary series given in Figure 3.2.

Let us consider the estimation of two DTR-IAR(1) models through the Kalman filter, optimised based on *RMSE*(1) and *RMSE*(12), e.g., based on 1-step-ahead and 12-steps-ahead forecast errors. The first 35% observations were ignored in the computation of those performance measures. The best pairs of values $\left(\alpha^*, NVR_\eta^*\right)$ are, approximately and respectively, $\left(0.85, 0.1\right)$ and $\left(0.95, 10^{-5}\right)$. The resulting predictive estimates, for each of horizons 1 and 12, are shown in Figure 3.4. The high value obtained for the noise variance ratio of the former model causes the predictions to have high variability, following narrowly the noise in the data.

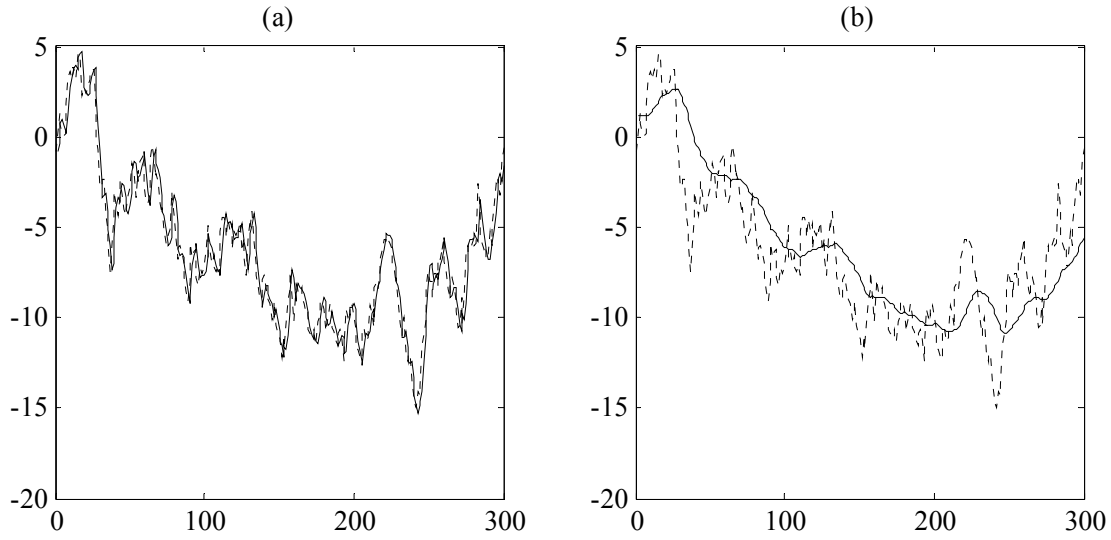

Figure 3.4: Predictive estimates of the difference-stationary series obtained from a DTR-IAR(1) model (solid line), optimised based on: (a) 1-step-ahead forecast errors; (b) 12-steps-ahead forecast errors.

In general, the predicted trend component of the difference-stationary series is expected to be smoother when one considers longer time horizons in the optimisation of the DTR-IAR(1) model. In Figure 3.5 we can observe that stochastic detrending is more effective that deterministic detrending in accounting for the very low-frequency effects in the series.

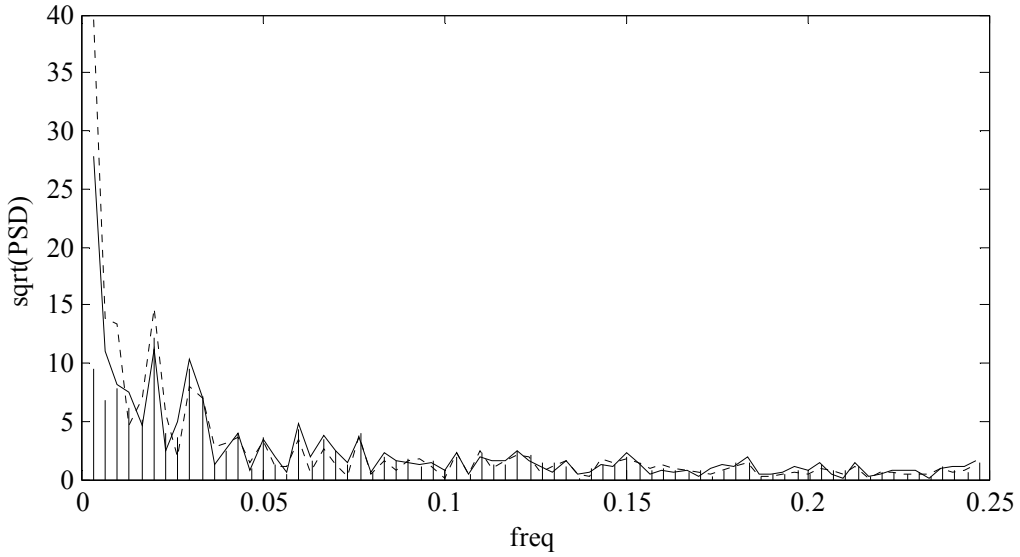

Figure 3.5: Comparing the periodograms of the difference-stationary series (dotted line), and the difference-stationary series after removing the trend component, either by deterministic detrending (solid line) or stochastic detrending (stems).

In case of doubt of which is the best classical preprocessing technique that should be employed, we find that the stochastic detrending procedure gives reasonable results, even if one considers default hyperparameter values, say $\alpha = 1$ and $NVR_\eta = 10^{-3}$.

After a successful stochastic detrending phase, a neural network should then be employed to take into account higher frequencies left in the detrended series. The hyperparameters, namely the number of inputs, *p*, the number of RBF units, *m*, the centres, $\{\mathbf{c}_i\}$, and the widths, $\{\sigma_i\}$, should then be identified based on forecast errors of the detrended time series. The final prediction is obtained by adding the predictive estimate given by a dynamic regressive model and the one obtained by a neural network.

The algorithm for the stochastic detrending methodology through sequential estimation can be summarised in Algorithm 3.1, where in the first phase we make use of a DTR model, with first-order integrated autoregressive parameters, estimated by the Kalman filter and, in the second phase, a RBFN is used, with constant parameters, estimated by the basic RLS algorithm. We denote this methodology as DTR;RBF.

PART 1:

Define values for hyperparameters: $\left(\alpha, NVR_\eta\right)$

Define transition and NVR matrices: $\mathbf{F} = \begin{bmatrix} 1 & 1 \\ 0 & \alpha \end{bmatrix}$ and $\mathbf{Q}_r = \begin{bmatrix} 0 & 0 \\ 0 & NVR_\eta \end{bmatrix}$

Define initial values, e.g. $\hat{\boldsymbol{\theta}}_0 = \mathbf{0}$ and $\mathbf{P}_0 = 10^4 \mathbf{I}_2$

Repeat, for $k = 1...N$:

Define input vector: $\mathbf{u}_k = \begin{bmatrix} 1 & 0 \end{bmatrix}^T$

Update estimates using Kalman filter: $\hat{\boldsymbol{\theta}}_{k|k}$ and $\mathbf{P}_{k|k}$

Compute prediction: $\hat{t}_{k+h|k} = \mathbf{u}_{k+h}^T \mathbf{F}^h \hat{\boldsymbol{\theta}}_{k|k}$

Define detrended time series: $z_k = y_k - \hat{t}_{k|k-h}$

Define RBF architecture: $\left(p, m\right)$

Define input/output patterns: $\left(z_k, \mathbf{x}_k := \begin{bmatrix} z_{k-h} & \cdots & z_{k-h-p+1} \end{bmatrix}^T\right)$

Identify centres and widths from patterns: $\left\{\mathbf{c}_i\right\}$ and $\left\{\sigma_i\right\}$

PART 2:

Define initial values, e.g. $\hat{\boldsymbol{\theta}}_0 = \mathbf{0}$ and $\mathbf{P}_0 = 10^4 \mathbf{I}_m$

Repeat, for $k = p + h...N$:

Define input vector: $\mathbf{u}_k = \begin{bmatrix} \phi_{1k} & \cdots & \phi_{mk} \end{bmatrix}^T$

Update estimates using RLS algorithm: $\hat{\boldsymbol{\theta}}_k$ and $\mathbf{P}_k$

Compute prediction: $\hat{z}_{k+h|k} = \mathbf{u}_{k+h}^T \hat{\boldsymbol{\theta}}_k$

Compute final prediction: $\hat{y}_{k+h|k} := \hat{t}_{k+h|k} + \hat{z}_{k+h|k}$

Algorithm 3.1: Stochastic detrending methodology (DTR;RBF).

3.3 Simultaneous Estimation of Hybrid Models

Once we have separately identified suitable models for the low-frequency and for the high-frequency effects in the data, these must be put together in the form of a complete model, to be estimated simultaneously, and tested. This will produce estimates that, in some cases, can be significantly different from those obtained by sequential estimation of the components. In other words, the estimates of a sum of components are not equal to the sum of the estimates of the individual components. The difference is smaller when there is little spectral overlapping between the components (quasi-orthogonality).

In the simultaneous approach, the identification might occur in a very high-dimensional space, much higher than in the sequential one. In the most complete form, where we consider a DTR model to take into account the long-run term or other low-frequency effects, and a Gaussian RBFN for higher-frequency effects, with parameters described by RW processes, we will have a 6-tuple $\left(\alpha, NVR_\eta, p, m, \sigma, NVR_\xi\right)$ to be identified, with NVR_ξ being the common noise variance ratio value associated to all the parameters of the latter model.

To facilitate the identification task, we will allow that some of the settings are inherited from the sequential estimation, namely the architecture of the neural network, *p* and *m*, the location of the centres, and the widths. Other hyperparameters will then be allowed to be optimised again, specifically the hyperparameters associated to the DTR model, α and NVR_η. In addition, the value for NVR_ξ should also be optimised. This gives extra flexibility to the simultaneous estimation, allowing that some readjustments can be done afterwards.

The hybrid model proposed here is actually a special case of the univariate Unobserved Components model and formulated as

$$y_k = t_k + c_k + s_k + f\left(\mathbf{u}_k\right) + \varepsilon_k, \quad e_k \sim N\left(0, \sigma_\varepsilon^2\right).$$

In our approach the trend component, t_k, the cyclical (or quasi-cyclical) component, c_k, and the seasonal component, s_k, are accounted by for the DHR model, whereas $f\left(\mathbf{u}_k\right)$ corresponds in our case to the RBFN. The simultaneous estimation procedure, in a state-space framework, is summarised in Algorithm 3.2.

PART 1:

The same as PART 1 in Algorithm 3.1.

PART 2:

Define $\mathbf{F} = \left[\begin{array}{cc|ccc} 1 & 1 & 0 & \cdots & 0 \\ 0 & \alpha & 0 & \cdots & 0 \\ \hline 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{array}\right]$ and $\mathbf{Q}_r = \left[\begin{array}{cc|ccc} 0 & 0 & 0 & \cdots & 0 \\ 0 & NVR_\eta & 0 & \cdots & 0 \\ \hline 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{array}\right]$

Define initial values, e.g. $\hat{\boldsymbol{\theta}}_0 = \mathbf{0}$ and $\mathbf{P}_0 = 10^4 \mathbf{I}_{m+2}$

Repeat, for $k = p + h...N$:

Define input vector: $\mathbf{u}_k = \left[\begin{array}{cc|ccc} 1 & 0 & \phi_{1k} & \cdots & \phi_{mk} \end{array}\right]^T$

Update estimates using Kalman filter: $\hat{\boldsymbol{\theta}}_{k|k}$ and $\mathbf{P}_{k|k}$

Compute final prediction: $\hat{y}_{k+h|k} := \mathbf{u}_{k+h}^T \mathbf{F}^h \hat{\boldsymbol{\theta}}_{k|k}$

Algorithm 3.2: Simultaneous estimation of hybrid models (DTR-RBF).

A simple illustration can be provided to demonstrate a possible behaviour produced by the simultaneous approach and then to contrast it with the one given by the sequential methodology. To do this, let us consider a simulated nonstationary time series generated by two distinct simple models:

$$\begin{cases} y_k = \cos\left(\dfrac{\pi}{200} k\right) + x_k \\ x_k = 0.5 x_{k-1} + 1.2\left(x_{k-1} - x_{k-2}\right) e^{-x_{k-1}^2} + \varepsilon_k, \quad \varepsilon_k \sim N\left(0, 0.1\right) \end{cases}$$

where $x_1 = \varepsilon_1$ and $x_2 = 0.5 x_1 + \varepsilon_2$. The first component (the trigonometric model with a very low-frequency component) aims to induce nonstationarity, and the other component (a nonlinear autoregressive model of order 2 – NAR(2)) aims to produce high-frequency effects.

Figure 3.6 shows a sample of the simulated time series. From the corresponding periodogram, one can notice the presence of a trend, as well as high-frequency effects spread along a range of frequencies, due to the nonlinearities in the generating model.

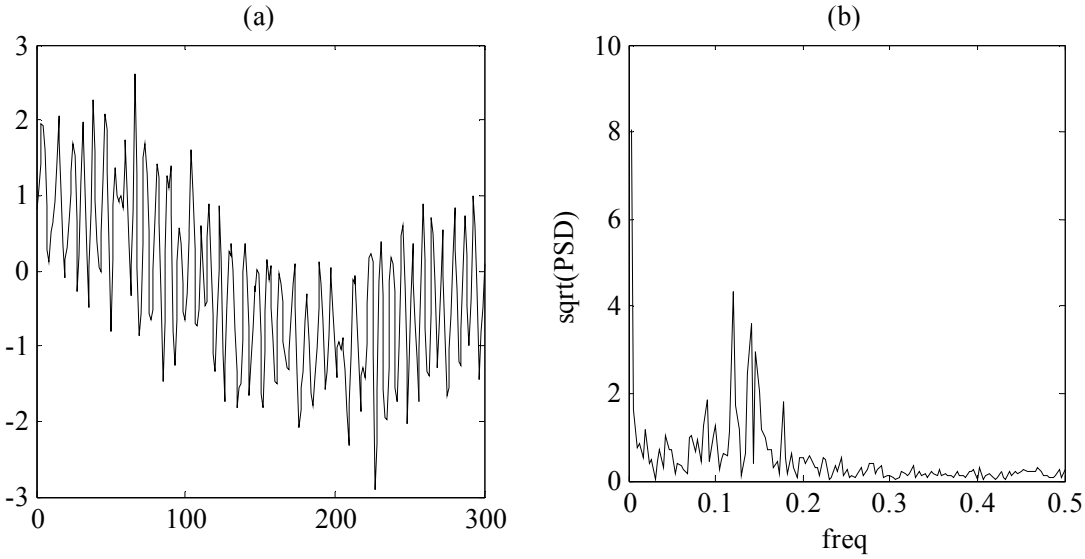

Figure 3.6: (a) A sample of a simulated nonstationary time series; (b) the corresponding periodogram.

Let us start by considering the first phase of the sequential methodology by setting a DTR-IAR(1) model with $\alpha = 1$ and $NVR_\eta = 10^{-3}$. After the dynamic model has been estimated, we have proceeded to the identification of a Gaussian RBFN, with constant parameters, to predict the resulting time series, $z_k = y_k - \hat{t}_{k|k-1}$, where $\hat{t}_{k|k-1}$ are the 1-step-ahead forecast estimates obtained by the former model. The 3-tuple (p, m, σ) was optimised and the best combination was found to be $(2, 5, 1.72)$, for which $RMSE(1) = 0.344$.

Adopting the settings found in the stochastic detrending phase, namely the values for the RBFN, but now reoptimising the values of α and NVR_η, we have estimated the composed model DTR-RBF. With the optimised pair $(\alpha^*, NVR_\eta^*) = (0, 0.1)$, we obtained $RMSE(1) = 0.327$. Figure 3.7 shows the 1-step-ahead predictive estimates obtained from each methodology. Although very similar, they differ in the way they were obtained.

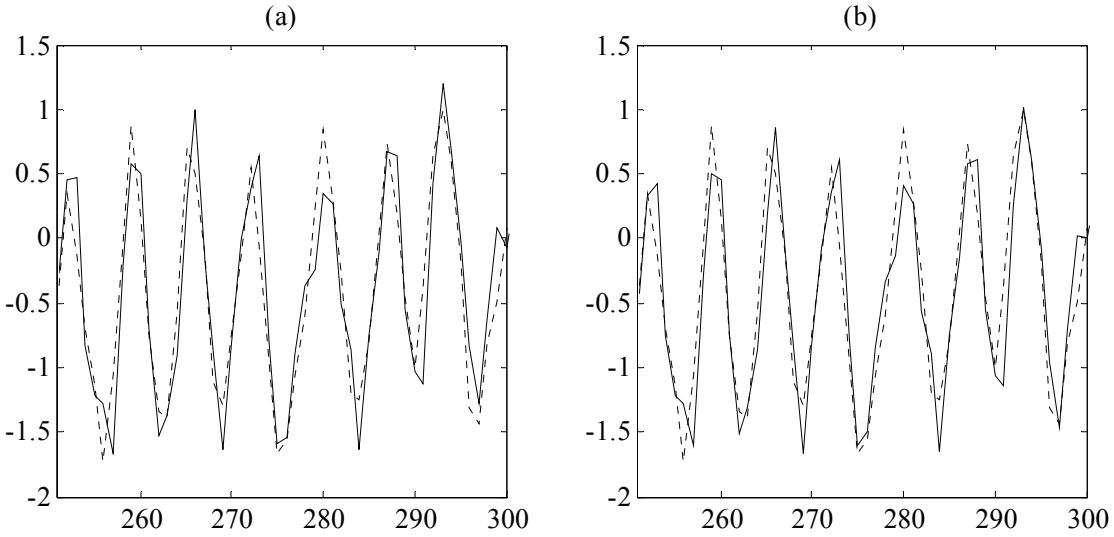

Figure 3.7: Last 50 of 1-step-ahead predictive estimates (solid line) of a NAR simulated series (dotted line) obtained either by (a) the sequential estimation or (b) the simultaneous estimation methodologies.

3.4 Experimental Results

For this experiment, and for the experiments described in Section 4.5, we have simulated 9 time series, through the following Gaussian RBFN, corrupted with additive white noise:

$$\begin{cases} x_k = \sum_{i=1}^{5} \theta_i \exp\left(-0.5 \left\| \mathbf{i}_k - \mathbf{c}_i \right\|^2 / \sigma_i^2\right) + \varepsilon_k \\ \mathbf{i}_k = \left(x_{k-2}, x_{k-1}\right)^{\mathrm{T}} \\ \mathbf{c}_i \sim \mathbf{U}\left(-a, a\right)^2, \quad \sigma_i = \sigma, \ \forall i \\ \varepsilon_k \sim N\left(0, 0.1\right) \end{cases}$$

The parameters are assumed to be constant in time and were chosen randomly. To generate the different sequences of data – named A,B,…,H,I – we have assigned different combinations of values for the pair $\left(a, \log_{10}\sigma\right)$, as defined in Table 3.1, as well as different locations for the clusters.

	A	B	C	D	E	F	G	H	I
a	1	2	3	1	2	3	1	2	3
$\log_{10}\sigma$	-0.25	0	0.25	0	0.25	-0.25	0.25	-0.25	0

Table 3.1: Simulation settings for 9 time series.

In Figure 3.8 we show the simulated time series A (the first 100 out of 300 data points) and its periodogram. One can notice a clear periodic effect, spread along a range of frequencies, due to the model nonlinearity. However, the difficulty in predicting time series like this one is due much more to the presence of noise in the observations than to the nonlinear autocorrelations.

Gaussian RBFNs were then also considered to predict the simulated data, but ignoring the specific settings used in the simulation. For each time series, a pseudo-best model, M0, estimated through the RLS algorithm, and with centres based on *k*-means clustering, was found with respect to the *RMSE* measure based on in-sample 1-step-ahead and 4-steps-ahead prediction errors (as usual, the first 35% observations were ignored in the calculation of the performance measure to avoid the initialisation bias), respectively, where the 3-tuple $\left(p, m, b\right)$ was optimised:

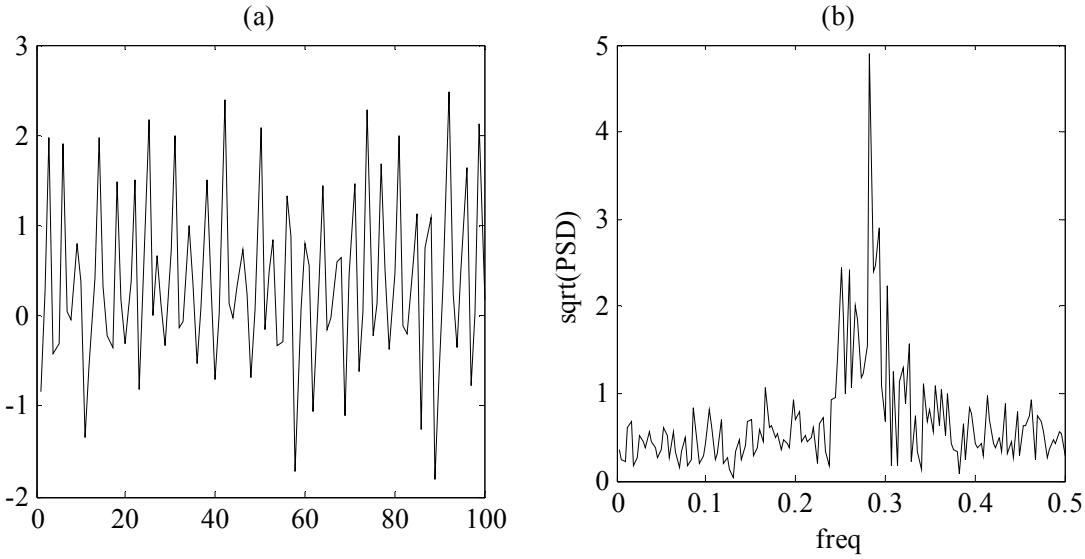

Figure 3.8: (a) First 100 values of time series A; (b) periodogram of A.

