Available online at www.sciencedirect.com
SCIENCE DIRECT®
European Journal of Operational Research 173 (2006) 801–814
EUROPEAN JOURNAL OF OPERATIONAL RESEARCH
www.elsevier.com/locate/ejor

Model combination in neural-based forecasting

Paulo S.A. Freitas [a,c,*], António J.L. Rodrigues [b,c]

[a] Departamento de Matemática e Engenharias da Universidade da Madeira, 9000-390 Funchal, Portugal
[b] Faculdade de Ciências da Universidade de Lisboa, 1749-016 Lisboa, Portugal
[c] Centro de Investigação Operacional, FCUL, Portugal

Received 31 October 2004; accepted 14 June 2005
Available online 15 November 2005

Abstract

This paper discusses different ways of combining neural predictive models or neural-based forecasts. The proposed approaches consider Gaussian radial basis function networks, which can be efficiently identified and estimated through recursive/adaptive methods. The usual framework for linearly combining estimates from different models is extended, to cope with the case where the forecasting errors from those models are correlated. A prefiltering methodology is proposed, addressing the problems raised by heavily nonstationary time series. Moreover, the paper discusses two approaches for decision-making from forecasting models: either inferring decisions from combined predictive estimates, or combining prescriptive solutions derived from different forecasting models.

Keywords: Forecasting; Neural networks; Model combination; Adaptive methods; Optimal decision-making

1. Introduction

Time series forecasting is a common goal in data mining applications, where most often the recorded data is indexed in time, and the variables or attributes have distributional properties and correlation effects that are nonstationary in time. Without sacrificing predictive accuracy, the models to be used should be not 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 classic point of view of identifying a single, "clearly best" model, which might require a high computational burden for its identification and optimization.

Most references in the literature on neural-based forecasting follow that traditional paradigm, usually referring to the application of

* Corresponding author. Address: Departamento de Matemática e Engenharias da Universidade da Madeira, 9000-390 Funchal, Portugal.
E-mail addresses: paulo@uma.pt (P.S.A. Freitas), ajrodrigues@fc.ul.pt (A.J.L. Rodrigues).

0377-2217/$ - see front matter
doi:10.1016/j.ejor.2005.06.057

multilayer perceptrons (see [28] for a review). These are highly nonlinear models requiring optimization in a high-dimensional space of a nonlinear least-squares cost function, inevitably with many local optima (see [18] for a comprehensive introduction to the optimization of neural networks). Parameter updating in those models, given new data, is cumbersome and, most importantly, their direct application to nonstationary data is inadequate (see [23,24]).

This paper seeks to discuss alternative approaches where, while still using supervised neural models, one can achieve good, if not better forecasting performance, through efficient recursive estimation and adaptive identification methods. All the parametric predictive models here proposed—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 optimization of a single neural model in terms of predictive accuracy, efficiency and robustness.

In most of the literature about time series forecasting and neural supervised learning, a classic paradigm is followed, where all effort is 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 parameterization. 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 some choices and assumptions have to be made such that the problem under study can be acceptably modelled and the underlying optimization problem solved. During this process, there are some issues that might be hard to sort out, such as choosing appropriate model selection criteria. In particular, if the chosen model is too complex or overparameterized, it can learn the noise intrinsic to the data, thus causing poor generalization performance, i.e., producing poor results when applied to new data (see, e.g., [3]).

As alternatives to the classic paradigm, several approaches have been proposed, where multiple models are explored and combined. This tends to minimize the implicit risk in taking into account just one model, even it is optimized, and that is limited in its capabilities with respect to the characteristics of the data and to the problem itself.

One may identify three fundamental ways of combining models, to yield a final estimate:

- *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; and,
- *model switching*, where the chosen estimate is selected from the estimates of different models.

The main goal of the present work is to propose some guidelines for combining neural models or neural-based forecasts. While most of the optimization problems reported here are related to time series forecasting, some may be adapted to other problems, such as classification or clustering. The main models used in the paper are Gaussian radial basis function networks, a type of supervised neural networks that may be conveniently used as filtering models, and not just as strictly regressive models (Section 2). We propose some new ideas of using the model mixing approach (Section 3), provided the data is reasonably stationary, or the model synthesis approach (Section 4), when the data is clearly nonstationary. As prediction is just a means for supporting decision-making, in Section 5 we discuss whether one should combine the predictive models or the prescriptive ones.

2. Forecasting using radial basis function networks

2.1. Model specification

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:

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

The network outputs are viewed as predicted estimates, in particular, one-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.

The one-step-ahead forecasting errors are defined as

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

and are used to optimize the model, for a given particular performance measure. In our study we will use the root mean squared error (RMSE) measure based on in-sample forecasting errors.

The main neural models proposed in this study are single-output Gaussian radial basis function (RBF) networks. The extension to networks with multiple outputs is relatively straightforward, with the different outputs representing, in a predictive set-up, predictive estimates for different horizons.