— p: number of inputs;

— m: number of RBF units;

— $b = \log_{10} \sigma$ (all units with equal widths, optimised in a logarithmic scale).

The following sets of possible combinations of values were considered for both time horizons:

$$p \in \{1, 2, ..., 8\}; \quad m \in \{2, 3, ..., 14\}; \quad b \in \{-1, -0.75, ..., 1\}$$

In Table 3.2 we show the best values obtained for each time series.

We have then considered 18 simulated nonstationary time series, built from the previous 9 time series:

— TS (*trend-stationary*): $y_k = 5 + 0.05k + x_k$

(A, plus a straight line model; similarly for B to I);

— DS (*difference-stationary*): $y_k = y_{k-1} + x_k$

(integration of A, as shown in Figure 3.9; similarly for B to I).

	A	B	C	D	E	F	G	H	I
p	2	2	2	2	3	2	2	2	2
m	10	4	10	12	10	6	6	12	8
b	-0.25	0.25	0.25	0	0.5	-0.25	0	0	0.25
RMSE(1)	0.32	0.30	0.33	0.32	0.34	0.30	0.31	0.34	0.32
p	2	2	2	4	2	2	2	4	2
m	14	11	13	12	4	3	3	7	13
b	-0.25	0	0.25	0.25	0.25	-0.25	0.25	0	-0.25
RMSE(4)	0.66	0.47	0.47	0.58	0.50	0.56	0.42	0.54	0.79

Table 3.2: Best values attained by pseudo-best model M0 for 9 simulated time series.

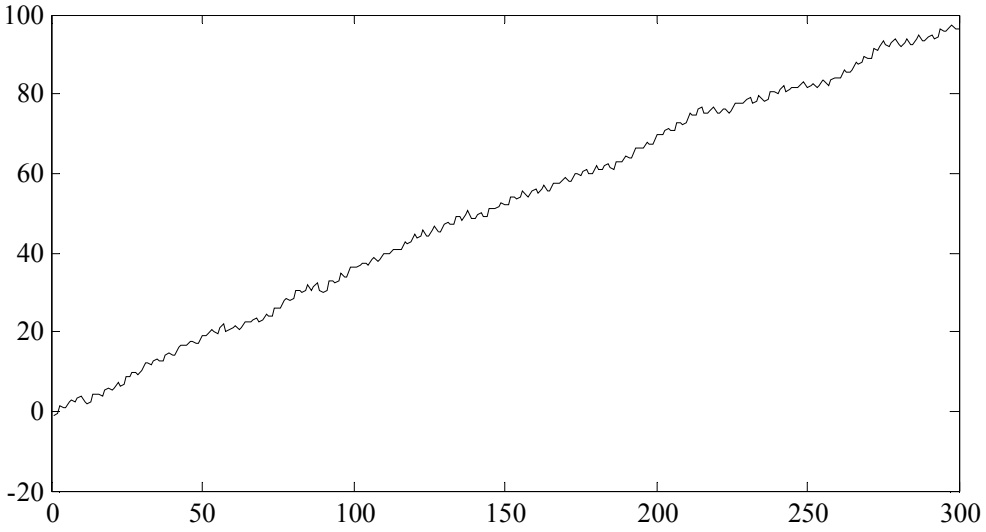

Figure 3.9: Integration of time series A (DS).

First-order differencing would be suitable for DS but not for TS, while deterministic detrending would be suitable for TS but not for DS.

First, we compared the performance of the DTR;RBF model with the classic and the alternative preprocessing approaches used in forecasting nonstationary time series. Furthermore, we aimed to assess, for both time series types – TS and DS –, the predictive performance of a model composed of a dynamic trend (DTR-IAR(1)) and Gaussian radial basis function network, with constant parameters (GRBFN), as a general purpose approach.

For completeness, we have compared the following models and approaches for forecasting both variants, TS and DS, for each of the time series, A to I:

1. GRBFN (constant parameter model);
2. GRBFN-RW (random walk parameters);
3. DTR-IAR(1);
4. tr;RBF (GRBFN modelling, after deterministic trend removal);
5. fd;RBF (GRBFN modelling, after first-order differencing);
6. pd;RBF (GRBFN modelling after pattern-differencing);
7. ps;RBF (GRBFN modelling after pattern-standardising);
8. DTR;RBF (GRBFN modelling, after a preliminary DTR-IAR(1) filtering);
9. DTR-RBF (simultaneous estimation of a model composed by the two submodels).

The first two (neural) models were directly applied to the time series, without any preprocessing.

Additionally, to assess the comparative performance of these approaches with a real-life application, we have also considered a well-known time series, here denoted LGNP: the logarithm of quarterly U.S. real gross national product, 1947Q1-2003Q3 (see Appendix A.1). This data set, or its annual version, has been the most studied to support the discussion of the trend-stationarity and difference-stationarity approaches to macroeconomic time series identification [22].

We have optimised the 3-tuple (p, m, σ) in all RBFNs under study, with the exception of the second approach, where there is a NVR to be identified. In the approaches DTR;RBF and DTR-RBF, we have stochastically detrended the data using the default values $\alpha = 1$ and $NVR = 10^{-3}$. Then, in the simultaneous estimation approach, we have optimised also these hyperparameters. The main results obtained are summarised in Table 3.3.

Namely, since the σ_ε value of the true model can be seen as a lower bound for the $RMSE(h)$ predictive performance (for both $h = 1$ and $h = 4$), we report the average, over the 9 time series, of the index

$$J(h) = 100 \times \left(RMSE(h) - \sigma_\varepsilon \right) / \sigma_\varepsilon$$

		Simulated data (A-I)				*Real data (LGNP)*	
Model	*Method*	*TS:* $\bar{J}(1)$	*DS:* $\bar{J}(1)$	*TS:* $\bar{J}(4)$	*DS:* $\bar{J}(4)$	*100* RMSE*(1)	*100* RMSE*(4)
GRBFN	RLS	99.2	9773.1	115.7	17503.4	1.21	2.56
GRBFN-RW	RLS-CA	84.3	991.4	115.3	1859.1	1.11	2.56
DTR-IAR(1)	KF	716.7	14118.3	745.0	14724.8	28.97	30.70
tr;RBF	RLS	**2.6**	58.4	**74.1**	158.7	0.83	**2.23**
fd;RBF	RLS	38.8	**1.6**	204.3	182.8	0.84	2.37
pd;RBF	RLS	34.4	5.0	88.7	**121.8**	0.84	2.32
ps;RBF	RLS	45.3	35.5	94.9	157.9	0.86	2.59
DTR;RBF	KF+RLS	24.3	33.5	90.8	152.6	**0.58**	2.68
DTR-RBF	KF	16.7	33.1	91.9	177.4	1.02	2.92

Table 3.3: Comparative predictive performance, 1 and 4-steps-ahead, of 9 different modelling approaches, for different types of nonstationarity (TS, DS, or unknown).

Therefore, the best approaches for the identification of a forecasting model for TS-type and DS-type time series were, as expectable, (tr;RBF) and (fd;RBF), respectively.

Analysing the results summarised in this table, we can notice that the predictive performance of the DTR and RBFNs is not satisfactory when they are applied separately. Moreover, the DTR;RBF produced much better results when compared to the 'wrong' model of the series under study. Similar results were obtained from the alternative preprocessing methods.

With real-world time series, one does not know what the 'true' model is: probably neither a pure deterministic trend process nor a pure random walk process. From this experiment, the classic differencing and the deterministic detrending were proven to be inadequate in the wrong model of the given series. In doubt, and based on evidence from experiments similar to this one, we consider that the coupling of a dynamic regressive model with a Gaussian RBFN can be used as a general approach for dealing with nonstationary time series with nonlinear autocorrelations.

CHAPTER 4

Combining Predictions

Contents

Abstract

In this Chapter, we propose some new ideas for using the model mixing approach. The usual framework for linearly combining estimates from different models is extended to cope with the case where the forecast errors from different models are correlated. Recursive and adaptive expressions are also considered for the cases where the strength relations between those errors are assumed to vary with time, as well as a procedure to revise the combining weight values when new information becomes available.

4.1 Introduction

Probably, the most common approach for model combination is what we may call "model mixing", where one combines the estimates produced by the individual models through weights. The combination of estimates requires the definition of a metamodel, which should then be optimised with respect to a predefined criterion. The literature contains a large number of combining metamodels and methods, as well as their benefits in practical situations. Most of them are based on linear parametric metamodels, where the parameters are viewed as weights in the combination. They differ from each other essentially in how those weights are chosen.

The simplest approach is to assign constant weights to each model using a simple average, so that each forecast would be assigned the same weight (see, e.g., [48]), or to choose the weights that minimise the sum of squared forecast errors (as proposed by Granger and Ramanathan [30]). Two types of combining methods are usually mentioned: "variance-covariance" and "regression-based" methods. In the variance-covariance method the weights are restricted to sum to unity, where their values are determined from the forecast error variances and covariances. In the regression method the combining weights are estimated by simply regressing the underlying individual forecasts. The former method can be viewed as a particular case of the latter in the sense that its optimal variance-covariance combining weight vector has a regression interpretation as the coefficient vector of a linear projection of the observations onto the forecasts, subject to two constraints: the weights sum to unity and no intercept is included [21].

The advantages of methods based on linear regression come from their simplicity and their flexibility to be modified into versions that include recursive or adaptive formulae, in particular dynamic versions of the RLS algorithm, all of them with the objective to optimally estimate the combining weight vector.

4.2 Classic Linear Combination

The classic linear combination approach can be viewed as a weighted sum of the estimates produced by the individual models plus a constant term, where the weights are the parameters in the linear combination model. Bellow we review the main formulae associated with this approach.

Let $\left\{\hat{y}_{k|k-h}^{(i)}\right\}$ be *m* sequences of *h*-steps-ahead predictive estimates obtained from different models. The usual linear weighting combination is given by

$$\hat{y}_{k|k-h}^{(c)} = w_0 + w_1\hat{y}_{k|k-h}^{(1)} + \cdots + w_m\hat{y}_{k|k-h}^{(m)} \tag{4.1}$$

where the constant term, w_0, is a bias that allows the linear model function to be not restricted to pass through the origin. Ideally, but not necessarily, each sequence of forecast errors, given by

$$e_{k|k-h}^{(i)} = y_k - \hat{y}_{k|k-h}^{(i)} \quad \left(i = 1,...,m\right)$$

should have white noise properties and should be independent of each other. For simplicity, we will use e_k as an abbreviation of the forecast error $e_{k|k-h}$, with origin $k-h$ and for a specific time horizon *h* of interest.

Particular (constrained) cases of Equation (4.1) are sometimes considered, by setting $w_0 = 0$, or also by normalising the sum of the weights to one, $w_1 + \cdots + w_m = 1$. The unrestricted formulation has the advantage that the combined estimates are unbiased even if the individual ones are biased, while in restricted formulations this can only be assured if both estimates are unbiased [30].

The values for the weights are usually found through the minimisation of the sum of squared errors:

$$SSE\left(h\right) \equiv SSE = \sum_{k=1+h}^{n}\left(e_k^{(c)}\right)^2$$

We note that in recursive estimation it is advisable to ignore the first errors as they are usually biased due to the initialisation process.

In matrix notation, the general classic linear combination can be written as

$$\hat{\mathbf{y}}^{(c)} = \mathbf{F}\mathbf{w}$$

where $\hat{\mathbf{y}}^{(c)} = \begin{bmatrix} \hat{y}_{1+h}^{(c)} & \cdots & \hat{y}_n^{(c)} \end{bmatrix}^T$, $\mathbf{w} = \begin{bmatrix} w_0 & w_1 & \cdots & w_m \end{bmatrix}^T$, $\mathbf{F} = \begin{bmatrix} \mathbf{1} \mid \hat{\mathbf{y}}^{(1)} \mid \cdots \mid \hat{\mathbf{y}}^{(m)} \end{bmatrix}$ ($\mathbf{1}$ is a column of ones) and $\hat{\mathbf{y}}^{(i)} = \begin{bmatrix} \hat{y}_{1+h}^{(i)} & \cdots & \hat{y}_n^{(i)} \end{bmatrix}^T$. Therefore, the optimisation problem with respect to the least-squares criterion (as in (2.4)) can be defined by

$$\mathbf{w} : \min SSE = \mathbf{e}^{(c)^T} \mathbf{e}^{(c)}$$

where $\mathbf{e}^{(c)} = \begin{bmatrix} e_{1+h}^{(c)} & \cdots & e_n^{(c)} \end{bmatrix}^T$ is the vector of combined forecast errors, and

$$SSE = \mathbf{y}^T\mathbf{y} - 2\mathbf{w}^T\mathbf{b} + \mathbf{w}^T\mathbf{V}\mathbf{w}$$

with $\mathbf{V} = \mathbf{F}^T\mathbf{F}$ and $\mathbf{b} = \mathbf{F}^T\mathbf{y}$ (see [30, 36] for derivations of that and subsequent formulae). The optimal solution is thus given by

$$\mathbf{w}_L^* = \mathbf{V}^{-1}\mathbf{b}$$

This solution exists provided $\mathbf{V}$ is non-singular. It should be noted that special attention must be paid to the characteristics of the relationship between individual sequences of estimates. Collinearity among those sequences should be avoided, or otherwise the generalisation ability and robustness of the combination scheme deteriorates [36].

The corresponding optimal SSE is, after substituting for $\mathbf{w}_L^*$:

$$SSE_L^* = \mathbf{y}^T\mathbf{y} - \mathbf{b}^T\mathbf{V}^{-1}\mathbf{b}$$

Since we are mainly interested in online estimation, a recursive expression to the weights can be provided from the RLS algorithm (see Section 2.3), where the weight vector $\hat{\mathbf{w}}_k$ is updated on the basis of 1-step-ahead prediction error, $e_{k|k-1}$, and a gain vector, $\mathbf{k}_k$,

$$\hat{\mathbf{w}}_k = \hat{\mathbf{w}}_{k-1} + \mathbf{k}_k e_{k|k-1}$$

When combining two or more unbiased sequences of predictive estimates, constraints on the weights can be considered, imposing them to be nonnegative, or imposing the sum to be normalised to one.

Let us thus suppose that the constant term is zero and other weights are constrained to sum to unity, that is,

$$\begin{cases} w_0 = 0 \\ \mathbf{1}^T \mathbf{w} = 1 \end{cases}$$

where $\mathbf{w} = \begin{bmatrix} w_1 & \cdots & w_m \end{bmatrix}^T$ is now the vector of weights. Then, the combined error is

$$\mathbf{e}^{(c)} = \mathbf{E}\mathbf{w}$$

where $\mathbf{E} = \left[\mathbf{e}^{(1)} \mid \cdots \mid \mathbf{e}^{(m)} \right]^T$ is the matrix of respective forecast errors obtained by the individual models. An important property can be noted here, namely: combining the predictive estimates is equivalent to combining the respective forecast errors, in terms of the optimal weight vector. Hence, the optimal weights can be expressed entirely in terms of the forecast errors.

The optimal weight vector for the restricted linear combination is given by

$$\mathbf{w}_R^* = \frac{\mathbf{S}^{-1}\mathbf{1}}{\mathbf{1}^T \mathbf{S}^{-1}\mathbf{1}} \tag{4.2}$$

where $\mathbf{S}$ is the variance-covariance error matrix of the individual models, defined by

$$\mathbf{S} \equiv \left[s_{ij} \right]; \quad s_{ij} = \sum_{k=1+h}^{n} e_k^{(i)} e_k^{(j)}; \quad i, j = 1, ..., m \tag{4.3}$$

The corresponding optimal SSE is

$$SSE_R^* = \frac{1}{\mathbf{1}^T \mathbf{S}^{-1}\mathbf{1}}$$

An original recursive version for this framework can be given through the variance-covariance error matrix defined by (4.3). Given all individual forecast errors calculated up to and including time *k*, it is easy to verify that the variance-covariance error matrix can be given by

$$\mathbf{S}_k = \sum_{i=1+h}^{k} \mathbf{e}_i \mathbf{e}_i^T$$

where $\mathbf{e}_k = \left[e_k^{(1)} \quad \cdots \quad e_k^{(m)} \right]^T$ is the vector of forecast errors given by the individual models at time instant *k*. Furthermore, using the matrix inversion lemma [28] and some further manipulation of matrix algebra, we obtain

$$\mathbf{S}_k^{-1} = \left(\mathbf{S}_{k-1} + \mathbf{e}_k \mathbf{e}_k^T \right)^{-1} = \mathbf{S}_{k-1}^{-1} \left(\mathbf{I}_m - \frac{\mathbf{e}_k \mathbf{e}_k^T \mathbf{S}_{k-1}^{-1}}{1 + \mathbf{e}_k^T \mathbf{S}_{k-1}^{-1} \mathbf{e}_k} \right)$$

This is a convenient recursive expression for computing the inverse of $\mathbf{S}_k$ and, then the corresponding weight vector can be computed as

$$\hat{\mathbf{w}}_k = \frac{\mathbf{S}_k^{-1} \mathbf{1}}{\mathbf{1}^T \mathbf{S}_k^{-1} \mathbf{1}}$$

The success of the linear combination method relies on avoiding the possible collinearity between the sequences of predictive estimates of the individual models, and on avoiding the correlation between the sequences of the corresponding forecast errors.

Equation (4.2) can be used to derive simpler formulae, assuming for instance that the individual sequences of forecast errors are independent. For the particular case of two models (see [4]), we may consider the weights to be simply defined in terms of the variances of the two sequences of forecast errors, as follows:

$$w_{R,i}^* = \frac{\left(\mathrm{Var}\left(\mathbf{e}^{(i)} \right) \right)^{-1}}{\left(\mathrm{Var}\left(\mathbf{e}^{(1)} \right) \right)^{-1} + \left(\mathrm{Var}\left(\mathbf{e}^{(2)} \right) \right)^{-1}}, \quad i = 1, 2 \tag{4.4}$$

It is possible to derive simple recursive or adaptive expressions for the weights based on (4.4). We next describe such adaptive expressions that can be used in the context of recursive estimation of time-varying parameters.

4.3 Adaptive Weights

The expressions for the combining weights given in the previous section assume that each sequence of forecast errors carries the same weight over time. As a possible way of partially coping with nonstationarity in the data, and also avoiding optimising the weights, we can make them adaptive, so they are estimated recursively. Since the given combining method is based on a linear parametric metamodel estimated with respect to the least-squares criterion, weight adaptation can be achieved through recursive expressions similar to those of the RLS algorithm.

The recursive/adaptation procedure is more flexible, namely for the cases when the models have not the same accuracy over time and allows a continuous revision of the weights as more observations come out. We concur with the point of view that, at each time-step, more weight should be given to the model with higher performance, thus hoping to potentially improve the combined error.

Here, we just briefly describe adaptive schemes for the following simple and constrained case ($w_0 = 0$ and $\mathbf{1}^T \mathbf{w} = 1$) of two models:

$$\hat{y}_{k+h|k}^{(c)} = w_{1,k}\hat{y}_{k+h|k}^{(1)} + w_{2,k}\hat{y}_{k+h|k}^{(2)}$$

According to the idea above, i.e., more weight should be given to "better" models, simple adaptation rules based on exponential forgetting procedure can be created, for instance:

$$w_{i,k} = \alpha \frac{\left(e_k^{(i)}\right)^{-2}}{\left(e_k^{(1)}\right)^{-2} + \left(e_k^{(2)}\right)^{-2}} + (1-\alpha) w_{i,k-1} \tag{4.5}$$

where α is a hyperparameter – the forgetting factor – chosen in the $(0,1)$ interval, and determines the extent to which past errors influence the combining weights. This is equivalent to:

$$w_{i,n} = \alpha \sum_{k=1}^{n} (1-\alpha)^{n-k} \frac{\left(e_k^{(i)}\right)^{-2}}{\left(e_k^{(1)}\right)^{-2} + \left(e_k^{(2)}\right)^{-2}} + (1-\alpha)^n w_{i,0}$$

This adaptation scheme indicates that more weight should be given to the most recent estimates and less to the distant past estimates.

Since $(1-\alpha)<1$, the initial estimate of $w_{i,0}$ will become negligible with *n* large enough. Because at the beginning of the adaptation process no prior knowledge is given about the precision of the individual models, one may to assume equal weights, that is, $w_{1,0}=w_{2,0}=1/2$.

An illustrative example can be provided to show a possible behaviour of scheme (4.5) for different values of α. For this purpose, let us consider two different sequences of 200 simulated one-step-ahead forecast errors from a normal distribution: both with zero mean but with different variances, 1.0 and 2.0. The combined forecast errors come from directly applying formula (4.5) to adapt the weights and then combining the individual errors. Figure 4.1 shows the evolution in time of the weight $w_{1,k}$ for different values of α.

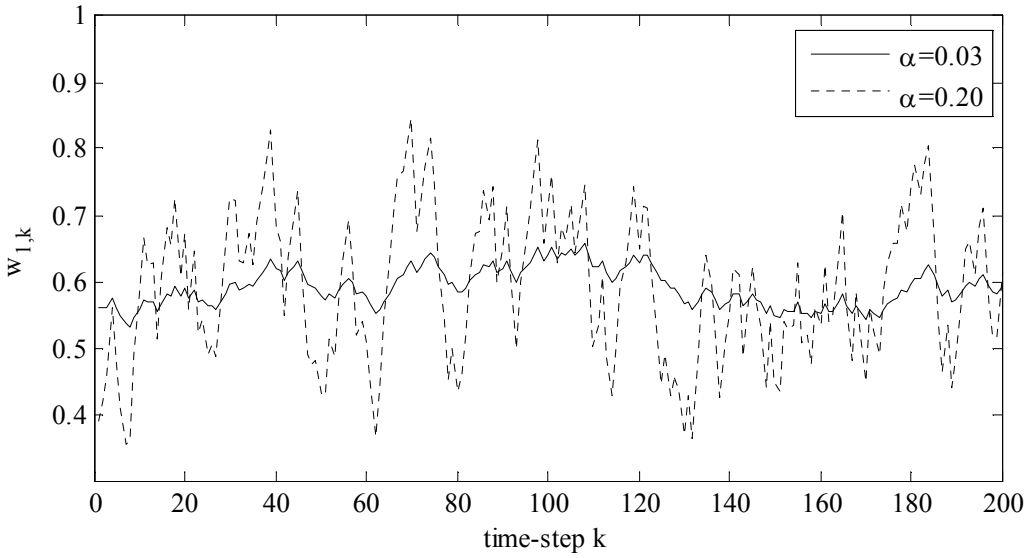

Figure 4.1: Illustration of the responses of combining weights in an adaptive framework given different values for the forgetting factor coefficient.

A small α produces a slow response to a change in the precision of individual models; correspondingly, a large value for α produces a rapid response, but also will make the weights respond to irregular changes in the accuracy. In general, through optimisation of α one finds that it is preferable to choose α close to zero, so that each new forecast error changes the weight very little.

The scheme defined by (4.5) can be extended to include additional past forecast errors. A general formula can then be considered, defined by

$$w_{i,k} = \alpha f_i\left(e_{1+h}^{(1)},...,e_k^{(1)},e_{1+h}^{(2)},...,e_k^{(2)}\right)+\left(1-\alpha\right)w_{i,k-1}$$

where f_i is a function of past forecast errors. In particular, one may define

$$f_i\left(e_{1+h}^{(1)},...,e_k^{(1)},e_{1+h}^{(2)},...,e_k^{(2)}\right)=\frac{S_{v,\lambda,k}^{(i)}}{S_{v,\lambda,k}^{(1)}+S_{v,\lambda,k}^{(2)}}$$

where $S_{v,\lambda,k}^{(i)}$ is the inverse of a sum of weighted squared forecast errors, given by

$$S_{v,\lambda,k}^{(i)}=\left(\sum_{t=k-v+1}^{k}\lambda^{k-t}\left(e_t^{(i)}\right)^2\right)^{-1};\quad 1\le v\le k,\ 0<\lambda\le 1$$

This general framework includes two particular cases, namely $\lambda=1$ or $v=k$, that have been formerly proposed by Winkler and Makridakis [73]. The first one defines a sliding time window formed by the last *v* forecast errors, giving same weight to those errors, and the second one is based on exponential weighting of past prediction errors. The hyperparameters *v* or λ can be optimised, but default values can also give reasonably good results. The exponential forgetting paradigm can thus be consistently used throughout, setting $0<\alpha=\lambda<1$, and using the exponential forgetting version of RLS to estimate the linear time-varying parameters of each predictive model.

In the exponential weighting methodology, it is usually assumed that the forgetting factor coefficient α is fixed over time but adaptive smoothing methods can also be considered. In these, α is re-estimated, say α_k, that is, it adapts itself to changes in the underlying accuracy of individual models, as each new observation becomes available. Such adaptation rules intend to revise the smoothing constants when the tracking signal gets outside the control limits. Various suggestions for choosing α_k have been proposed, including the well-known Trigg and Leach heuristic method [70].

More complex adaptation rules, yet straightforward to compute, can be considered. In particular, Yang [76] proposed an approach to combine different forecast estimates based on Gaussian functions:

$$w_{i,k} = \frac{w_{i,k-1}\left(s_k^{(i)}\right)^{-1/2} g_{ki}}{\sum_{j=1}^{2} w_{j,k-1}\left(s_k^{(j)}\right)^{-1/2} g_{kj}} \tag{4.6}$$

where $g_{ki} = \exp\left(-\left(e_k^{(i)}\right)^2 \Big/ 2s_k^{(i)}\right)$ and $s_k^{(i)}$ is an estimation of the variance of forecast errors produced by the model *i* at time instant *k*. The weights calculated by formula (4.6) are nonnegative and normalised.

4.4 Extended Linear Combination

In this section we propose an extension to the classic linear combination of estimates (initially introduced by Freitas and Rodrigues [27]), where nonlinear terms are included to cope with possible nonlinearities that may exist between the different sequences of estimates.

Naturally, there are similarities, to some extent, among the estimates produced by different models. It is common to find a more noticeable disagreement in small sized errors from different models or methods than in larger ones. This is more obvious in the case of outliers being present in the data. Hashem [36] has alerted for the possible collinearity that might exist among different sequences of estimates, as we already have mentioned, and that can undermine the robustness – and, therefore, the generalisation ability – of the combined model.

Our approach is motivated by the following: suppose we have two forecast errors given by different forecasting models. Either they have the same sign or they have different ones. If the signs are different, the error can be reduced drastically with just a simple average. However, with equal signs, we don't expect any improvements by simply averaging them. Thus, the combined forecast error is expected to be mostly improved in the cases when the errors have opposite signs when averaging them. Our aim is to take into account the correlation that usually exists between the sequences of estimates to be combined.

We next propose a formulation for a nonlinear combination of estimates that stays linear in the parameters. We restrict the discussion to two models not only for the sake of simplicity, but also because this might be adequate enough in many cases in practice. As we will see, the proposed formulation is an extension to the usual linear combination considered by other authors, with the potential ability to improve the performance of that combination.

Let $\left\{\hat{y}_{k|k-h}^{(1)}\right\}$ and $\left\{\hat{y}_{k|k-h}^{(2)}\right\}$ be the sequences of h-steps-ahead predictive estimates from two different models. We consider the following *extended* linear weighting combination (XLC):

$$\hat{y}_{k|k-h}^{(c)} = w_0 + w_1\hat{y}_{k|k-h}^{(1)} + w_2\hat{y}_{k|k-h}^{(2)} + \pi\hat{y}_{k|k-h}^{(1)}\hat{y}_{k|k-h}^{(2)}$$

The nonlinear term is included in order to take into account the possible correlation between the two sequences of corresponding errors. As stated before, each sequence of forecast errors should have white noise properties and should be independent of each other.

In matrix notation, we have

$$\hat{\mathbf{y}}^{(c)} = \mathbf{F}\mathbf{w} + \pi\mathbf{f},$$

where $\mathbf{w} = \begin{bmatrix} w_0 & w_1 & w_2 \end{bmatrix}^T$, $\mathbf{F} = \left[\mathbf{1} \mid \hat{\mathbf{y}}^{(1)} \mid \mathbf{y}^{(2)} \right]$ and $\mathbf{f} = \begin{bmatrix} \hat{y}_{1+h}^{(1)}\hat{y}_{1+h}^{(2)} & \cdots & \hat{y}_{n}^{(1)}\hat{y}_{n}^{(2)} \end{bmatrix}^T$. The respective SSE is defined by

$$\begin{aligned} SSE &= \left(\mathbf{y} - \mathbf{F}\mathbf{w} - \pi\mathbf{f}\right)^T \left(\mathbf{y} - \mathbf{F}\mathbf{w} - \pi\mathbf{f}\right) \\ &= \mathbf{y}^T\mathbf{y} - 2\mathbf{w}^T\mathbf{b} + \mathbf{w}^T\mathbf{V}\mathbf{w} + 2\pi\mathbf{w}^T\mathbf{z} - 2\pi d + \pi^2 u \end{aligned} \tag{4.7}$$

where $\mathbf{b}$ and $\mathbf{V}$ as before, and $\mathbf{z} = \mathbf{F}^T\mathbf{f}$, $d = \mathbf{f}^T\mathbf{y}$ and $u = \mathbf{f}^T\mathbf{f}$. Differentiating (4.7), and setting the normal equations:

$$\begin{cases} -\mathbf{b} + \mathbf{V}\mathbf{w} + \pi\mathbf{z} = \mathbf{0} \\ \mathbf{w}^T\mathbf{z} - d + \pi u = 0 \end{cases}$$

we get the optimal solution:

$$\begin{cases} \mathbf{w}^* = \mathbf{V}^{-1}\left(\mathbf{b} - \pi^*\mathbf{z}\right) \\ \pi^* = \dfrac{d - \mathbf{z}^T\mathbf{V}^{-1}\mathbf{b}}{u - \mathbf{z}^T\mathbf{V}^{-1}\mathbf{z}} \end{cases} \tag{4.8}$$

This solution exists provided $\mathbf{V}$ is not singular and $\mathbf{F}\mathbf{F}^+ \neq \mathbf{I}$, where $\mathbf{F}^+$ is the Moore-Penrose pseudoinverse of matrix $\mathbf{F}$. In case $\mathbf{F}\mathbf{F}^+ = \mathbf{I}$ one gets $u = \mathbf{z}^T\mathbf{V}^{-1}\mathbf{z}$ and thus $d = \mathbf{z}^T\mathbf{V}^{-1}\mathbf{b}$, leading to an indeterminate value for π^*.

The corresponding optimal SSE is, after substituting for $\mathbf{w}^*$ and π^*:

$$SSE^* = \mathbf{y}^T\mathbf{y} - \mathbf{b}^T\mathbf{V}^{-1}\mathbf{b} - \frac{\left(d - \mathbf{z}^T\mathbf{V}^{-1}\mathbf{b}\right)}{u - \mathbf{z}^T\mathbf{V}^{-1}\mathbf{z}} \tag{4.9}$$

Formulae (4.8) and (4.9) are valid irrespectively of the nature of vector $\mathbf{f}$. Therefore, these can be viewed as an extension of the classic linear form to the case where there is an additional sequence of estimates to be considered in the combination. Indeed, those formulae can be written in terms of the classical solutions, yielding:

$$\begin{cases} \mathbf{w}^* = \mathbf{w}_L^* - \pi^* \mathbf{V}^{-1}\mathbf{z} \\ \pi^* = \dfrac{d - \mathbf{z}^T \mathbf{w}_L^*}{u - \mathbf{z}^T \mathbf{V}^{-1}\mathbf{z}} \end{cases} \tag{4.10}$$

and

$$SSE^* = SSE_L^* - \frac{\left(d - \mathbf{z}^T \mathbf{w}_L^*\right)}{u - \mathbf{z}^T \mathbf{V}^{-1}\mathbf{z}} = SSE_L^* - \pi^* \mathbf{f}^T \mathbf{e}_L^{(c)} \tag{4.11}$$

where $\mathbf{e}_L^{(c)}$ is the sequence of errors obtained from the classic linear form. Formulae (4.10) and (4.11) are generalisations of the known solutions for the restricted formulation, where $\pi = 0$.

The sign of π^* is directly associated to the sign of the correlation between the sequence of errors given by the classic linear form and the sequence of values defined by the nonlinear term in the proposed combination formula. Thus, if these sequences are sufficiently uncorrelated, there is no significant advantage in considering the proposed extended formula instead of the usual one.

It is easy to show that, consistently,

$$SSE^* \leq SSE_L^* \leq SSE_R^* \leq \min\left\{SSE^{(1)*}, SSE^{(2)*}\right\} \tag{4.12}$$

where $SSE^{(1)*}$ and $SSE^{(2)*}$ are the minima associated to the individual models. We note that, however, in practice the chain relations (4.12) might not always hold, that is, there is no guarantee that the out-of-sample forecast error is reduced by combining. This is mostly due to data sampling errors that predominantly contaminate the combining weight estimates. Moreover, the collinearity that typically exists among the individual forecasts leads to an intensification of those sampling errors.

The recursive and adaptive schemes given in previous sections 4.2 and 4.3 can also be employed to adapt any combining weight in the extended formulation, including π.

4.5 Experimental Results

We briefly illustrate the application of either approach for combining two suboptimal radial basis function networks — the classic one ($\pi = 0$) and the extended one (unconstrained π). For each of the two combining methods, we have considered two different ways to estimate the weights, either based on the unrestricted formulation or on the restricted one. In all methods, we used the exponential weighting paradigm to adaptively estimate the weights, with forgetting factor coefficient λ. The methodologies used are summarised as following:

— CLC (*classic linear combination*): unconstrained $\mathbf{w}$ and $\pi = 0$;

— CRLC (*classic restricted linear combination*): $w_0 = 0$, $\mathbf{1}^T \mathbf{w} = 1$ and $\pi = 0$;

— XLC (*extended linear combination*): unconstrained $\mathbf{w}$ and π;

— XRLC (*extended restricted linear combination*): $w_0 = 0$ and $\mathbf{1}^T \mathbf{w} + \pi = 1$.

For this experiment, we have used the pseudo-best model M0 obtained from a Gaussian RBFN, for each of the 9 simulated time series introduced in Section 3.4.

Keeping most of the optimised values fixed, we have estimated, for each time series, four more RBFNs (MR1, MR2, MS1 and MS2): in the first two models, the centres were chosen randomly from the input patterns; in the last two models, we have considered different values for the widths: slightly above and slightly below the optimal ones.

In Table 4.1, we compare the relative performance of the five models, as well as the relative performance of the formulations described before for the linear combination. The average of the relative performance over the 9 time series is reported.

These results confirm what would be expected. While, in these examples, the improvement in predictive performance is somewhat marginal, the proposed approach might be of much greater value in other cases. A more exhaustive experimental study on the classic approach of combining many Gaussian RBF forecasting models – differing in centre locations, widths values, etc. – was done by Freitas [26].

Model	*Avg. perf. index* $\bar{J}(1)$ *(%)*	*Avg. perf. index* $\bar{J}(4)$ *(%)*
M0: clustering-based centres	1.6	73.8
MR1: random centres	17.2	91.2
MR2: random centres	9.6	88.0
MR1, MR2 combined using CLC	8.0	84.5
MR1, MR2 combined using CRLC	7.7	84.3
MR1, MR2 combined using XLC	7.5	84.7
MR1, MR2 combined using XRLC	7.7	84.1
MS1: smaller widths	13.1	82.8
MS2: larger widths	7.5	79.7
MS1, MS2 combined using CLC	3.4	75.4
MS1, MS2 combined using CRLC	3.2	74.7
MS1, MS2 combined using XLC	3.6	76.6
MS1, MS2 combined using XRLC	3.2	75.0

Table 4.1: Average performance of some individual models and their combinations over 9 series.

CHAPTER 5

Combining Decisions

Contents

Abstract

Two combination approaches for optimal decision-making based on forecasts are discussed: either inferring decisions from combined predictive estimates, or combining prescriptive solutions derived from different forecasting models.

5.1 Combining Decisions vs. Combining Predictions

Prediction is just a means – albeit a very important one – of supporting decision-making.

Decision-making may sometimes be treated separately as a process and an outcome. Separating process and outcome is often convenient because it helps to explain that a good decision-making process does not guarantee a good outcome, and that a good outcome does not presuppose a good process.

In the context of prediction-decision problems, while optimal decisions are favoured by the availability of optimal forecasts, optimal decision-making does not require optimal forecasting. We may identify two main approaches:

— with emphasis on the prescriptive role of the global model (e.g., reinforcement learning [67]);

— with emphasis on the predictive role of the global model.

Based on these approaches, we aim to discuss two possible combining approaches in the context of optimal decision-making (initially introduced by Freitas and Rodrigues [27]):

— *(CDec)* first, for each of the predictive models, determine the corresponding best decision, and then combine (using optimal weights) those decisions; or,

— *(CPred)* first, combine (using optimal weights) the predictive estimates produced by distinct models and then determine the corresponding best decision.

The key issue here is that, while the predictive model can be conveniently estimated with respect to the least-squares (LS) criterion, the prescriptive model is, in general, assessed in terms of a more realistic least-cost (LC) non-equivalent, and non-differentiable, performance measure.

Instead of trying to "predict the best decision to make" (e.g., trying to predict quantiles – see [11]), perhaps one should rather compute an "optimal" decision from a forecast of the density of the future observation (regarded as a random variable), and considering the appropriate cost function for the ultimate (practical) problem to solve. Thus, the prediction method should produce density forecasts, namely using information from past errors. It may not suffice, or even be useful for decision-making, to compute prediction intervals. And, if the errors are correlated, it does not suffice to estimate the overall variance.

In view of that, one might conjecture that the second of the above two approaches (CPred) could be, in general, preferable, and that one would benefit from improved performance as early as possible in the predictive-prescriptive global model. However, as we will see, this might depend on several factors, including the nature of the LC function and the time horizon of the predictive estimates.

Moreover, while supervised neural networks can be used as prescriptive models, with outputs representing the proposed decisions, when addressing model mixing issues, it might be better to use them only for predictive purposes, since supervised learning is usually defined in terms of Euclidean distances and, therefore, the LS criterion.

5.2 Asymmetric Costs

Many real-world applications, namely in economics and business, require the computation of decisions, as quantiles of some control variable, sequentially in time, which are assessed in terms of an asymmetrical cost function. A typical example is inventory management: managing resources of equipment or goods in order to satisfy client demand. One of the difficulties of the problem lies on the stochastic and usually nonstationary behaviour of demand. In order to obtain "optimal" decisions associated to the management problem, predictions of future demand are required. Therefore, forecasting is almost inevitably a prerequisite for most business decisions. In inventory management, one has to determine the ideal quantity to be ordered so to satisfy the demand, at least partially, for a given time horizon. One common policy is to repeat the ordering procedure at regular time intervals, and on the basis of the analysis of a time series of total demand observed at past intervals.

Over-prediction of demand may result in stock-keeping costs, while under-prediction results in lost sales revenue. Hence, the costs arising from suboptimal decisions based on over- versus under-prediction are different for errors of identical magnitude.

In decision problems, such as the above, the traditional symmetric error functions, including those based on the LS criterion, should be abandoned in favour of more meaningful ones, where the sign of the errors plays a significant role. In particular, we may consider the piecewise linear asymmetric cost function (regarding the time horizon h), common in many real-world applications, defined by

$$(\min)\ LC(h) = \frac{1}{n - n_0} \sum_{k=n_0+1}^{n} d_{k|k-h} \tag{5.1}$$

where

$$d_{k|k-h} = \begin{cases} ue_{k|k-h}, & e_{k|k-h} \geq 0 \\ -ve_{k|k-h}, & e_{k|k-h} < 0 \end{cases}$$

The values u and v define the slopes of the linear sections of the LC function, and measure the unit costs associated to decisions by default or by excess, respectively. The ratio of u and v defines the degree of asymmetry between positive and negative decision errors. If

$u = v = 1$, the LC function is identical to the MAD cost function. Figure 5.1b shows two examples of LC functions.