A single-output RBF network is a linear combination of the outputs produced by a number of radially symmetrical activation functions, which are nonlinear in the inputs:

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

where $\{\boldsymbol{\theta}_k\}_{k=1,2,\ldots}$ are vector time-varying linear parameters, each $\mathbf{u}_k = [\,1 \quad \phi_{1k} \quad \cdots \quad \phi_{mk}\,]^{\mathrm{T}}$ is a regressor vector, and the sequence $\{\varepsilon_k\}$ is assumed to be a white noise process. Usually, Gaussian basis functions are considered:

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

$$1 \leqslant i \leqslant m.$$

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 [5,6]—as it is unpractical to approach it as a nonlinear optimization problem in a very high-dimensional space. In particular, it is common and adequate to choose the location of the centres through the k-means clustering algorithm. However, instead of directly clustering the input patterns, we apply the clustering procedure in the complete input–output space, and then project the cluster centres into the input space. Bagirov et al. [1] have approached the clustering problem via nonsmooth and global optimization methods.

The identification and estimation of a RBF network can be accomplished in two learning stages: firstly, the nonlinear hyperparameters (centres and widths) are identified through an unsupervised learning heuristic method; secondly, the linear parameters are estimated in a supervised manner. The structure of the network, namely the number of inputs, p, defining the order of regression of the model, and the number of radial basis units, m, have also to be defined early in the identification stage.

2.2. Online estimation

In real-world optimization problems, where huge collections of data are available and continuously observed, we shall consider recursive estimation procedures so that the model parameters can be estimated in a more efficient way, even in the limit case they are assumed 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 y_k, the general updating scheme for a forward processing recursive estimation procedure is

$$\hat{\boldsymbol{\theta}}_k := f(y_k, \hat{\boldsymbol{\theta}}_{k-1}; \psi),$$

where ψ is the set of model hyperparameters, i.e., any special parameters in the model or in the learning algorithm whose values condition the actual estimation.

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

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

seeking to minimize a least-squares type of loss function.

The above scheme encompasses several well-known algorithms, including the recursive least squares (RLS) and its variants, particularly the "dynamic RLS", or covariance addition method (RLS-CA), which in fact is a particular case of the Kalman filter (KF) (see, e.g., [4,16,22,27]).

In RLS, the gains, $\mathbf{k}_k$, are computed from auxiliary square matrices, $\mathbf{P}_k$, also recursively estimated:

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{k}_k \mathbf{u}_k^{\mathrm{T}})\mathbf{P}_{k-1}.$$

Assuming the unknown parameters are constant in time, i.e., $y_k = \boldsymbol{\theta}^{\mathrm{T}}\mathbf{u}_k + \varepsilon_k$, the RLS algorithm efficiently produces a sequence of estimates of the parameter vector which asymptotically converges to the ordinary least squares estimates. As much as the well-known error backpropagation algorithm—commonly used for training multilayer perceptrons—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 [23].

In RLS-CA, one assumes that the unknown parameters follow a vector random walk (RW) model,

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

so the updating equation for $\mathbf{P}_k$ becomes:

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{k}_k \mathbf{u}_k^{\mathrm{T}})\mathbf{P}_{k-1} + \mathbf{Q}.$$

Another simple approach for coping with time-varying parameters is the exponential forgetting version of RLS, where

$$\mathbf{P}_k = \left(\mathbf{I} - \mathbf{k}_k \mathbf{u}_k^{\mathrm{T}}\right)\mathbf{P}_{k-1}/\lambda, \quad 0 < \lambda < 1.$$

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 environments, it may be preferable to prefilter the data, as described in Section 4.

2.3. Online identification

So far, we have considered the centres and widths of the Gaussian RBF networks to be fixed during the estimation process. However, 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

$$\begin{aligned}\mathbf{c}_{i,k+1} &= \mathbf{c}_{i,k} + \alpha_{i,k}(\mathbf{x}_{k+1} - \mathbf{c}_{i,k}) \\ &= \alpha_{i,k}\mathbf{x}_{k+1} + (1 - \alpha_{i,k})\mathbf{c}_{i,k},\end{aligned}$$

where $\alpha_{i,k}$ (usually defined in the interval $(0, 1)$) is the learning rate, responsible for the speed and accuracy of the adaptation.

There are several versions of this scheme and suggestions for the choice of the values $\alpha_{i,k}$. One of the simplest is the *sequential* k-means clustering, where only the centre nearest to $\mathbf{x}_{k+1}$ is adapted. Some proposals for the value of the learning rate were compared in [5].

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 that one unit or even to all units. One possible approach, we are currently investigating, is to adapt all centres in a way inspired by particle swarm optimization [9], drawing from the analogy between RBF centres and the particles in a 'swarm'.

3. The combination of estimates

Probably, the most common approach to model combination is model mixing, where one combines the estimates produced by the individual models through weights. The first studies go back to Bates and Granger [2], and possibly others, who considered the linear combination of two different forecasting models. This approach was later extended to more models in [11,20,26]. Since then, many contributions have emerged on the linear combination of supervised neural models, including [15,21] for regression problems; [13,25] for classification problems; and particularly [29] in forecasting problems.

The combination of estimates requires the definition of a model to perform such combination—a model which should then be optimized with respect to a predefined criterion. Most combination methods proposed in the literature are based on linear parametric models, where the parameters are viewed as weights in the combination.