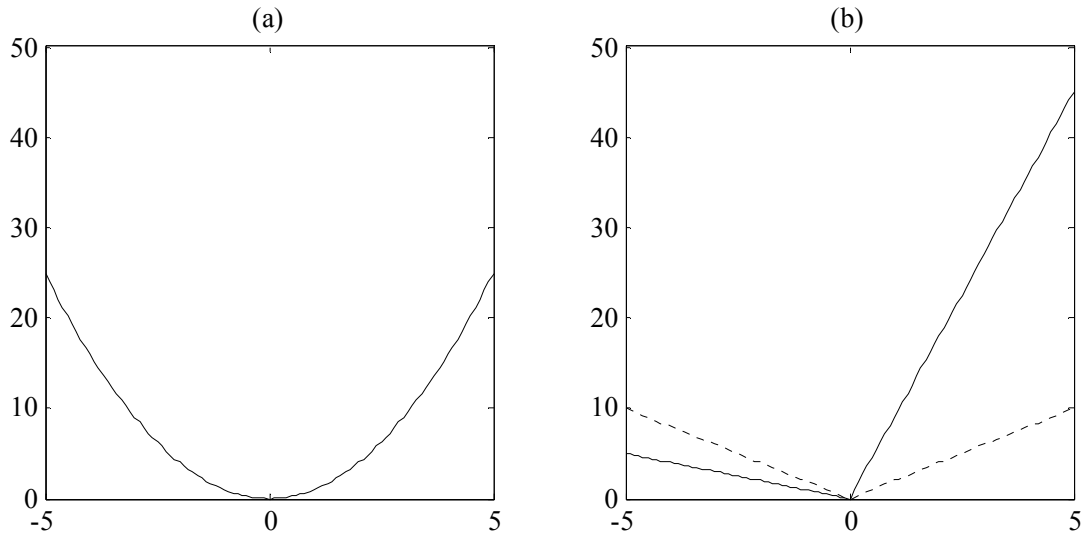

Figure 5.1: (a) The LS cost function; (b) Two examples of LC functions, for $u = 9$ *and* $v = 1$ *(solid line), and for* $u = v = 2$ *(dotted line).*

At time *k*, the optimal (minimum expected cost) decision for the next planning horizon (based on *h*-steps-ahead predictions), $Q_{k+h|k}$, can be defined as a quantile of the distribution of the random variable Y_{k+h}, conditional to the observations available up to time *k*:

$$F_k\left(Q_{k+h|k}\right) = \mathrm{P}\left[Y_{k+h} \le Q_{k+h|k}\right] = \frac{u}{u+v}$$

where F_k is the conditional distribution function of Y_{k+h}, which can be empirically estimated from past prediction errors, in the simplest case. This is illustrated in Figure 5.2 for a Gaussian distribution centred on forecast $\hat{y}_{k+h|k}$.

For the estimation of quantiles, it is convenient, but not really critical, that the errors have (Gaussian) white noise characteristics. If the errors are uncorrelated and stationary, one may simply use the sample distribution of the errors; otherwise, density forecasts may be heuristically computed using exponentially-weighted kernels [11].

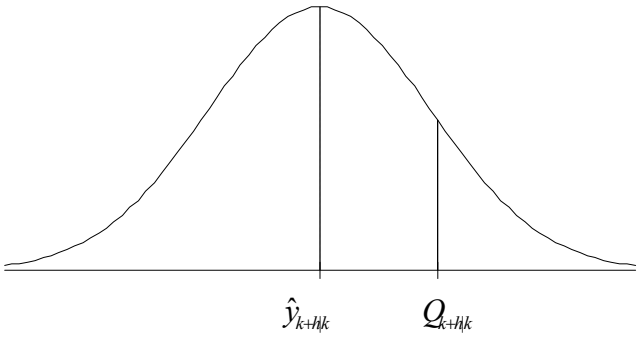

Figure 5.2: Quantile estimate from a Gaussian distribution.

Often, it is highly recommended to apply a preprocessing procedure (such as an appropriate nonlinear transformation) to the time series, or to the sample of training patterns, to favour desirable properties of the prediction errors, namely closeness to stationarity and normality (symmetry, in particular).

5.3 Experimental Results

To illustrate and assess the difference between the two approaches, we set up an experiment using 9 time series – named T1,T2,...,T9 –, of equal length, simulated according to DTR-IAR(1) models, assigning different combinations of values for the pair (α, NVR), as defined in Table 5.1. To make the series somewhat more interesting, the observation error of each time series, $\{\varepsilon_k\}$, was generated from an asymmetric Generalised Lambda Distribution, with zero mean and variance one.

	T1	T2	T3	T4	T5	T6	T7	T8	T9
α	0.75	0.82	0.87	0.90	0.92	0.94	0.96	0.98	0.99
NVR	10^{-3}	10^{-2}	10^{-4}	10^{-3}	10^{-2}	10^{-4}	10^{-3}	10^{-2}	10^{-4}

Table 5.1: Simulation settings for 9 time series.

For the sake of simplicity, we next describe and discuss only the results obtained for the series T4 (see Figure 5.3).

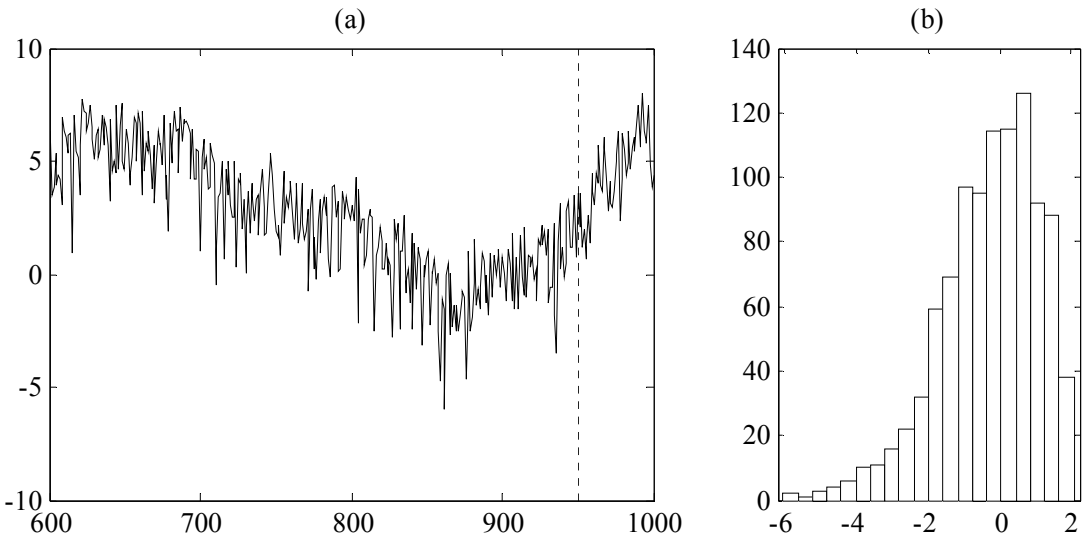

Figure 5.3: (a) Last 400 values of time series T4; (b) Histogram of observation error.

In order to assess out-of-sample predictive accuracy of the estimated models and approaches discussed here, the data were divided into two sets: a training set (first 950 values) and a test set (last 50 values). The former was used for identification and estimation tasks, whereas the latter was used only for generalisation purposes. Nonetheless, as usual in recursive estimation applications, the parameters were updated also during the test set period. The reason for choosing a small test set comparing to that of the training set is mainly due to the nonstationarity feature of the data and the consideration of constant weights in the combination paradigm for both approaches, CPred and CDec. Furthermore, the combining weights were estimated "*en-block*" using only the training set, and weren't recursively updated in the test set, principally due to the difficulties raised by the LC loss function in the CDec approach.

For each of two time horizons, 1 and 12, two different suboptimal forecasting models have been considered: the DTR-RW ($\alpha = 0$) and the DTR-IRW ($\alpha = 1$) models, with optimised noise variance ratios. The best results are shown in Table 5.2.

	Prediction (*RMSE*(1)*)	*Prediction* (*RMSE*(12)*)
DTR-RW	1.50 (10^{-2})	1.84 (10^{-1})
DTR-IRW	1.27 (10^{-4})	1.51 (10^{-5})

Table 5.2: T4 results: Best values attained by two predictive models, in the generalisation phase, for two time horizons, 1 and 12. The best NRVs are indicated in parentheses.

The last 200 one-step-ahead predictive estimates produced by the two models are shown in Figure 5.4. In this case, it is noticeable the difference among the predictive estimates, particularly in the test set.

Then, we have considered the same extended linear combination scheme for both approaches (CPred: combining predictions; CDec: combining decisions):

$$\hat{y}^{(c)}_{k+h|k} = w_0 + w_1 \hat{y}^{(1)}_{k|k+h} + w_2 \hat{y}^{(2)}_{k+h|k} + \pi\, \hat{y}^{(1)}_{k+h|k}\, \hat{y}^{(2)}_{k+h|k}$$

$$Q^{(c)}_{k+h|k} = w'_0 + w'_1 Q^{(1)}_{k+h|k} + w'_2 Q^{(2)}_{k+h|k} + \pi'\, Q^{(1)}_{k+h|k}\, Q^{(2)}_{k+h|k}$$

where the weights were optimised with respect to the corresponding loss functions: LS for predictions and LC for decisions. At this point, we notice that the weights are considered constant in time, as already mentioned above, and thus we used the OLS algorithm to

estimate them in the CPred approach. The combining weights associated to the CDec approach were obtained by means of regression quantiles through the Frisch-Newton interior point algorithm (described by Koenker and Portnoy [41]). We set $u = 9$ and $v = 1$.

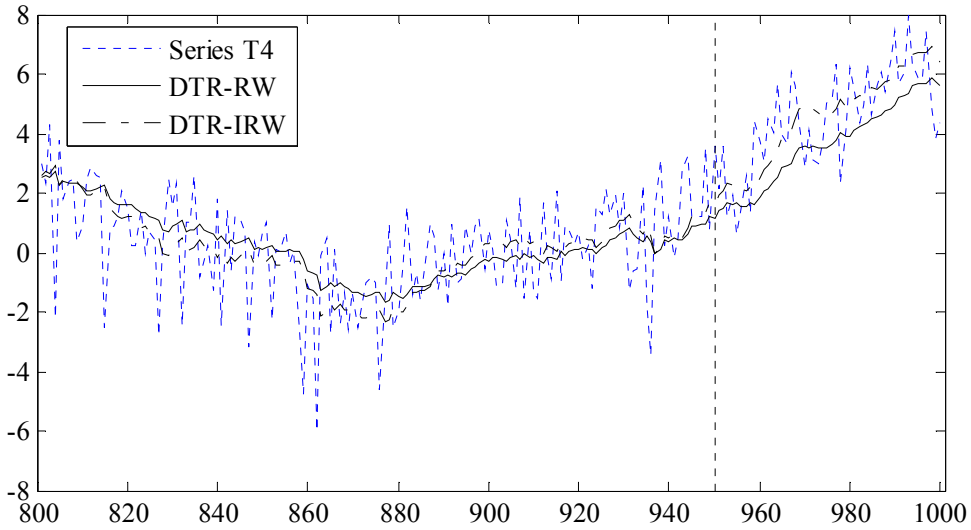

Figure 5.4: One-step-ahead predictive estimates from models DTR-RW and DTR-IRW, for time series T4 (last 200 values).

To determine the quantiles that define the optimal decisions, one needs to estimate the distribution function associated to the random variables Y_k. To simplify this process, we may empirically determine those quantiles by sorting (ranking) the last N (say, the last 200) estimated forecast errors in ascending order to obtain a sequence $z_1 \le z_2 \le \cdots \le z_N$. The qth-quantile z_q is then obtained by taking the rank qth-value $z_{(N+1)q}$ (or an average of neighbouring values if $(N+1)q$ is not an integer):

$$z_q = \begin{cases} z_{(N+1)q}, & \text{if } (N+1)q \text{ is an integer} \\ 0.5\left(z_{\lfloor (N+1)q \rfloor} + z_{\lfloor (N+1)q \rfloor + 1}\right), & \text{otherwise} \end{cases}$$

In addition, this allowed us to overcome the problem of non-normality and autocorrelation of the errors that were obtained by the individual models (see Figure 5.5). We notice that those errors are not quite normally distributed, contrarily to those used in the simulation process.

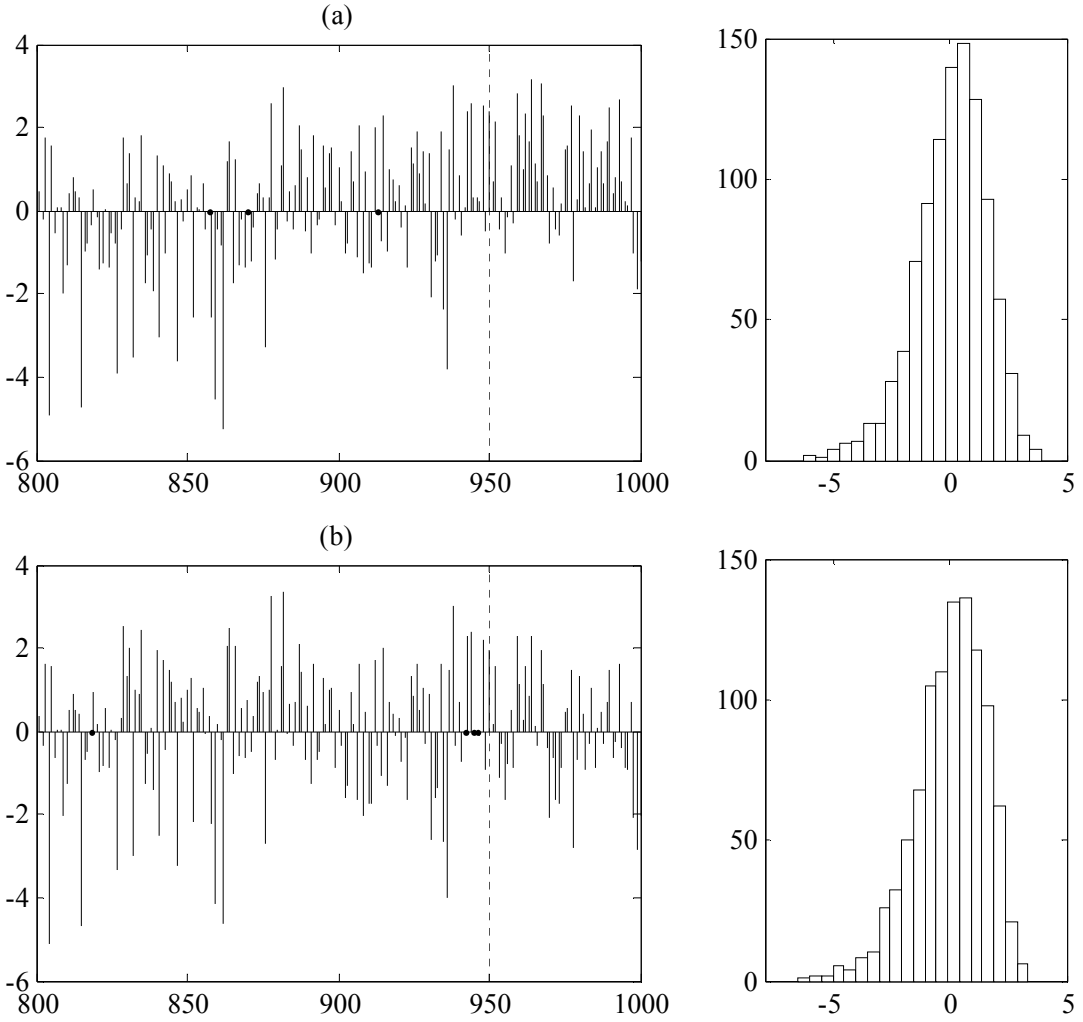

Figure 5.5: One-step-ahead forecast errors (left) and the corresponding histograms (right) from models applied to T4: (a) DTR-RW; (b) DTR-IRW.

Table 5.3 shows the results (with respect to generalisation accuracy) obtained by the individual models and by their combination related to prediction and decision problems, for each time horizon 1 and 12.

	Prediction (*RMSE*(1)*)	*Decision* (*LC*(1)*)	*Prediction* (*RMSE*(12)*)	*Decision* (*LC*(12)*)
DTR-RW	1.50	2.59	1.84	2.15
DTR-IRW	1.27	2.15	1.51	2.50
Optimal comb.	1.27	2.04	1.51	2.27

Table 5.3: Series T4: Results of individual and combined models with respect to prediction and decision optimisation problems, in terms of generalisation, for two time horizons, 1 and 12.

In either approach, the combination of models could not outperform the individual ones, except for the combination of decisions for time horizon 1. The quantiles and the histogram of the corresponding prescription errors, obtained by each approach for time horizon 1, are shown in Figure 5.6 (last 200 values). As expected, most of the errors are negative (about 90% of the values), due to the degree of asymmetry among costs (i.e., the ratio of u and u) – in this case, $u / v = 9$.

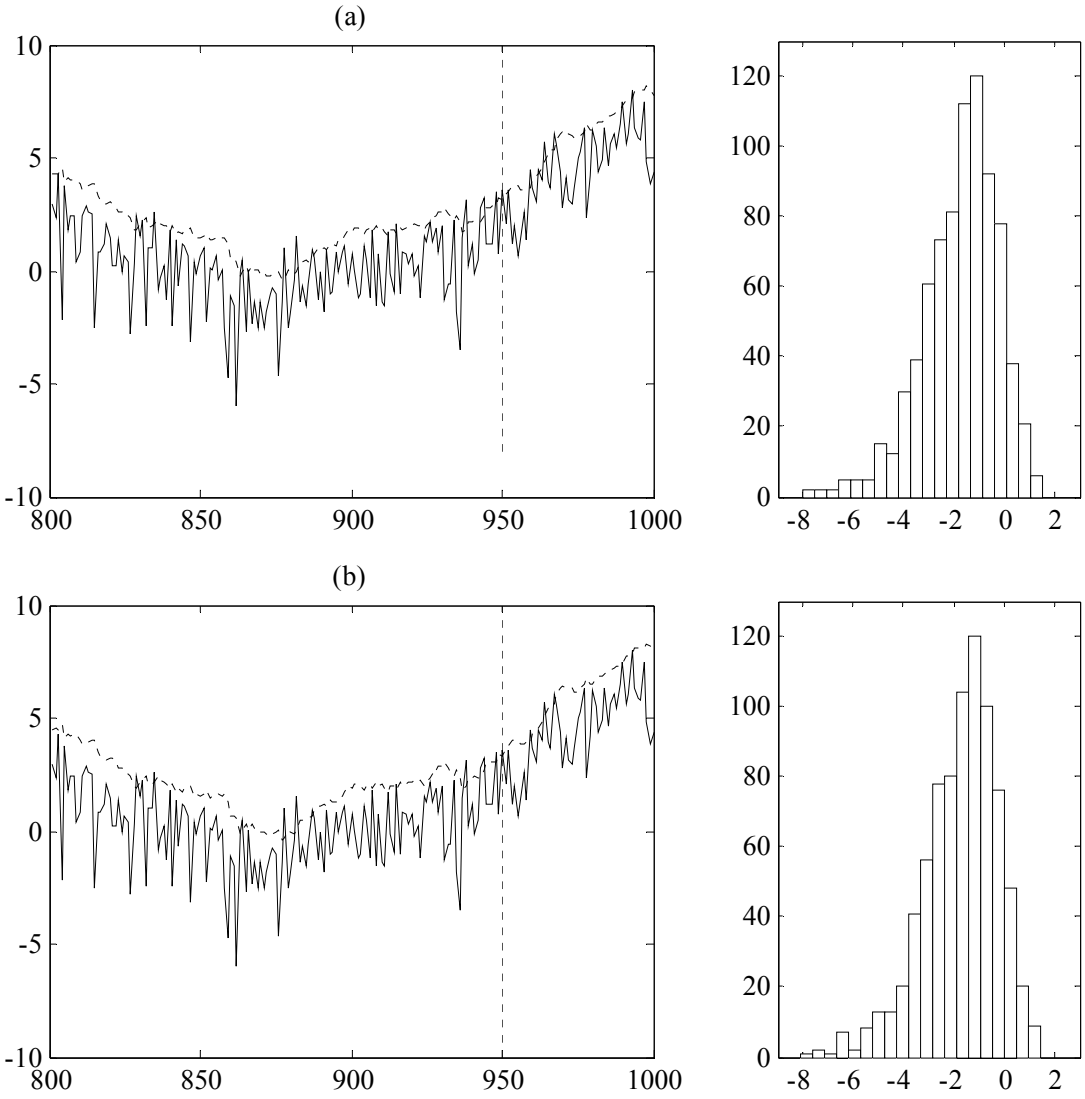

Figure 5.6: Series T4: Sequences of quantiles (dotted line on left) and the histogram of the corresponding prescription errors (left) resulting from two approaches: (a) CPred; (b) CDec.

The decision costs and respective histograms, resulting from each approach and time horizon 1, can be seen in Figure 5.7 (last 200 values); it is noticeable that the two sequences are somewhat dissimilar.