3.1. Constant weights

Naturally, there are some similarities, to some extent, among the estimates produced by different models. It is common to find a stronger disagreement in small sized errors from different models than in larger ones. This is more obvious in the case of outliers being present in the data. Hashem [15] has alerted for the possible collinearity that might exist among different sequences of estimates and that can undermine the robustness—and, therefore, the generalization ability—of the combined model.

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 to improve the performance of that combination.

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

$$\hat{y}^{(c)}_k = w_0 + w_1\hat{y}^{(1)}_k + w_2\hat{y}^{(2)}_k + \pi\hat{y}^{(1)}_k\hat{y}^{(2)}_k.$$

The nonlinear term is included in order to take into account the possible correlation between the two sequences of corresponding errors. Ideally, each sequence of errors should have white noise properties. Other authors have considered the simpler formulation where $\pi = 0$, in some cases together with additional constraints, such as $w_0 = 0$ and $w_1 + w_2 = 1$.

The values for the weights $\mathbf{w} = [\, w_0 \quad w_1 \quad w_2 \,]^{\mathrm{T}}$ and π can be found by minimization of the sum of squared errors (as usual, a few ones should be ignored in the beginning of the recursive estimation process):

$$\mathrm{SSE} = \sum_{k=n}^{N} \left(e^{(c)}_k\right)^2.$$

In Appendix A we derive formulae for the optima: $\mathbf{w}^*$, π^* and SSE^*. These are generalizations of the known solutions for the restricted formulation where $\pi = 0$. Denoting by SSE^*_L the minimum SSE of the latter, it is easy to show that

$$\mathrm{SSE}^* \leqslant \mathrm{SSE}^*_L \leqslant \min\{\mathrm{SSE}^{(1)*}, \mathrm{SSE}^{(2)*}\},$$

where $\mathrm{SSE}^{(1)*}$ and $\mathrm{SSE}^{(2)*}$ are the minima associated to the individual models. 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.

3.2. Experimental results

We briefly illustrate the application of either approach for combining two suboptimal RBF networks—the classic one ($\pi = 0$) and the extended one (unconstrained π).

For this experiment, and for the experiments described in Section 4.2, we have simulated nine time series, through the following Gaussian RBF model, corrupted with additive white noise:

$$x_k = \theta_{0,k} + \sum_{i=1}^{5} \theta_{i,k} \exp(-0.5\|\mathbf{i}_k - \mathbf{c}_i\|^2/\sigma_i^2) + \varepsilon_k,$$

where

$$\begin{aligned}
&\mathbf{i}_k = (x_{k-2}, x_{k-1})^{\mathrm{T}},\\
&\mathbf{c}_i \sim \mathbf{U}(-a, a)^2,\ \sigma_i = \sigma,\ \forall i,\\
&\theta_{i,k} = \theta_{i,k-1} + \eta_{i,k},\ \eta_{i,k} \sim N\left(0, \sigma_\eta^2\right),\ \varepsilon_k \sim N\left(0, \sigma_\varepsilon^2\right).
\end{aligned}$$

This means that the parameters are described by distinct (and uncorrelated) random walk processes, with a common noise variance ratio, $\mathrm{NVR} = \sigma_\eta^2/\sigma_\varepsilon^2$. To generate the different sequences of data—named $A, B, \ldots, H, I$—we have assigned different combinations of values for the 4-tuple $(a, \log_{10}\sigma, \sigma_\varepsilon^2, \log_{10}\mathrm{NVR})$, as defined in Table 1, as well as different locations for the clusters.

In Fig. 1 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

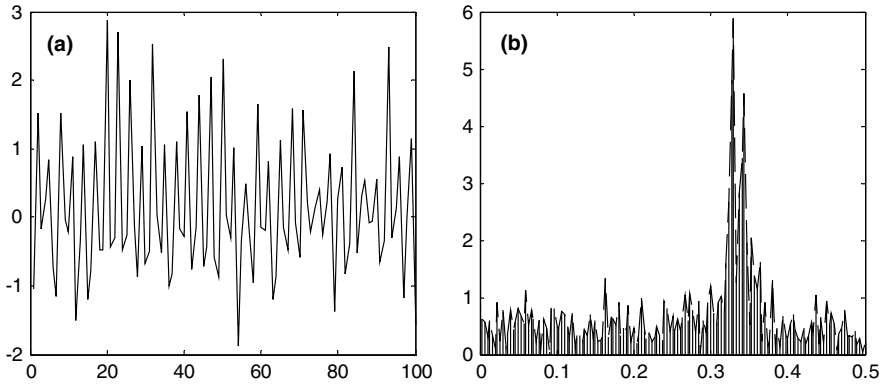

Fig. 1. (a) First 100 values of time series A; (b) periodogram of A.

Table 1
Simulation settings for 9 time series and best values attained by pseudo-best model M0

	A	B	C	D	E	F	G	H	I
a	1	2	3	3	2	3	1	2	2
$\log_{10}\sigma$	−0.25	0.25	0.25	0.0	0.0	0.25	0.0	−0.25	−0.25
σ_ε^2	0.1	1.0	1.0	1.0	0.5	1.0	0.1	0.5	0.5
$\log_{10}$ NVR	−3	−3	−3	−3	−3	−2	−3	−2	−3
p	2	2	2	2	2	2	2	2	6
m	12	4	4	5	8	5	8	12	8
c	−0.25	0.25	0.5	0.25	0.25	0.25	0.25	−0.25	0.0
b	−3	−3	−3	−3	−4	−2	−4	−2	−6
RMSE	0.33	1.05	1.11	1.02	0.69	1.13	0.30	0.78	0.71

nonlinearity. However, the difficulty in predicting time series like this one is due much more to the presence of noise in the parameters and in the observations than to the nonlinear autocorrelations.

Gaussian RBF networks 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-CA algorithm, and with centres based on k-means clustering, was found with respect to the RMSE measure based on in-sample one-step-ahead prediction errors, where the 4-tuple (p, m, c, b) was optimized:

- p: number of inputs;
- m: number of RBF units;
- $c = \log_{10}\sigma$ (all units with equal widths, optimized in a logarithmic scale);
- $b = \log_{10}\text{NVR}$ (parameters with equal noise variance ratios, optimized in a log scale).

In Table 1 we show the best values attained for each time series.

Keeping most of the optimized values fixed, we have estimated, for each time series, four more RBF models (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 2, we compare the relative performance of the five models, as well as the relative performance of the two formulations for the linear combination (in either case, we left w_0, w_1, w_2 unrestricted). Namely, since the σ_ε value of the true model can be seen as a lower bound for the RMSE predictive performance, we report the average, over the 9 time series, of the index

$$J(1) = 100 \times (\text{RMSE}(1) - \sigma_\varepsilon)/\sigma_\varepsilon.$$

Table 2
Average performance of some individual models and their combinations over 9 series

Model	Avg. perf. index $\bar{J}(1)$ (%)
M0: clustering-based centres	4.4
MR1: random centres	13.8
MR2: random centres	28.5
MR1, MR2 combined with $\pi = 0$	11.3
MR1, MR2 combined with unconstrained π	10.7
MS1: smaller widths	18.3
MS2: larger widths	9.6
MS1, MS2 combined with $\pi = 0$	5.3
MS1, MS2 combined with unconstrained π	4.7

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 in [10].

3.3. Adaptive weights

As a possible way of partially coping with nonstationarity in the data, or simply in order to avoid optimizing the weights in the combination of estimates, those weights may be made adaptive, and estimated recursively. Since the combining method is based on a linear parametric model estimated with respect to the least squares criterion, weight adaptation can be achieved through recursive expressions similar to those of the RLS algorithm.

Here, we just briefly describe adaptive schemes for the following simple case ($w_0 = \pi = 0$):

$$y_k^{(c)} = w_{1,k} y_k^{(1)} + w_{2,k} y_k^{(2)}.$$

The adaptation of each weight can be based on exponential forgetting:

$$w_{i,k+1} = \alpha f\left(e_1^{(i)}, \ldots, e_k^{(i)}\right) + (1-\alpha) w_{i,k},$$

where f is a function of past prediction errors, and α is a hyperparameter, chosen in the $(0,1)$ interval. In special, one may define

$$f\left(e_1^{(i)}, \ldots, e_k^{(i)}\right) = \frac{S_{v,k}^{(i)}(k)}{S_{v,k}^{(1)}(k) + S_{v,k}^{(2)}(k)},$$

where $S_{v,\lambda}^{(i)}$ is the inverse of a weighted covariance error matrix,

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

Particular cases can be considered, namely $\lambda = 1$ or $v = k$ [26]—the first one defines a sliding time window formed by the last v prediction errors and the second one is based on exponential weighting of past prediction errors. The hyperparameters v or λ might be optimized, 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.

4. Nonstationary time series

4.1. Introduction

In the context of supervised learning, we define extrapolation as generalization when the input pattern is outside the convex hull defined by the previously seen input patterns. Naturally, extrapolation is regarded as riskier than interpolation. If we seek to forecast a nonstationary time series by using a neural model, in autoregressive fashion, it is very convenient to find means of adequately preprocessing the data so to render the training patterns relatively stationary, and therefore reducing the likelihood of extrapolation occurring [24].

Common approaches in practice for achieving stationarity in the mean of a time series include differencing and deterministic detrending, but either approach has drawbacks. First-order differencing and linear regression detrending (or, correspondingly, higher-order extensions of these) are likely to produce different types of distortion in the spectral characteristics of the resulting data

(see [7,14]). 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 [19]. This induces some form of pseudo-periodic behaviour, not present in the original data.