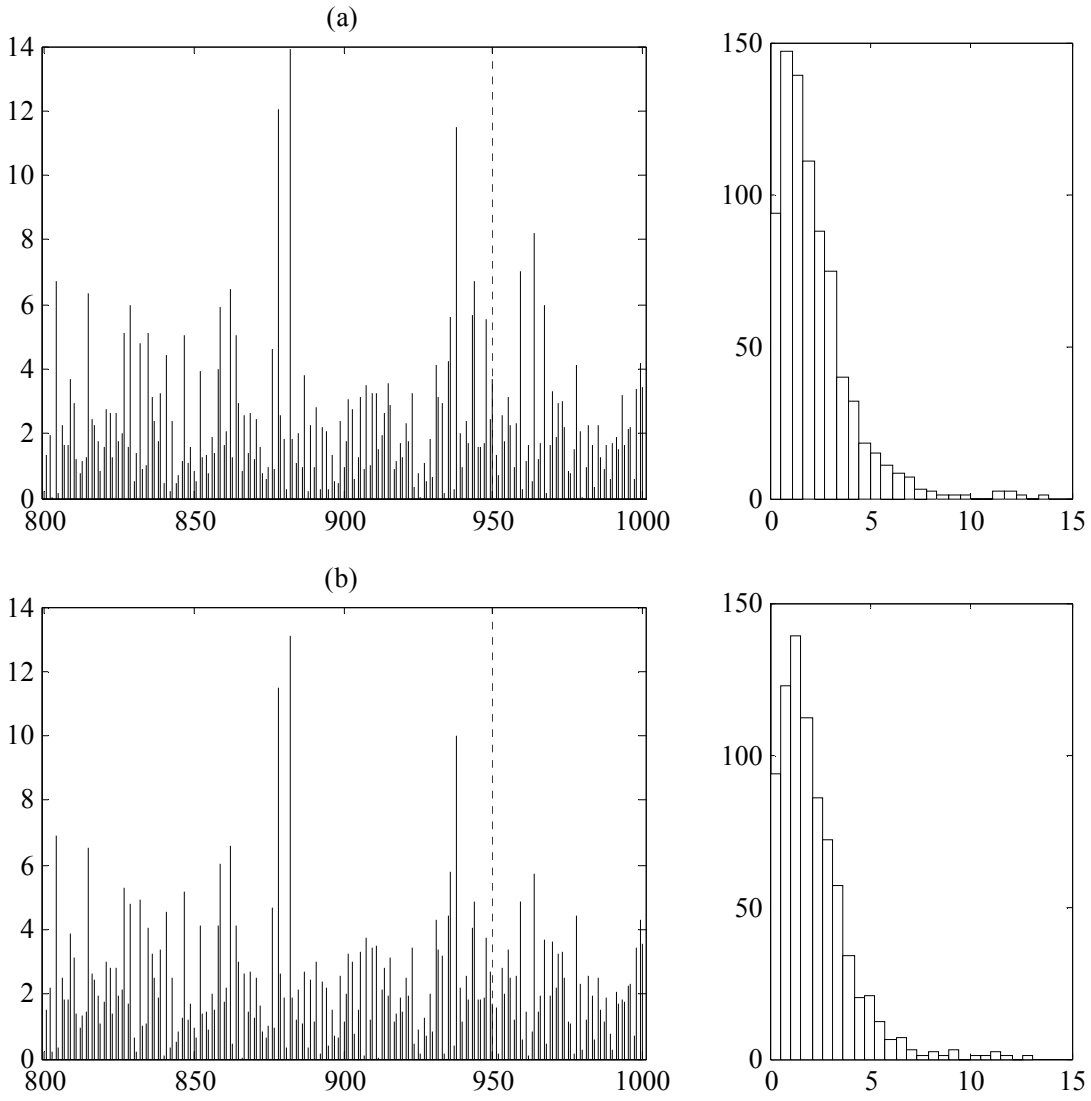

Figure 5.7: Series T4: Sequences of quantiles (stems on left) and respective decision costs (right) resulting from two approaches: (a) CPred; (b) CDec.

The minimum average costs were: for approach CPred, $LC(1) = 2.10$ and $LC(12) = 2.69$; and, for approach CDec, $LC(1) = 2.04$ and $LC(12) = 2.27$. This means that, in the present time series, it was preferable to combine quantiles, rather than to combine predictive estimates, for either time horizon. The difference between the methodologies was more noticeable for time horizon 12. This may suggest different behaviours for different time horizons, and perhaps for different degrees of asymmetry among costs.

If we now apply the same procedures for different time horizons and degrees of asymmetry among decision costs, we may better understand how each approach may behave. Therefore, the previous steps were extended to all simulated time series. Afterwards, we computed the ratio index defined below, for a range of time horizons (*h*) and a degree of asymmetry among decision costs ($\beta = u/v$), for each time series:

$$J_i(h,\beta) = \frac{LC_{CPred}^{(i)}}{LC_{CDec}^{(i)}}, \quad i = 1,2,...,9$$

where $LC_{CPred}^{(i)}$ and $LC_{CDec}^{(i)}$ are the minimum averages costs for approaches CPred and CDec, respectively, applied to time series T_i.

For series T4, the results can be seen in Figure 5.8. The following combinations were considered in obtaining those results:

$$h \in \{1,2,...,19,20\}, \quad \beta \in \left\{2^x : x = -4,-3.5,...,3.5,4\right\}$$

In this example, CDec almost always led to better results than CPred. To some extent, the time horizon and the degree of asymmetry among decision costs seem to contribute to the discrepancy between the approaches.

More general results, namely the mean and the sample standard deviation over indices $\{J_i\}$, are shown in Figure 5.9. Although, some precautions should be taken in interpreting these new results, we notice that CPred led now, in average, to better results than CDec, and was also proven to be less volatile than the latter everywhere, at least for this set of simulated time series presently considered. For the sake of evidence, we report also, for instance, the individual results for time series T3 and T7 – see Figure 5.10. It is noticeable how the ratio of LC values obtained by methodologies CPred and CDec may differ, even among data sets with some similar characteristics.

We conclude that the forecasting horizon and the degree of asymmetry among decision costs may strongly influence the performance of either approach, but it is unclear how this may depend on characteristics of the time series. Nonetheless, the above experiments suggest that, while the application of CPred is computationally more tractable, CDec should also be considered, for possible significant cost reductions.

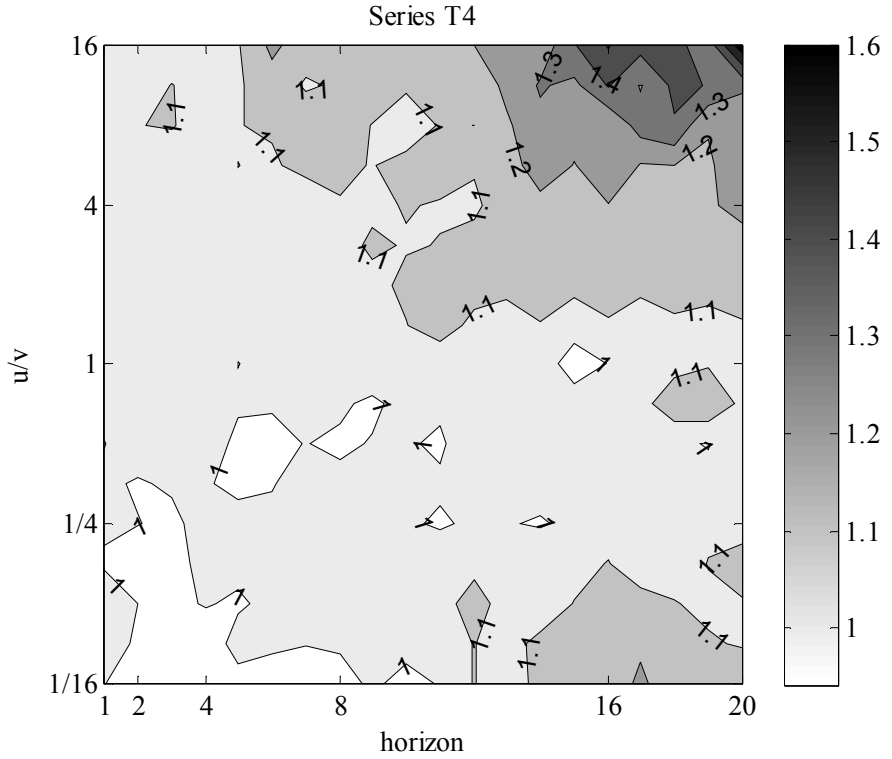

Figure 5.8: Series T4: Contour plot of the average of the ratio of LC values obtained by methodologies CPred and CDec, by varying the time horizon and degree of asymmetry among decision costs.

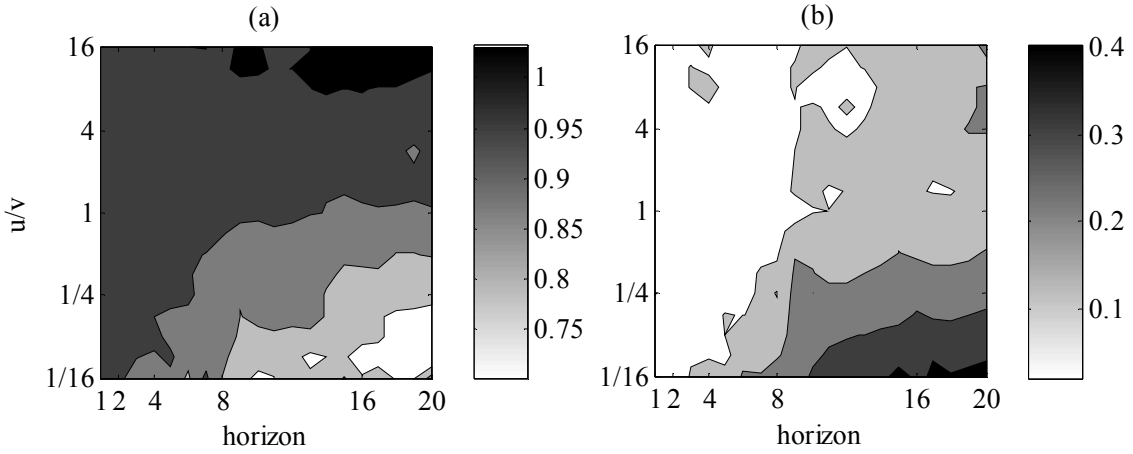

Figure 5.9: Contour plots of the ratio of – (a) averages, and (b) sample standard deviations – of the ratio of LC values obtained by methodologies CPred and CDec, over 9 time series, by varying the time horizon and the degree of asymmetry among decisions costs.

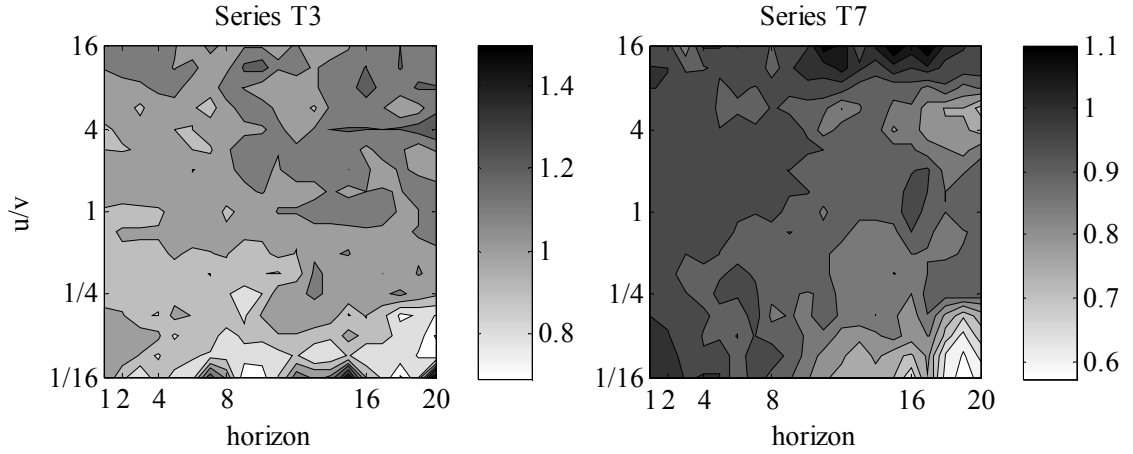

Figure 5.10: Contour plots of the ratio of LC values obtained by methodologies CPred and CDec, by varying the time horizon and degree of asymmetry among decision costs, for two time series.

CHAPTER 6

Case Studies

Contents

Abstract

We illustrate the application of some of the models and methods proposed and discussed in the previous Chapters to two case studies, related to the analysis of the Lisbon Stock Exchange Index PSI20, and the time series of monthly totals of guests entered in Madeira, Portugal.

6.1 Financial Data

The time series we intend to study here is PSI20, the main Index of the Lisbon Stock Exchange, which reflects the evolution of the prices of 20 selected Portuguese companies. In fact, two series were considered, recorded in two distinct periods: PSI20W – a series of weekly close prices – and PSI20D – a series of daily close prices – see Appendices A.2 and A.3, respectively. The stochastic, nonstationary, and nonlinear characteristics exhibited by either series, make them very difficult to be modelled and forecasted.

Our aim is ultimately to compute predictive estimates for short time horizons by means of the stochastic detrending methodology, where a DTR-IAR(1) model is used to cope with low-frequency effects in the data. Then, in a second stage, we identify a RBFN, with time-varying parameters, to deal with high-frequency effects and possibly nonlinear correlations. After the identification of each submodel, a hybrid model is composed and estimated. The Naïve1 method, where $\hat{y}_{k+h|k} = y_k$, for any horizon h, is also considered, for the sake of comparison. In case of a time series being truly a simple random walk, Naïve1 is the optimal forecasting method.

The study reported below starts with the analysis of the time series PSI20W and, afterwards, the time series PSI20D is handled in three different ways.

For each of the time series considered we divided it into two subsets: the training set (first 75% of the values) and the test set (last 25% of the values). The former was used for identification and estimation tasks, while the latter was used only for predictive (generalisation) assessments.

From the periodograms in Figures A.2 and A.3, one cannot detect any noticeable high-frequency effects on the data, other than possibly nonlinear correlations. Moreover, there is a high volatility in the time series of first-differences (or "returns") of PSI20W, but not in the first-differences of PSI20D (see Figure 6.1), making it even more difficult to predict, for any given time horizon.

In order to remove the low-frequency effects out from the periodogram, we identified a DTR-IAR(1) model. A relatively high horizon was considered in the optimisation of the hyperparameters, so to avoid catching high-frequency variations in the levels along time. Setting the time horizon to 15, we found $\left(\alpha^*, NVR^*\right) \approx \left(0.64, 1\right)$. These were chosen by comparing the performance in the training set over the following possibilities:

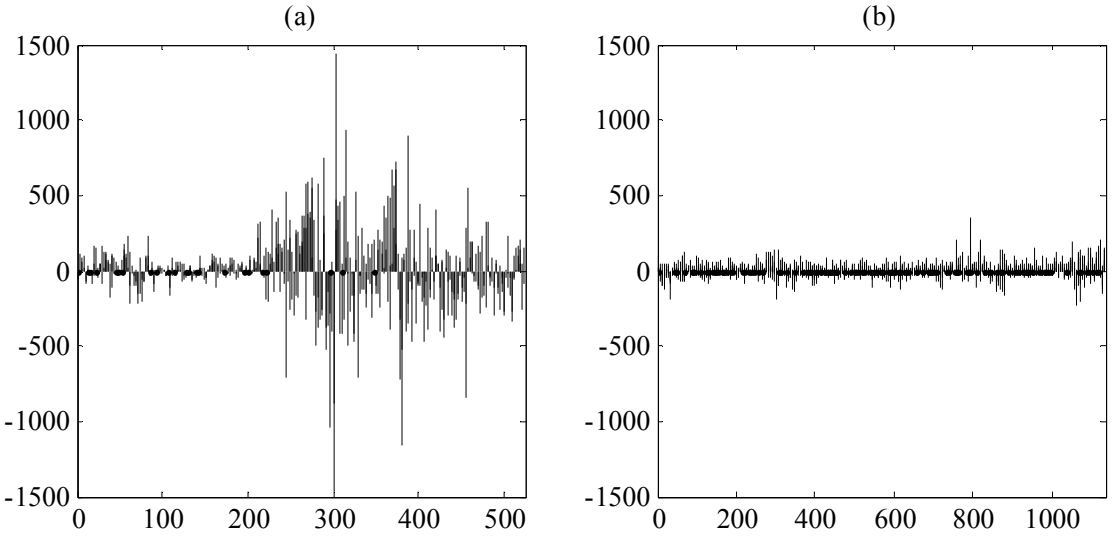

Figure 6.1: First-differences of: (a) PSI20W; (b) PSI20D.

$$\alpha \in \left\{ g(x) = \sqrt[6]{x/14} : x = 0,1,...,15 \right\}$$

$$NVR \in \left\{ h(x) = 10^{x} : x = -\infty, -5, -4.5, ..., 1.5, 2 \right\}$$

Note that, for $g(x)$, with $x = 15$, $\alpha \approx 1.01$, i.e., we allow α to be higher than 1. Furthermore, we also explore the extreme cases: $\alpha = 0$ and $\alpha = 1$.

Figure 6.2 shows the last 200 values of the 1-step-ahead predictive estimates obtained from the DTR-IAR(1) model mentioned above and the corresponding detrended series resulting from removing it out from series PSI20W. The dotted vertical line separates the training and the test sets. From the corresponding periodograms, one notices that the low-frequency effects were reasonably removed.

Making a further and deeper study in the identification of a DTR-IAR(1) by varying the time horizon, we noticed that the corresponding best values attained for the hyperparameters α and *NVR* were somewhat unexpected (see Figure 6.3). Explicitly, for time horizon 1, the noise variance ratio attained the value $NVR = 10^{0.5}$, for time horizons 2, 4, 10 and 15 it attained the value 1, and for time horizons 20 and 30, it attained the value 10^{2}, whereas α attained the values around 0.64, 0.64 and 0, respectively. For this time series, the higher the time horizon, the lower is α, while *NVR* goes down and then rises.

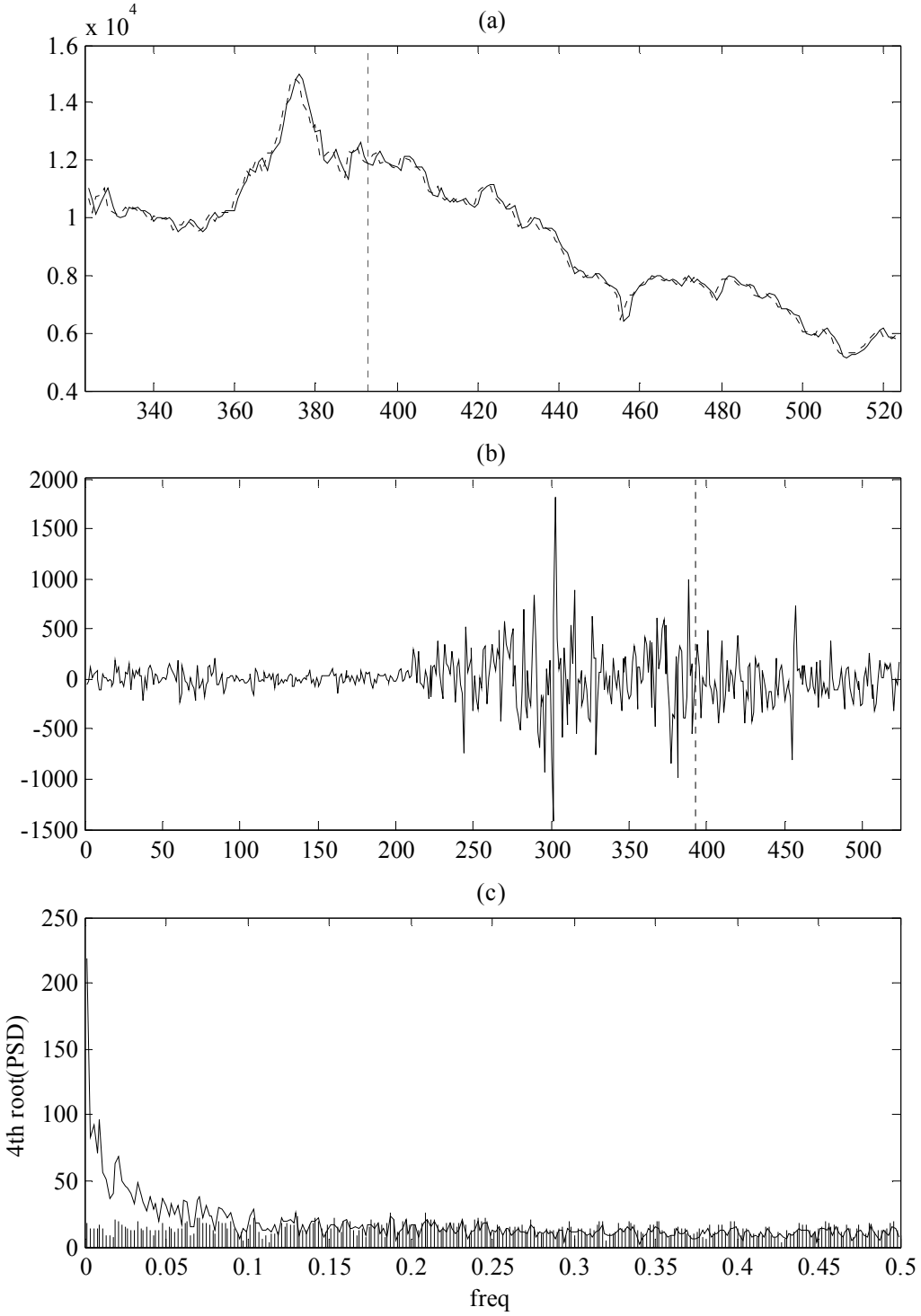

Figure 6.2: Series PSI20W: (a) Last 200 of the 1-step-ahead predictive estimates (solid line) of a DTR-IAR(1) model, for $\alpha = 0.64$ *and* $NVR = 1$*; (b) the detrended series; (c) the periodogram of the detrended series (stems).*

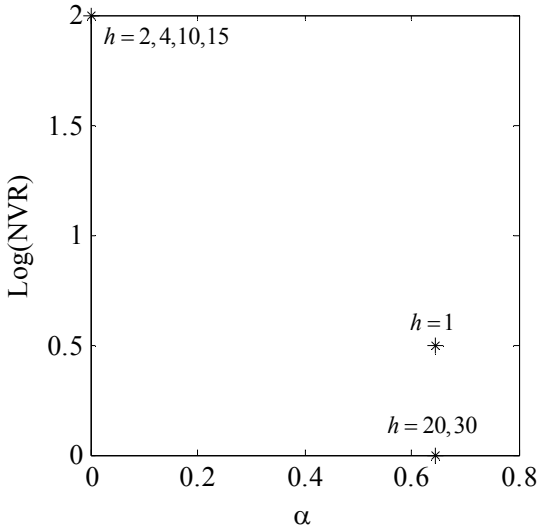

Figure 6.3: Series PSI20W: Values of the pair (α, NVR) of a DTR-IAR(1) model, by varying the time horizon.