We defend that the above techniques should be replaced by more elaborate alternative ones, such as stochastic detrending (or prefiltering). Specifically, to facilitate the analysis and modelling of the high-frequency effects in the data, we may consider the preliminary estimation and removal of a dynamic stochastic model, to account for the trend or other low-frequency effects [23]. Two popular methods aimed at removing low frequency variation from a time series are the Hodrick–Prescott and the Baxter–King filters but they have been the subject of several criticisms [12]. For simplicity, in this study we will make use of the dynamic trend regressive model (DTR) with integrated first-order autoregressive parameters (IAR(1)), estimated through the Kalman filter:

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

This model includes the particular cases of random walk (RW) and integrated random walk (IRW) level parameters—$\alpha = 0$ and $\alpha = 1$, respectively. The risk of producing distortions in the residuals spectrum is then much attenuated or negligible provided one considers suitable values for the hyperparameters, α and $\text{NVR} = \sigma_\eta^2/\sigma_\varepsilon^2$.

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 the higher-frequency, nonlinear autocorrelations. The two models can then be put together in the form of a complete model, to be estimated simultaneously, and tested.

4.2. Experimental results

For the following experiment, we considered 18 simulated nonstationary time series, built from the 9 time series introduced in Section 3.2 (*A* to *I*):

- 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 Fig. 2; similarly for *B* to *I*).

First-order differencing would be suitable for DS but not for TS, while deterministic detrending would be suitable for TS but not for DS. Our main goal in the experiment is to empirically evaluate the opportunity loss cost in failing to use the correct preprocessing approach, and assess, for both time series, the predictive performance of a model composed of a dynamic trend (DTR-IAR(1)) and a Gaussian RBF network, with random walk parameters (GRBFN-RW), as a general approach.

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

1. GRBFN (constant parameter model);
2. GRBFN-RW;
3. DTR-IAR(1);

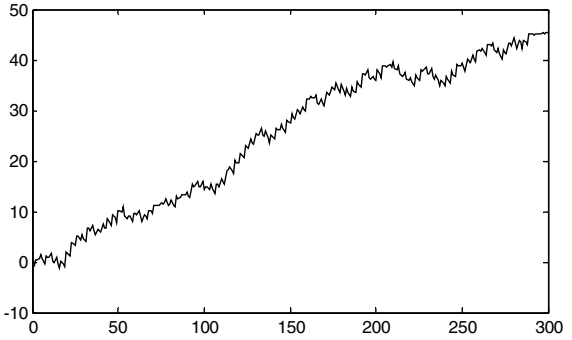

Fig. 2. Integration of time series *A* (DS).

Table 3
Comparative predictive performance, 1 and 4 steps ahead, of seven different modelling approaches, for different types of nonstationarity (TS, DS, or unknown)

Model	Method	Simulated data (*A–I*)				Real data (LGNP)	
		TS: $\bar{J}(1)$	DS: $\bar{J}(1)$	TS: $\bar{J}(4)$	DS: $\bar{J}(4)$	100* RMSE(1)	100* RMSE(4)
GRBFN	RLS	101.4	1255	116.8	1914	2.40	4.22
GRBFN-RW	RLS_CA	56.4	116.3	73.5	243.2	2.03	4.18
DTR-IAR(1)	KF	63.4	60.8	65.3	171.1	1.40	4.12
tr;RBF	RLS_CA	*4.1*	77.1	*45.6*	170.3	*0.87*	*2.38*
fd;RBF	RLS_CA	40.7	*4.3*	101.2	*129.5*	*0.86*	*2.35*
DTR;RBF	KF + RLS_CA	26.4	32.7	51.8	141.4	0.90	2.59
DTR-RBF	KF	15.2	33.3	49.9	145.0	1.23	2.64

4. tr;RBF (GRBFN-RW modelling, after deterministic trend removal);
5. fd;RBF (GRBFN-RW modelling, after first differencing);
6. DTR;RBF (GRBFN-RW modelling, after DTR-IAR(1) prefiltering);
7. 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 US real gross national product, 1947Q1-2003Q3. This data set, or its annual version, has been the most studied supporting the discussion of the trend-stationarity and difference-stationarity approaches to macroeconomic time series identification [8].

We have optimized the 4-tuple $(p, m, \sigma, \text{NVR})$ in all RBF models under study, with the exception of the first approach, where there is no NVR to be identified. In the approaches DTR;RBF and DTR-RBF, we have prefiltered the data using the default values $\alpha = 1$ and $\text{NVR} = 10^{-3}$. Then, in the simultaneous estimation approach, we have optimized also these hyperparameters. The main results obtained are summarized in Table 3.

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. 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 RBF network can be used as a general approach for dealing with nonstationary time series with nonlinear autocorrelations.

5. Combining decisions vs. combining predictions

5.1. Asymmetrical costs

Prediction is just a means—albeit a very important one—of supporting decision-making. In this section, we wish to discuss two possible combining approaches in the context of optimal decision-making:

- (CPred) first, combine (using optimal weights) the predictive estimates produced by distinct models and then determine the corresponding best decision; or,
- (CDec) first, for each of the predictive models, determine the corresponding best decision, and then combine (using optimal weights) those decisions.

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) nonequivalent (and non-differentiable) performance measure.

One might conjecture that the first of the above two approaches is, in general, preferable, and that one would benefit from improved performance as

early as possible in the predictive–prescriptive global model. However, this might depend on several factors, including the nature of the LC function.

Moreover, while supervised neural models 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.

In particular, we can consider the following piecewise linear asymmetrical cost function, common in many real-world applications:

$$(\min)D = \frac{1}{N-l+1}\sum_{k=l}^{N} d_k,$$

where

$$d_k = \begin{cases} ue_k, & e_k \geqslant 0, \\ -ve_k, & e_k < 0. \end{cases}$$

At time k, the optimal (minimum expected cost) decision for the next time step, Q_{k+1}, can be defined as a quantile of the distribution of the random variable Y_{k+1}, conditional to the observations available up to time k:

$$F_k(Q_{k+1}) = P[Y_{k+1} \leqslant Q_{k+1}] = \frac{u}{u+v},$$

where F_k is the conditional distribution function of Y_{k+1}, which can be empirically estimated from past predictive errors.

5.2. Experimental results

To illustrate and assess the difference between the two approaches we set up an experiment, using a time series, T (shown in Fig. 3), simulated according to a DTR-IAR(1) model with $\alpha = 0.9$ and $\mathrm{NVR} = 10^{-2}$. To make the series somewhat more interesting, the observation error, $\{\varepsilon_k\}$, was generated from an asymmetrical generalized lambda distribution.

Two different suboptimal forecasting models have been considered: the DTR-RW ($\alpha = 0$) and the DTR-IRW ($\alpha = 1$) models, with optimized noise variance ratios. The best results were obtained with $\mathrm{NVR} = 10^{-1}$ and $\mathrm{NVR} = 10^{-2}$, for which $\mathrm{RMSE}^{(1)} = 1.70$ and $\mathrm{RMSE}^{(2)} = 1.69$, respectively. The estimates produced by the two models are shown in Fig. 4.

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

$$\hat{y}_k^{(c)} = w_0 + w_1\hat{y}_k^{(1)} + w_2\hat{y}_k^{(2)} + \pi\hat{y}_k^{(1)}\hat{y}_k^{(2)},$$
$$Q_k^{(c)} = w_0' + w_1'Q_k^{(1)} + w_2'Q_k^{(2)} + \pi'Q_k^{(1)}Q_k^{(2)},$$

where the weights were optimized with respect to the corresponding loss functions: LS for predictions and LC for decisions. We set $u = 9$ and $v = 1$.

To determine the quantiles that define the optimal decisions, one needs to estimate the distribution function associated to the random variables Y_{k+1}. To simplify this process, we have just considered a sliding time window consisting of the last 50 estimated forecasting errors; these were sorted,

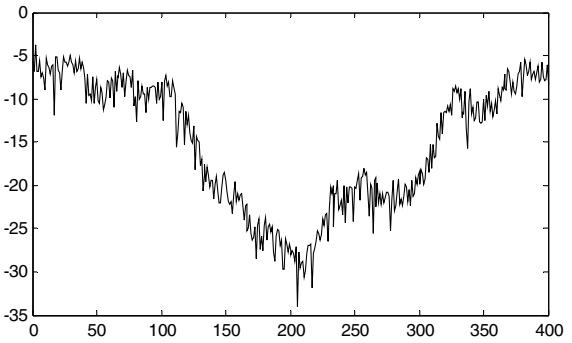

Fig. 3. Plot of time series T.

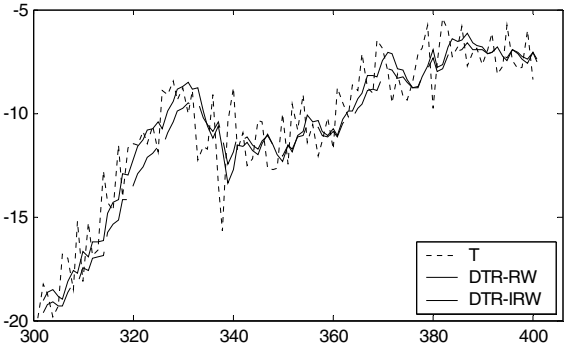

Fig. 4. One-step-ahead predictive estimates from models DTR-RW and DTR-IRW, for time series T (last 100 values).

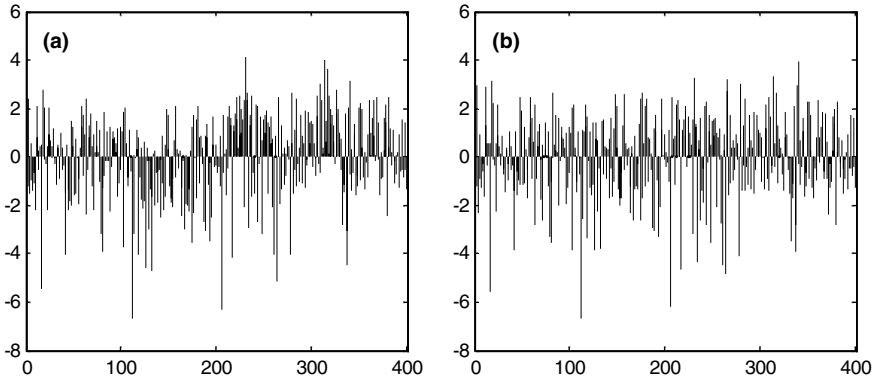

Fig. 5. Forecasting errors from two models applied to T: (a) DTR-RW, (b) DTR-IRW.

and estimates for the quantiles were then easily determined. In addition, this allowed us to overcome the problem of nonnormality and autocorrelation of the errors that were obtained by the individual models (see Fig. 5).