After stochastic detrending, and according to Algorithm 3.1, we identified a RBFN, with time-varying parameters modelled by random walk processes, for the detrended series. The hyperparameters that were optimised for the network were:

— the 2-tuple (p, m), where p is the number of inputs, m is the number of units;

— the widths σ_i: we just set a common value $\sigma_i = \sigma$, computed using Equation (2.9), and optimised b, defined as the magnitude (in *log*-scale) of the widths ($s = 10^b \sigma$);

— the centres $\{\mathbf{c}_i\}$: these were identified by the *k*-means algorithm;

— the NVRs, set equal for all the time-varying parameters.

All the following alternatives were tested and compared:

$$p \in \{1,2,...,5\}; \quad m \in \{2,3,4,5,6,8,10,12,15,20,25,30\}$$

$$b \in \{g(x) = 10^x : x = -1,-0.5,...,1\}$$

$$NVR \in \{h(x) = 10^x : x = -\infty,-5,-4,...,-1\}$$

The best combination found for the 4-tuple (p, m, b, NVR) was $(1, 2, 1, 0)$, for which $RMSE(1) = 207.83$ in the test set. We notice that, the number of inputs is minimum (1), for the range of values considered. Extending the range of values for *p*, e.g., $p \in \{1, 2, .., 12\}$, the best combination for the given 4-tuple is now $(12, 12, -0.5, 10^{-2})$, for which $RMSE(1) = 238.44$ in the test set. Although the predictive performance can be improved in the training set, the predictive performance in the test set (generalisation accuracy) is poorer, due to the over-parameterisation of the model. The higher the number of inputs, the more dispersed the data becomes in the input space, leading to an increased number of units required in the network. Therefore, we decided to keep the original range of values ($1 \le p \le 5$) in the subsequent experiments.

We then proceeded to the estimation of the hybrid model DTR-RBF, optimising again the pair (α, NVR), but keeping the neural network structure fixed. The best combination obtained was $(0, 10^2)$, for which the final predictive performance measure in the testing set was $RMSE(1) = 204.11$. Figure 6.4 shows the 1-step-ahead predictive estimates (last 200 values) obtained from the hybrid model DTR-RBF and the corresponding prediction errors. In the series of the prediction errors, we can notice a range of correlated values, almost all with positive sign, indicating the presence of a signal not caught by the hybrid model.

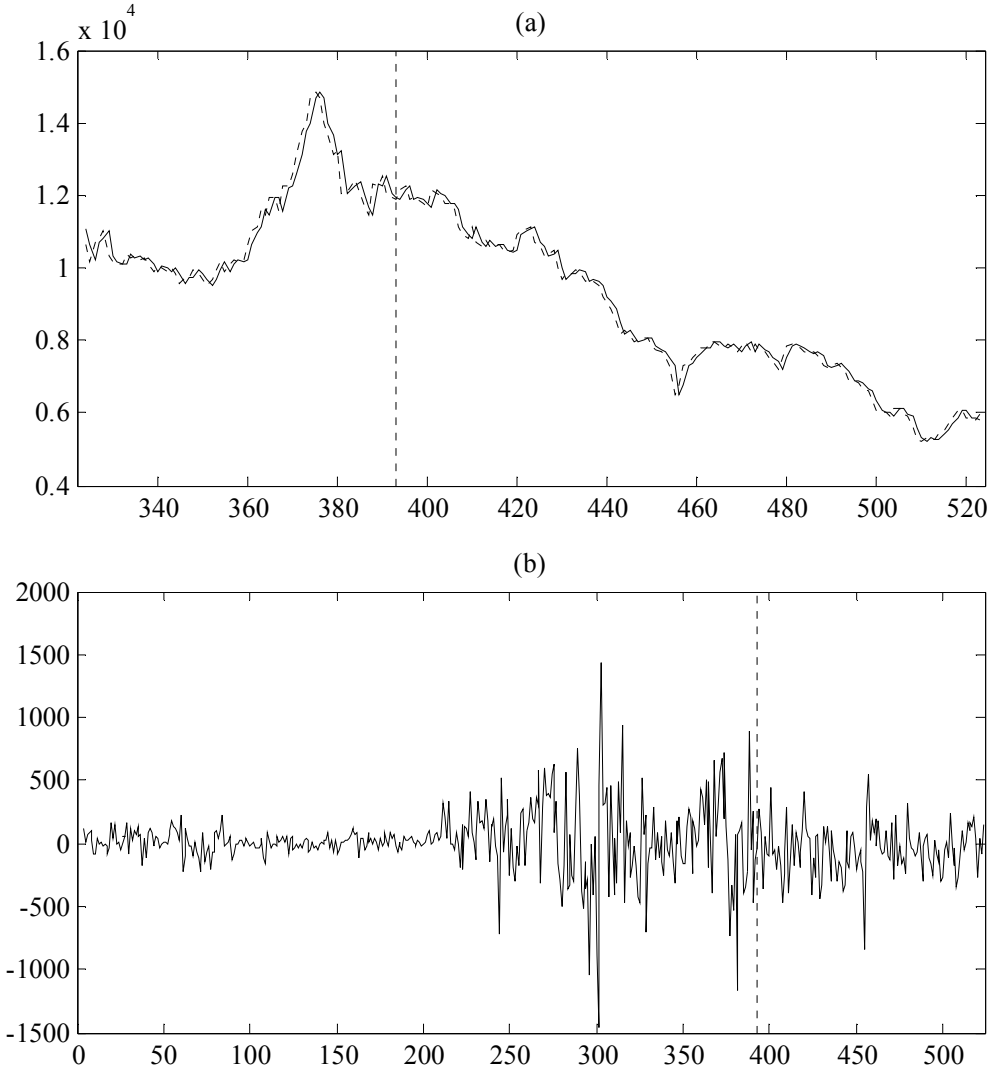

Figure 6.4: Series PSI20W: (a) Last 200 of the 1-step-ahead predictive estimates obtained from the hybrid model DTR-RBF (solid line); (b) the corresponding prediction errors.

Comparing the detrended series in Figure 6.2b and the prediction errors in Figure 6.4b, we conclude that the RBFN in the hybrid model could not deal with the volatility present in the data in a reasonable manner, hence conditioning the success of the overall procedure. In fact, if we optimise separately a DTR-IAR(1) with respect to RMSE(1), we note that a similar result is obtained, i.e., with a predictive performance measure $RMSE(1) = 208.59$ in the testing set.

Considering now a classical preprocessing approach, explicitly first-order differencing, and then a radial basis function network using the same conditions as before in the

identification task, we get a predictive performance $RMSE(1) = 225.48$ in the testing set, which is worse than for the other models, although it was the best one in the training set.

Additionally, for the sake of curiosity, we combined the fd;RBF and the DTR-RBF models by means of the extended linear combination method, with time-varying weights, modelled as random walk processes. All results for time series PSI20W are summarised in Table 6.1.

	RMSE(1)*	
	Training	Test
Naïve1	316.32	203.86
DTR-IAR(1)	316.73	208.59
(1) fd;RBF	310.23	225.48
DTR;RBF	317.55	207.83
(2) DTR-RBF	316.58	204.11
XLC of (1) and (2)	313.77	202.54

Table 6.1: Results for time series PSI20W.

None of the identified models was able to significantly improve the results from the Naïve1 method, at least, in terms of predictive accuracy, likely due to the apparent "quasi-random-walk" behaviour of PSI20W. Moreover, the combination approach was not able to lead to much better results, but a marginal improvement was achieved in the test set.

In another set of experiments, we considered the time series PSI20D, pertaining to daily close prices. PSI20D is less volatile than PSI20W, as already noticed in Figure 6.1, and thus one may expect to achieve better results in this case.

All the experiments were carried out by applying steps identical to those formerly used. The corresponding results are summarised in Table 6.2. In one case, we used the time series PSI20D without any preprocessing procedure. We also considered aggregated values: time series PSI20D1 and PSI20D2. PSI20D1 records the weekly averages of the values of PSI20D – the number of averaged values in each week is not always the same, due to the absence of observations in holidays. PSI20D2 was obtained by applying a moving average of length 5 to time series PSI20D, thus resulting in a smoother time series in comparison to PSI20D1. The main advantage of having aggregated values is that the resulting time series is less volatile than the original one. Nevertheless, some drawbacks may arise, namely the loss of some nonlinear correlations that might exist in the daily data.

	RMSE(1)*					
	Training			Test		
	PSI20D	PSI20D1	PSI20D2	PSI20D	PSI20D1	PSI20D2
Naïve1	47.72	97.66	22.88	65.43	150.51	36.32
DTR-IAR(1)	47.76	91.61	12.66	65.55	146.51	18.23
(1) fd;RBF	47.51	90.12	11.63	64.36	141.34	16.01
DTR;RBF	47.75	91.99	11.38	64.98	138.70	15.08
(2) DTR-RBF	47.92	91.02	13.07	65.50	142.33	18.11
XLC of (1) and (2)	47.19	92.75	11.49	64.28	145.27	16.13

Table 6.2: Results for time series PSI20D, PSI20D1 and PSI20D2.

From Table 6.2 we conclude that one can significantly improve the results of the Naïve1 method, only with the aggregated time series PSI20D1 and, most noticeably, PSI20D2.

We conclude that time series PSI20, pertaining to either weekly or daily data, is very difficult to be modelled and forecasted, as expected. We also considered aggregated values from the daily data, to attempt smoothing down the high volatility present there. The moving average preprocessing procedure favoured more the DTR model, rather than the RBFN, in the hybrid model.

6.2 Tourism Data

Tourism in Madeira is an important industry for the development of this Region. Throughout the years, the number of tourists in Madeira has been increasing, even with some fluctuations, depending on economic cycles. Although the totals of guests entered in Madeira (see Figure A.4) are not the only factor that contributes heavily for the local economy growth, it is very important to follow its evolution in time, so that any policies can be supported by more accurate technical decision-making tasks. The consequences of successive non-supported or suboptimal decisions can be very critical, usually leading to the loss of revenues, and putting at risk the stability and sustainability of many companies and families.

The combination paradigm is meant to be used in the context of decision-making. Two different predictive models are identified, in a suboptimal way, such that a basis for the combination of both predictions and decisions can be provided. The individual models are obtained from the model synthesis approach, since the time series considered contains complex characteristics, namely the long-run trend component and other low-frequency effects, as discussed in Chapter 3. As usual, the data were divided into two subsets – the training set (the first 227 values) and the test set (the last 75 values) – in order to assess out-of-sample predictive accuracy.

We start by noting that the time series (see Figure A.4) is seasonal, with a main periodicity at frequency 1/12 and harmonic sub-frequencies 1/6, 1/4 and 1/3. It is convenient to transform the series so to render the variance nearly stationary, and the *log*-transformed time series (LGUESTS) is shown in Figure 6.5.

We first considered a preliminary filtering of the data through a DHR model, with an IAR(1) process for the trend parameter and RW processes for the periodic components. Appropriate values for the hyperparameters, namely the pair (α, NVR_1) and NVR_2 for the given processes, were chosen such that low-frequency effects could be reasonably removed from the series.

For the second phase of the methodology, we considered two different types of models, one to deal with possible nonlinearities and another with linear correlations, namely, a Gaussian RBF model and an AR model of order *p*, respectively, both of order *p* which shall provide dissimilar features caught by the models, favouring, to some extent, the combination paradigm.

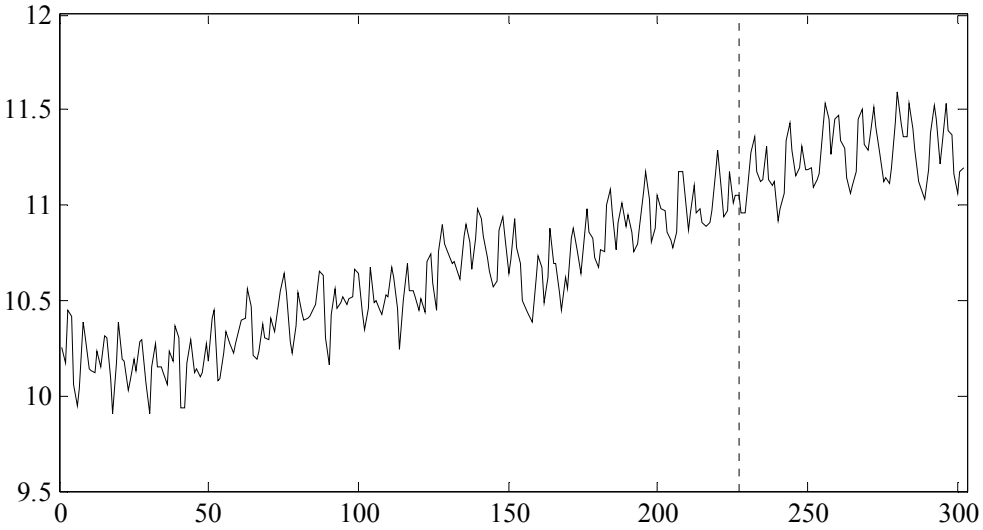

Figure 6.5: Series LGUESTS.

After investigating the periodogram of LGUESTS, we have decided to include in the DHR model two periodic components, namely at frequencies 1/12 and 1/6, allowing the others to be modelled by the subsequent models. These favour the identification of AR and RBF models, which should not be too complex. Noting that the most significant periodic component left in the data is the one at frequency 1/4, in both cases we considered an autoregression (linear and nonlinear, respectively) or order 4.

We noticed that the series LGUESTS grows approximately linearly, with some degree of damping. Setting $\alpha = 0.99$, we may have a reasonable approximation to the low-frequency effects, provided suitable choices are made for NVR_1 and NVR_2. In Figure 6.6, we compare the periodograms of the resulting detrended series and of series LGUESTS, with the settings $NVR_1 = NVR_2 = 0.01$. We obtain reasonable solutions, since the detrended series is nearly stationary, suitable to be modelled through the other models.

We then considered RBF and AR models, with 4 inputs, with all parameters constant in time. Then, the number of RBF units, *m*, was optimised, while the centres and the common width, σ, were identified through appropriate heuristic methods, namely the *k*-means clustering algorithm and Equation (2.9), respectively. Furthermore, we also considered a multiple value of the width, $s = 10^b \sigma$, optimised in a *log*-scale. The best results were attained with $m = 15$ and $b = -0.5$. Finally, hybrid models DHR-RBF and DHR-AR were created and estimated.

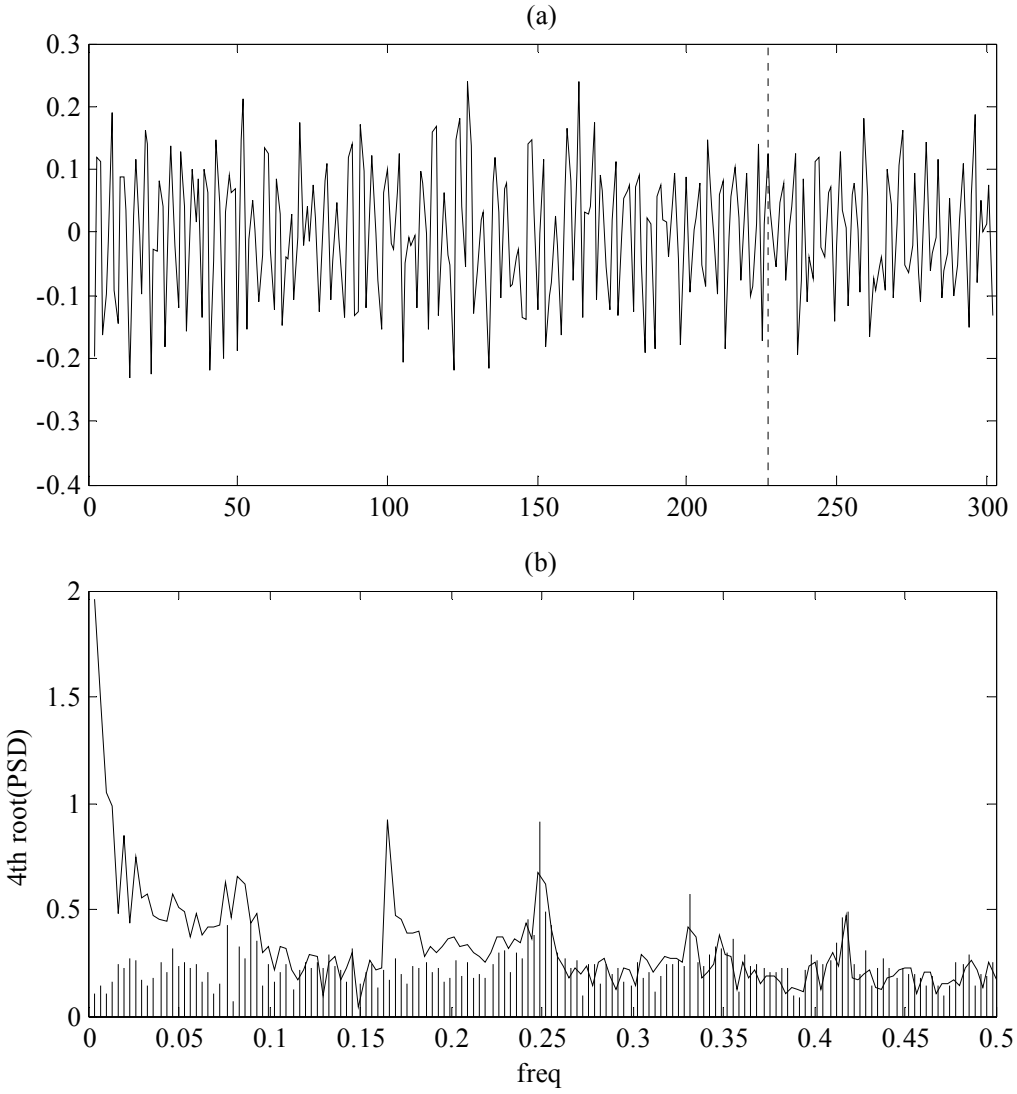

Figure 6.6: (a) Detrended series of LGUESTS through DHR detrending; (b) comparing periodograms of series LGUESTS and of the detrended series.

Our study of the series was continued by considering the combination paradigm, both for predictions and decisions. Specifically, we considered the extended linear combination introduced in Chapter 3, and the main comparative results are summarised in Table 6.3.