Table 4 shows the results obtained by the individual models and the combination of them related to prediction and decision problems, respectively. In both approaches the combination of models has performed better than the individual ones. The decision costs obtained by each approach are shown in Fig. 6; it is noticeable that the two sequences are very similar, but distinct. The minimum average costs were: for approach CPred, $D = 2.539$; and, for approach CDec, $D = 2.427$. This means that, in the present example, it was preferable to combine the quantiles, rather than to combine the predictive estimates.

Table 4
Results of individual and combined models with respect to the prediction and decision optimization problems

	Prediction (RMSE*)	Decision (D^*)
DTR-RW	1.704	2.540
DTR-IRW	1.668	2.630
Optimal comb.	1.656	2.427

6. Conclusions

Computational efficiency is of critical importance when dealing with large data sets, large collections of data, or with streamed data. Further difficulties arise when the data is noisy, nonstationary and nonlinear, requiring more complex and

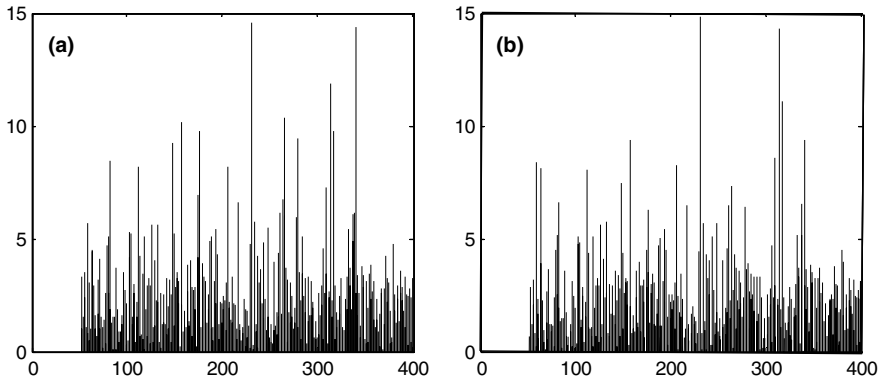

Fig. 6. Sequences of decision costs resulting from two approaches: (a) CPred; (b) CDec.

flexible yet robust forecasting models. Model optimality is very difficult, or virtually impossible to achieve, but through the optimal combination of suboptimal solutions, one may hope to efficiently obtain better quality solutions.

In this paper, we studied different ways of combining 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 patterns to the neural networks.

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.

In the context of model mixing, we proposed an extension of the usual linear combination framework, able to cope with the case where the forecasting errors from the different models are significantly cross-correlated.

In the context of model synthesis, we proposed prefiltering the data, before the identification of a neural model. This methodology might 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, or rather combining several decisions inferred from different forecasting models. Experiments like the one presented above give indication that none of the approaches is universally preferable, so it may be advisable to try both of them, and choose the one that minimizes the expected cost.

Appendix A. Optimal linear combination

First, we review some basic results on the optimization of vector functions.

Given $L(\mathbf{w})$, a real valued differentiable function of vector $\mathbf{w} = [\, w_1 \quad \cdots \quad w_m \,]^{\mathrm{T}}$, a necessary condition for a local optimum (minimum or maximum) at $\mathbf{w}_0$ is that $\frac{\partial L}{\partial \mathbf{w}}(\mathbf{w}_0) = \mathbf{0}$. If this condition is satisfied and the Hessian matrix of second partial derivatives, $\frac{\partial^2 L}{\partial \mathbf{w}\, \partial \mathbf{w}^{\mathrm{T}}}$, is positive/negative definite for $\mathbf{w} = \mathbf{w}_0$, then $\mathbf{w}_0$ is a local optimum.

The vector $\mathbf{w}$ that minimizes the least-squares cost function

$$L(\mathbf{w}) = (\mathbf{y} - \mathbf{U}\mathbf{w})^{\mathrm{T}}(\mathbf{y} - \mathbf{U}\mathbf{w})$$

is given by $\mathbf{w}^* = \mathbf{U}^{+}\mathbf{y}$, where $\mathbf{U}^{+} = (\mathbf{U}^{\mathrm{T}}\mathbf{U})^{-1}\mathbf{U}^{\mathrm{T}}$ is the Moore–Penrose pseudo-inverse of the regressor matrix $\mathbf{U}$ (provided $\mathbf{U}^{\mathrm{T}}\mathbf{U}$ is not singular) [17]. As the second derivative of L with respect to $\mathbf{w}$ is the matrix $2\mathbf{U}^{\mathrm{T}}\mathbf{U} \geqslant 0$, the solution to the problem is indeed a minimum.