In the context of decision-making, we considered the asymmetric cost function LC(1), as defined in Equation (5.1):

$$LC(1) = \frac{1}{n - n_0} \sum_{k=n_0}^{n} d_{k|k-1}$$

where

$$d_{k|k-1} = \begin{cases} ue_{k|k-1}, & e_{k|k-1} \geq 0 \\ -ve_{k|k-1}, & e_{k|k-1} < 0 \end{cases}$$

is the cost associated to the decision errors made for time *k*. For this case study, we have set $u = 9$ and $v = 1$, giving more weight to positive errors than to negative ones. Accordingly, 1-step-ahead decisions were obtained from 0.9-quantiles, estimated empirically from past errors using sliding time windows consisting of the last 50 estimated forecast errors, as described in Section 5.3.

Table 6.3 shows the results obtained by the individual models and the combination of them concerning prediction and decisions problems, respectively. Figure 6.7 shows the errors associated to the decisions, the corresponding histograms, and respective decision costs obtained by each combining approach, CPred and CDec. The minimum average costs were: for approach CPred, $LC(1) = 0.122$; and, for approach CDec, $LC(1) = 0.156$.

	Prediction *RMSE*(1)*	Decision *LC*(1)*
DHR-RBF	0.082	0.123
DHR-AR(4)	0.089	0.135
Combination	0.074	0.156

Table 6.3: Series LGUESTS: Results of individual and combined models with respect to the prediction and decision optimisation problems (out-of-sample assessments).

In this case, CPred turned out to be the better approach, i.e., combining prescriptive solutions lead to worse results than to the individual ones, probably due to noise interference in the test set (see Figure 6.7, on bottom). As expected, most of the errors associated to the decisions are lesser than zero (see Figure 6.7, on centre), due to the degree of asymmetry among decision costs, i.e., it is preferable to have negative errors than positive ones.

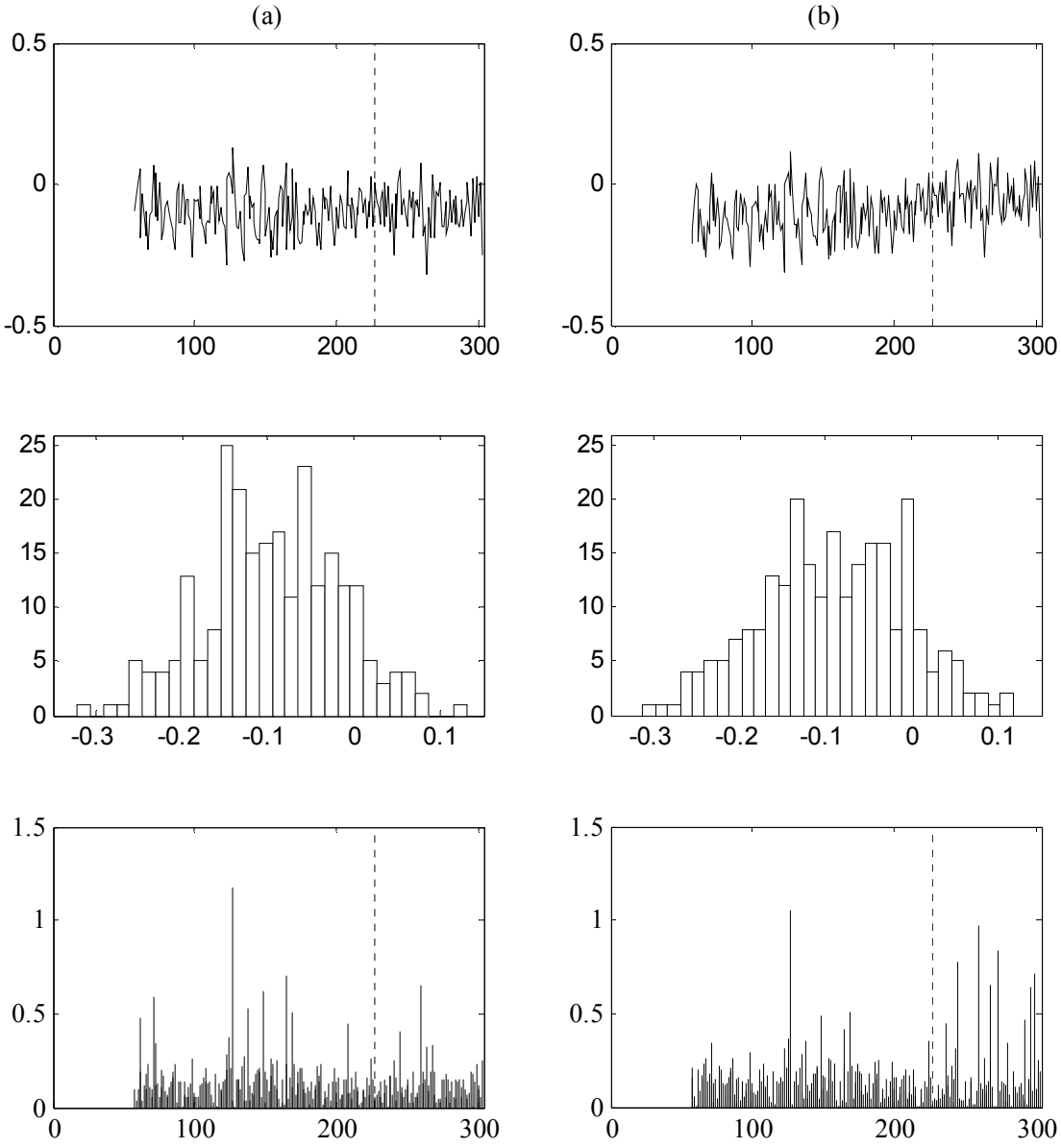

Figure 6.7: Series LGUESTS: Sequences of errors associated to decisions (top), the corresponding histograms (centre) and respective decision costs (bottom) from models: (a) DHR-RBF; (b) DHR-AR.

CHAPTER 7

Conclusions

Contents

Abstract

We summarise the main ideas explored in this thesis, and suggest possible ways of continuing the work initiated here.

7.1 A Review of the Main Results

Computational efficiency is of critical importance when dealing with large data sets, large collections of data, or with streamed data. Further difficulties come up when the data is noisy, nonstationary and nonlinear, requiring more complex and flexible yet robust forecasting models. Model optimality is very difficult to achieve but, through the optimal combination of suboptimal solutions, one may hope to efficiently obtain better quality solutions.

In this work, we studied different ways of combining predictive models or forecasts (in particular, neural predictive models or neural-based forecasts). In all cases, linear parametric models, possibly with time-varying parameters, were considered, in order to accomplish estimation problems recursively, i.e., with a single presentation of the sequence of observations, or of patterns to the model.

The main predictive models considered were Gaussian radial basis function networks, and we discussed several ways of "training" them, through recursive estimation and adaptive identification methods, all related to "prediction-error" updating schemes.

Among other possibilities, two particular combining approaches were explored – model mixing and model synthesis –, with the aim to improve predictive performance.

In the context of model mixing, we proposed an extension of the usual linear combination framework, able to cope with the case where the forecast errors from the different models – restricted to only two models – are significantly cross-correlated. This was accomplished with the inclusion of a nonlinear term, defined by a point-by-point multiplication of the sequences of individual predictive estimates. We derived generalisations of the analytical results known for the classic linear combination.

It was shown that the performance improvement is higher when the cross-correlation between the sequence of errors given by the classic linear formulation and the sequence of values defined by the supplementary nonlinear term is numerically large.

In the context of model synthesis, we proposed a preliminary filtering of the data by means of a dynamic regressive model – either dynamic trend regressive or dynamic harmonic regressive models –, before the identification of a neural network. We then defend the subsequent estimation of a complete hybrid model, including the dynamic regressive

model and the RBFN, which can be viewed as a particular case of the univariate Unobserved Components model.

Although more effective methods are already reported in the literature, we recommend considering appropriate choices for the values of the hyperparameters of the dynamic regressive model, based on the predictive accuracy for long time horizons, so to reduce the risk of having interference between the individual components – dynamic model and neural network. Furthermore, we advocate appropriate strategies to identify the complete hybrid model, due to the presence of many hyperparameters, inducing identifiability problems in a very high-dimensional space. We achieved reasonably good results by keeping the neural network structure found in the sequential estimation fixed, but optimising again the hyperparameters of the dynamic model, in order to facilitate the identification task.

In the computational experiments, we have shown that the classic differencing and the deterministic detrending approaches may be very inadequate. We concluded that the stochastic detrending methodology may be an appropriate default choice for nonstationary time series, especially when it is not clear whether the series is trend-stationary or difference-stationary.

Finally, we discussed the topic of model mixing in the context of optimal decision-making, based on forecasting models but under more realistic loss functions than the least-squares one. We described, and illustrated two approaches: combining different predictive estimates before inferring optimal decisions (CPred), or rather combining several decisions inferred from different forecasting models (CDec).

While predictive models can be more efficiently estimated through the least-squares criterion, in a recursive or adaptive manner, prescriptive models, which are mostly based on non-differentiable functions criteria, like the asymmetric linear function considered in this study, are much more difficult to be estimated through recursive or adaptive methods.

We have shown that the CPred and CDec methodologies are both sensitive to variations on the time horizon used for predictions and on the degree of asymmetry between decision costs, but none of the approaches dominates the other in performance terms. Nonetheless, the experiments suggest that, while the application of CPred is computationally more tractable, CDec should also be considered, for possible significant cost reductions.

7.2 Future Research

Research is an ongoing process to achieve iteratively the goals previously established, either by theory or through empirical evidence. During the course of this work, several methodological issues were identified as deserving further investigation.

Although the examples that were considered in the experiments showed the applicability of the proposed methods, one should carry more extensive experiments, with more data sets, to gather more substantial evidence of the advantages that can be gained.

Although the model mixing methodology was shown to be able to improve upon the classic combination framework, further research is needed so to make those potential improvements more significant in practical situations. It may be interesting to explore alternatives to the extended formulation, namely the inclusion of more complex terms, or other types.

In the experiments concerning the comparison of approaches CPred and CDec, it is somewhat uncertain how the time horizon and the degree of asymmetry of decisions costs contribute to distinguish their behaviour. Furthermore, since none of the approaches dominates the other, further research is essential in order to precise which factors, if any, may contribute to the success of either approach when compared to the other.

The consideration of the combination paradigm in the context of decision-making may be extended to combining density forecasts (see, e.g. [32]), rather than combining only point estimates, and our attention will also be directed in the future to this barely explored problem.

Other minor issues identified in the course of the work deserve further research, namely the extension of the application of the sequential *k*-means algorithm to all centres (including one inspired by particle swarm optimisation methods), and the application of adaptive weights to the extended linear combination, as the correlation between individual estimates obtained from distinct models is likely to change with time.

A P P E N D I X A

Data Sets

Contents

We list and plot the real time series used in our empirical studies in Chapters 3 and 6, including the sample spectrum representation (periodogram).

A.1 LGNP

Description: Log-transformed quarterly U.S. real gross national product, in billions.

Length: 227 (from 1947, 1st quarter, to 2003, 3rd quarter).

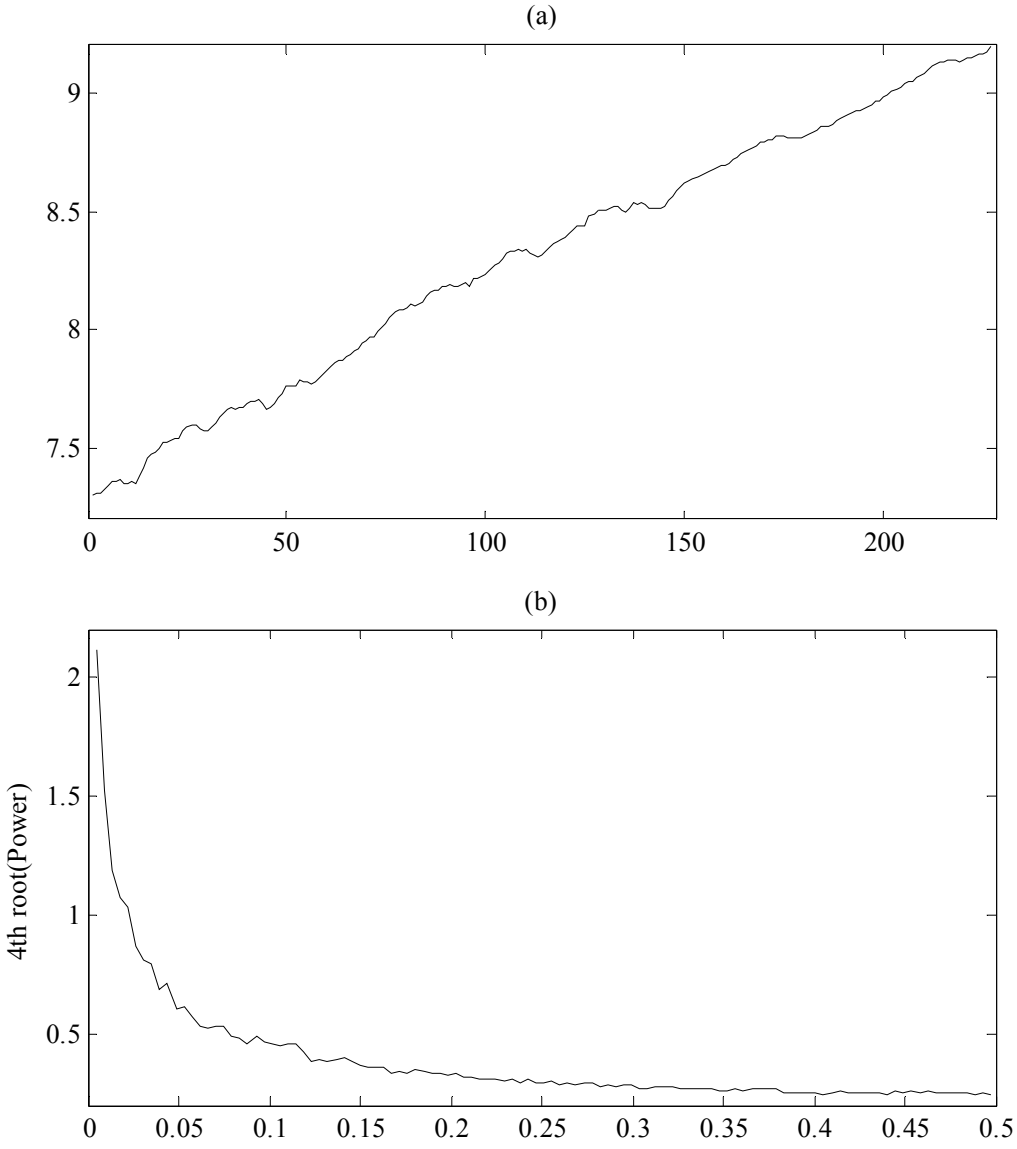

Figure A.1: (a) Time series LGNP; (b) Its periodogram.

A.2 PSI20W

Description: The Lisbon Stock Exchange Index PSI20 – week close prices.

Length: 523 (from December 31, 1992 (index = 3000), to January 03, 2003).

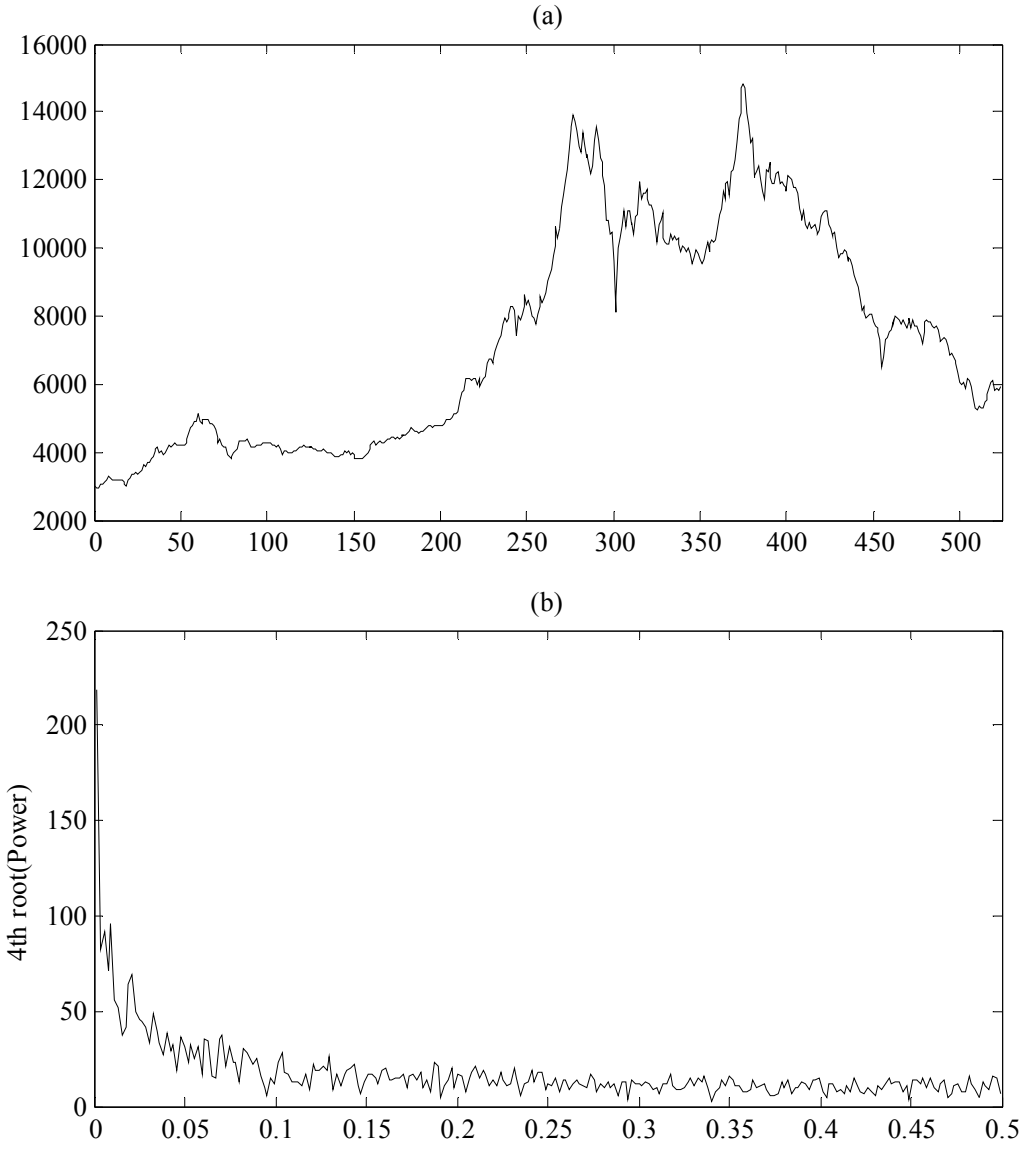

Figure A.2: (a) Time series PSI20W; (b) Its periodogram.

A.3 PSI20D

Description: The Lisbon Stock Exchange Index PSI20 – day close prices.

Length: 1136 (from January 06, 2003, to June 15, 2007).

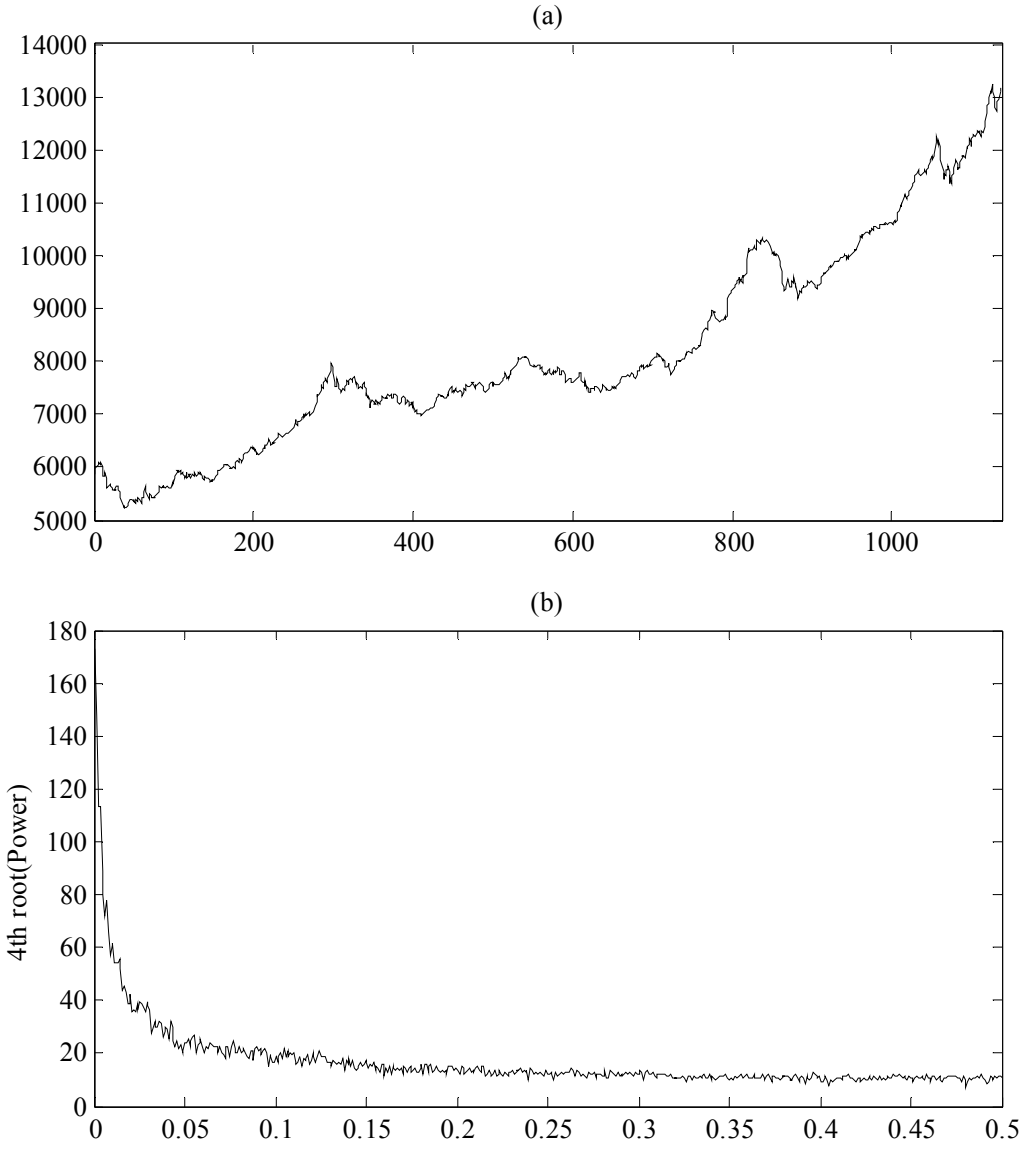

Figure A.3: (a) Time series PSI20D; (b) Its periodogram.

A.4 GUESTS

Description: Monthly totals of guests entered in Madeira, Portugal, i.e., all tourists (from Portugal Mainland or foreign countries) that check-in at any hotel.

Length: 302 (from January 1980, to February 2005).

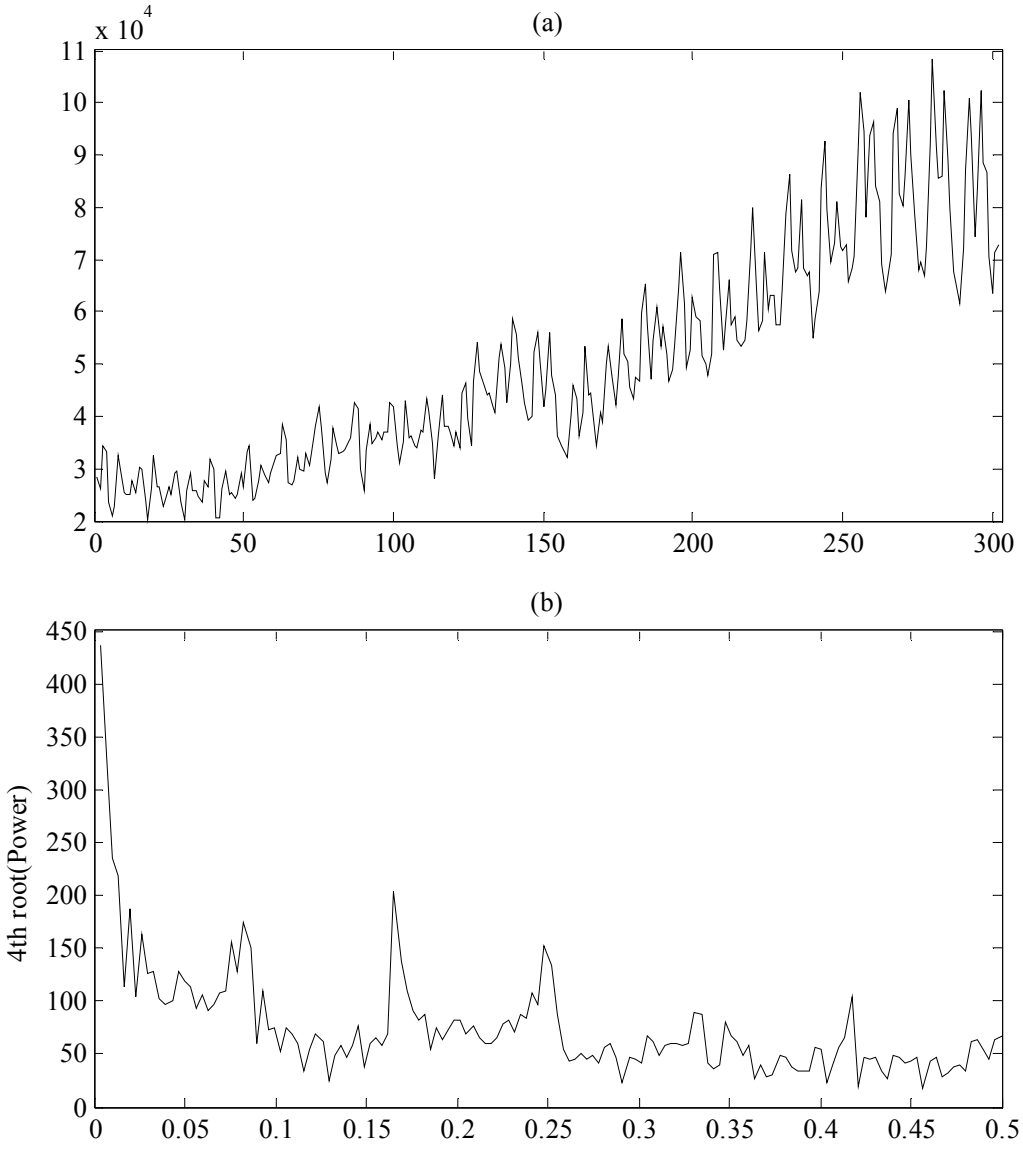

Figure A.4: (a) Time series GUESTS; (b) Its periodogram.

References

[1] Armstrong, J.S., *Combining forecasts*, in *Principles of Forecasting: A Handbook for Researchers and Practitioners*, J.S. Armstrong (Ed.). 2001, Kluwer Academic Publishers, Norwell, MA. pp. 417-439.

[2] Bagirov, A.M., A.M. Rubinov, N.V. Soukhoroukova and J. Yearwood, *Unsupervised and supervised data classification via nonsmooth and global optimization*. TOP: An Official Journal of the Spanish Society of Statistics and Operations Research, 2003. **11**(1): pp. 1-93.

[3] Bangzhu, Z. and L. Jian, *A novel feature extraction-based selective & nonlinear neural network ensemble model for economic forecasting*. International Journal of Computer Science and Network Security, 2007. **7**(2).

[4] Bates, J.M. and C.W.J. Granger, *The combination of forecasts*. Operational Research Quarterly, 1969. **20**(4): pp. 451-468.

[5] Bishop, C.M., *Neural Networks for Pattern Recognition*. 1995, Oxford University Press, Oxford.

[6] Bitmead, R.R., *Adaptive control algorithms*, in *Numerical Linear Algebra, Digital Signal Processing and Parallel Algorithms*, G.H. Golub and P. Van Dooren (Eds.). 1991, Springer-Verlag. pp. 19-40.

[7] Box, G.E.P., G.M. Jenkins and G.C. Reinsel, *Time Series Analysis: Forecasting and Control*. 3rd ed. 1994, Prentice Hall, Englewood Clifs, NJ.

[8] Broomhead, D.S. and D. Lowe, *Multivariable functional interpolation and adaptive networks*. Complex Systems, 1988. **2**: pp. 321-355.

[9] Campbell, S.L. and C.D. Meyer, *Generalized Inverses of Linear Transformations*. 1991, Dover Publications, New York.

[10] Carmo, J.L., *Redes de funções de base radiais: identificação e aplicação na previsão de séries irregularmente espaçadas*. 1997, M.Sc. thesis (in Portuguese), FCUL, University of Lisbon, Portugal.

[11] Carmo, J.L., *Previsão da procura e decisão optimal: modelos e métodos avançados*. 2007, Ph.D. thesis (in Portuguese), FCUL, University of Lisbon, Portugal.

[12] Carmo, J.L. and A.J.L. Rodrigues, *Identificação de redes neuronais Gaussianas como modelos de previsão* (in Portuguese). Investigação Operacional, 2002. **22**: pp. 43-57.

[13] Carmo, J.L. and A.J.L. Rodrigues, *Adaptive forecasting of irregular demand processes*. Engineering Applications of Artificial Intelligence, 2004. **17**(2): pp. 137-143.

[14] Chan, K.H., J.C. Hayya and J.K. Ord, *A note on trend removal methods: the case of polynomial regression versus variate differencing*. Enconometrica, 1977. **45**(3): pp. 737-744.

[15] Chatfield, C., *Time Series Forecasting*. 2001, Chapman & Hall, London.

[16] Chen, S., S.A. Billings and P.M. Grant, *Recursive hybrid algorithm for non-linear system identification using radial basis function networks*. International Journal of Control, 1992. **55**(5): pp. 1051-1070.

[17] Clemen, R.T., *Combining forecasts: a review and annotated bibliography*. International Journal of Forecasting, 1989. **5**(4): pp. 559-583.

[18] Clements, M.P. and D.F. Hendry, *Forecasting Economic Time Series*. 1998, Cambridge University Press.

[19] Crone, S.F., *Prediction of white noise time series using artificial neural networks and asymmetric cost functions*, in *Proceedings of the International Joint Conference on Neural Networks, IJCNN'03*, Portland, D. Wunsch, M. Hasselmo, K. Venayagamoorthy and D. Wang (Eds.). Vol. 4. 2003. pp. 2460-2465.

[20] De Gooijer, J.G. and R.J. Hyndman, *25 years of time series forecasting*. International Journal of Forecasting, 2006. **22**(3): pp. 443-473.

[21] Diebold, F.X. and J.A. Lopez, *Forecast evaluation and combination*, in *Handbook of Statistics*, G.S. Maddala and C.R. Rao (Eds.). Vol. 14. 1996, Elsevier Science, Amsterdam. pp. 241-268.

[22] Diebold, F.X. and A.S. Senhadji, *Deterministic vs. stochastic trend in U.S. GNP, yet again*. NBER Working Paper No. 5481. 1996, National Bureau of Economic Research.

[23] Donaldson, R.G. and M. Kamstra, *Forecast combining with neural networks*. Journal of Forecasting, 1996. **15**: pp. 49-61.

[24] Eberhart, R.C. and J. Kennedy, *A new optimizer using particle swarm theory*, in *Proceedings of the 6th International Symposium on Micro Machine and Human Science, MHS'95*, Nagoya, Japan. 1995. pp. 39-43.

[25] Fausett, L., *Fundamentals of Neural Networks: Architectures, Algorithms, and Applications*. 1994, Prentice Hall, Englewood Cliffs, NJ.

[26] Freitas, P.S.A., *Combinação de modelos neuronais na previsão de séries temporais*. 1999, M.Sc. thesis (in Portuguese), FCUL, University of Lisbon, Portugal.

[27] Freitas, P.S.A. and A.J.L. Rodrigues, *Model combination in neural-based forecasting*. European Journal of Operational Research, 2006. **173**(3): pp. 801-814.

[28] Golub, G.H. and C.F. Van Loan, *Matrix Computations*. 3rd ed. 1996, Johns Hopkins University Press.

[29] Granger, C.W.J., *Combining forecasts: twenty years later*. Journal of Forecasting, 1989. **8**(3): pp. 167-173.

[30] Granger, C.W.J. and R. Ramanathan, *Improved methods of combining forecasts*. Journal of Forecasting, 1984. **3**(2): pp. 197-204.

[31] Guay, A. and P. St-Amant, *Do the Hodrick-Prescott and Baxter-King filters provide a good approximation of business cycles?* Cahiers de Recherche/ Working Paper No. 53. 1997, CREFÉ/ CREFE, Université du Québec à Montreal.

[32] Hall, S. and J. Mitchell, *Density forecast combination*. International Journal of Forecasting, 2007. **23**: pp. 1-13.

[33] Han, D., W.-Q. Niu and C. Yu, *The comparative study on linear and non-linear forecast-combination methods based on neural networks*, in *International Conference on Wireless Communications, Networking and Mobile Computing*, Shanghai, China. 2007.

[34] Hansen, L.K. and P. Salamon, *Neural network ensembles*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1990. **12**(10): pp. 993-1001.

[35] Hart, J.D., *Differencing as an approximate de-trending device*. Stochastic Processes and Their Applications, 1989. **31**: pp. 251-259.

[36] Hashem, S., *Optimal linear combinations of neural networks*. Neural Networks, 1997. **10**(4): pp. 599-614.

[37] Haykin, S., *Neural Networks: A Comprehensive Foundation*. 2nd ed. 1999, Prentice Hall, Upper Saddle River, NJ.

[38] Kalman, R.E., *A new approach to linear filtering and prediction problems*. Transactions of the ASME: Journal of Basic Engineering, 1960. **82**: pp. 35-45.

[39] Kalman, R.E. and R.S. Bucy, *New results in linear filtering and prediction theory*. Transactions of the ASME: Journal of Basic Engineering, 1961. **83**: pp. 95-108.

[40] Khotanzad, A., H. Elragal and T.-L. Lu, *Combination of artificial neural-network forecasters for prediction of natural gas consumption*. IEEE Transactions on Neural Networks, 2000. **11**(2): pp. 464-473.

[41] Koenker, R. and S. Portnoy, *The Gaussian hare and the Laplacean tortoise: computability of squared-error vs absolute error estimators*. Statistical Science, 1997. **12**: pp. 279-300.

[42] Li, F. and G. Tkacz, *Evaluating linear and non-linear time-varying forecast-combination methods*. Working Paper No. 01-12. 2001, Bank of Canada, Ottawa.

[43] Light, W., *Ridge functions, sigmoidal functions and neural networks*, in *Approximation Theory*, E.W. Cheney, C.K. Chui and L.L. Schumaker (Eds.). Vol. 7. 1992, Academic Press, Boston. pp. 163-206.

[44] Lin, G.-F. and L.-H. Chen, *Time series forecasting by combining the radial basis function network and the self-organizing map*. Hydrological Processes, 2005. **19**(10): pp. 1925-1937.

[45] Ljung, L. and T. Söderström, *Theory and Practice of Recursive Identification*. 1983, The MIT Press, Cambridge, MA.

[46] Lloyd, S.P., *Least squares quantization in PCM*. IEEE Transactions on Information Theory, 1982. **28**(2): pp. 129-137.

[47] Lowe, D., *Adaptive radial basis function nonlinearities, and the problem of generalization*, in *Proceedings of the 1st IEEE International Conference on Artificial Neural Networks*, London. 1989. pp. 171-175.

[48] Makridakis, S. and R.L. Winkler, *Averages of forecasts: some empirical results*. Management Science, 1983. **29**(9): pp. 987-996.

[49] Moody, J. and C. Darken, *Fast learning in networks of locally-tuned processing units*. Neural Computation, 1989. **1**(2): pp. 281-294.

[50] Nelles, O., *Nonlinear System Identification*. 2000, Springer-Verlag, New York.

[51] Nelson, C.R. and H. Kang, *Spurious periodicity in inappropriately detrended time series*. Econometrica, 1981. **49**(3): pp. 741-751.

[52] Newbold, P. and C.W.J. Granger, *Experience with forecasting univariate time series and the combination of forecasts*. Journal of the Royal Statistical Society, Series A, 1974. **137**(2): pp. 131-165.

[53] Ng, C.N. and P.C. Young, *Recursive estimation and forecasting of nonstationary time series*. Journal of Forecasting, 1990. **9**(2): pp. 173-204.

[54] Orr, M.J., *Regularization in the selection of radial basis function centers*. Neural Computation, 1995. **7**(3): pp. 606-623.

[55] Ozun, A. and A. Cifter, *Nonlinear combination of financial forecast with genetic algorithm*. MPRA Paper No. 2488. 2007, Marmara University, Munich.

[56] Park, J. and I.W. Sandberg, *Universal approximation using radial basis function networks*. Neural Computation, 1991. **3**(2): pp. 246-257.

[57] Pedregal, D.J. and P.C. Young, *Some comments on the use and abuse of the Hodrick-Prescott filter*. Review on Economic Cycles, 2001. **2**(1): pp. 93-104.

[58] Perrone, M.P. and L.N. Cooper, *When networks disagree: ensemble methods for hybrid neural networks*, in *Neural Networks for Speech and Image Processing*, R.J. Mammone (Ed.). 1993, Chapman & Hall. pp. 126-142.

[59] Poggio, T. and F. Girosi, *A theory of networks for approximation and learning*. A.I. Memo No. 1140. 1989, MIT Artificial Intelligence Laboratory, Cambridge, MA.

[60] Pollock, D.S.G., *Recursive estimation in econometrics*. Computational Statistics & Data Analysis, 2003. **44**: pp. 37-75.

[61] Powell, M.J.D., *Radial basis function for multivariable interpolation: a review*, in *Algorithms for Approximation*, J.C. Mason and M.G. Cox (Eds.). 1987, Clarendon Press, Oxford. pp. 143-167.

[62] Reed, R.D. and R.J. Marks, *Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks*. 1999, The MIT Press, Cambridge, MA.

[63] Ripley, B.D., *Pattern Recognition and Neural Networks*. 1996, Cambridge University Press, Cambridge.

[64] Rodrigues, A.J.L., *Dynamic regression and supervised learning methods in time series modelling and forecasting*. 1996, Ph.D. thesis, Lancaster University, U.K.

[65] Silva, G.X.G., *Aprendizagem automática por redes neuronais de unidades locais*. 2006, Ph.D. thesis (in Portuguese), FCUL, University of Lisbon, Portugal.

[66] Silva, P.X.G., *Previsão de séries temporais não estacionárias por redes neuronais*. 1998, M.Sc. thesis (in Portuguese), FCUL, University of Lisbon, Portugal.

[67] Sutton, R.S. and A.G. Barto, *Reinforcement Learning: An Introduction*. 1998, The MIT Press, Cambridge, MA.

[68] Tan, S., J. Hao and J. Vandewalle, *Nonlinear systems identification using RBF neural networks*, in *Proceedings of the International Joint Conference on Neural Networks, IJCNN '93*, Nagoya. Vol. 2. 1993. pp. 1833-1836.

[69] Teixeira, J.C. and A.J.L. Rodrigues, *An applied study on recursive estimation methods, neural networks and forecasting*. European Journal of Operational Research, 1997. **101**(2): pp. 406-417.

[70] Trigg, D.W. and A.G. Leach, *Exponential smoothing with an adaptive response rate*. Operational Research Quarterly, 1967. **18**(1): pp. 53-59.

[71] Waterhouse, S.R. and G. Cook, *Ensemble methods for phoneme classification*, in *Advances in Neural Information Processing Systems*, M.C. Mozer, M.I. Jordan and T. Petsche (Eds.). Vol. 9. 1997, The MIT Press, Cambridge. pp. 800-806.

[72] Wettschereck, D. and T. Dietterich, *Improving the performance of radial basis function networks by learning center locations*, in *Advances in Neural Information*

Processing Systems, J.E. Moody, S.J. Hanson and R.P. Lippmann (Eds.). Vol. 4. 1992, Morgan Kaufmann Publishers, San Mateo, CA. pp. 1133-1140.

[73] Winkler, R.L. and S. Makridakis, *The combination of forecasts*. Journal of the Royal Statistical Society, Series A, 1983. **146**(2): pp. 150-157.

[74] Xu, L., A. Krzyzak and E. Oja, *Rival penalized competitive learning for clustering analysis, RBF net, and curve detection*. IEEE Transactions on Neural Networks, 1993. **4**(4): pp. 636-649.

[75] Yang, Y., *Regression with multiple candidate models: selecting or mixing?* Statistica Sinica, 2003. **13**(3): pp. 783-809.

[76] Yang, Y., *Combining forecasting procedures: some theoretical results*. Econometric Theory, 2004. **20**(1): pp. 176-222.

[77] Young, P.C., *Recursive Estimation and Time Series Analysis: An Introduction*. 1984, Springer-Verlag, New York.

[78] Young, P.C., D.J. Pedregal and W. Tych, *Dynamic harmonic regression*. Journal of Forecasting, 1999. **18**: pp. 369-394.

[79] Zhang, G., B.E. Patuwo and M.Y. Hu, *Forecasting with artificial neural networks: the state of the art*. International Journal of Forecasting, 1998. **14**(1): pp. 35-62.

[80] Zhang, G.P. and V.L. Berardi, *Combining multiple neural networks for time series forecasting*, in *Proceedings of the Decision Science Institute Annual Meeting*. 2000. pp. 966-968.

[81] Zou, H. and Y. Yang, *Combining time series models for forecasting*. International Journal of Forecasting, 2004. **20**(1): pp. 69-84.