Incidentally, the least-squares solution to the estimation problem of the static linear model $y_k = \mathbf{u}_k^{\mathrm{T}}\boldsymbol{\theta} + \varepsilon_k$, with constant parameters, is $\boldsymbol{\theta}^* = \mathbf{U}^{+}\mathbf{y}$.

Let us now consider the following framework for the combination of the forecasts produced by two models:

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

where

$$\mathbf{w} = [\, w_0 \quad w_1 \quad w_2 \,]^{\mathrm{T}},$$
$$\mathbf{F} = \left[\, \mathbf{1} \quad | \, \hat{\mathbf{y}}^{(1)} \, | \quad \hat{\mathbf{y}}^{(2)} \,\right]$$

($\mathbf{1}$ is a column vector of ones),

$$\hat{\mathbf{y}}^{(i)} = \left[\, \hat{y}_n^{(i)} \quad \cdots \quad \hat{y}_N^{(i)} \,\right]^{\mathrm{T}} \quad \text{and}$$

$$\mathbf{f} = \left[\, \hat{y}_n^{(1)}\hat{y}_n^{(2)} \quad \cdots \quad \hat{y}_N^{(1)}\hat{y}_N^{(2)} \,\right]^{\mathrm{T}}.$$

The optimization problem with respect to the least squares criterion is defined by

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

where $e_k^{(c)} = y_k - \hat{y}_k^{(c)}$, $n \leqslant k \leqslant N$.

Then, the sum of squared errors is

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

where $\mathbf{V} = \mathbf{F}^{\mathrm{T}}\mathbf{F}$, $\mathbf{b} = \mathbf{F}^{\mathrm{T}}\mathbf{y}$, $\mathbf{z} = \mathbf{F}^{\mathrm{T}}\mathbf{f}$, $d = \mathbf{f}^{\mathrm{T}}\mathbf{y}$ and $u = \mathbf{f}^{\mathrm{T}}\mathbf{f}$.

After differentiating, and setting the normal equations system:

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

the optimal solution is found to be

$$\begin{cases} \mathbf{w}^* = \mathbf{V}^{-1}(\mathbf{b} - \pi^*\mathbf{z}), \\ \pi^* = \dfrac{d - \mathbf{z}^{\mathrm{T}}\mathbf{V}^{-1}\mathbf{b}}{u - \mathbf{z}^{\mathrm{T}}\mathbf{V}^{-1}\mathbf{z}}. \end{cases}$$

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

The corresponding optimal sum of squared errors is, after substituting for $\mathbf{w}^*$ and π^*:

$$\mathrm{SSE}^* = \mathbf{y}^{\mathrm{T}}\mathbf{y} - \mathbf{b}^{\mathrm{T}}\mathbf{V}^{-1}\mathbf{b} - \frac{(d - \mathbf{z}^{\mathrm{T}}\mathbf{V}^{-1}\mathbf{b})^2}{u - \mathbf{z}^{\mathrm{T}}\mathbf{V}^{-1}\mathbf{z}}.$$

The formulae derived above are valid irrespectively of the nature of vector $\mathbf{f}$. Therefore, this can be viewed as an extension of the classic linear form to the case where there is a third sequence of estimates to be considered in the combination.

If π is set to zero, then we have the usual linear formulation, for which the optimal solution is

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

and

$$\mathrm{SSE}_L^* = \mathbf{y}^{\mathrm{T}}\mathbf{y} - 2\mathbf{w}_L^{*\mathrm{T}}\mathbf{b} + \mathbf{w}_L^{*\mathrm{T}}\mathbf{V}\mathbf{w}_L^* = \mathbf{y}^{\mathrm{T}}\mathbf{y} - \mathbf{b}^{\mathrm{T}}\mathbf{V}^{-1}\mathbf{b}.$$

Hence, the optimal weights for the extended form can be written as

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

and

$$\mathrm{SSE}^* = \mathrm{SSE}_L^* - \frac{(d - \mathbf{z}^{\mathrm{T}}\mathbf{w}_L^*)^2}{u - \mathbf{z}^{\mathrm{T}}\mathbf{V}^{-1}\mathbf{z}} = \mathrm{SSE}_L^* - \pi^*\mathbf{f}^{\mathrm{T}}\mathbf{e}_L^{(c)},$$

where $\mathbf{e}_L^{(c)}$ is the sequence of errors obtained from the classic linear form.

References

[1] A.M. Bagirov, A.M. Rubinov, N.V. Soukhoroukova, J. Yearwood, Unsupervised and supervised data classification via nonsmooth and global optimization, Top 11 (1) (2003) 1–93.

[2] J.M. Bates, C.W.J. Granger, The combination of forecasts, Operational Research Quarterly 20 (4) (1969) 451–468.

[3] C.M. Bishop, Neural Networks for Pattern Recognition, Clarendon Press, Oxford, 1995.

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

[5] J.L. Carmo, A.J. Rodrigues, Identificação de redes neuronais Gaussianas como modelos de previsão (in Portuguese), Investigação Operacional 22 (2002) 43–57.

[6] J.L. Carmo, A.J. Rodrigues, Adaptive forecasting of irregular demand processes, Engineering Applications of Artificial Intelligence 17 (2004) 137–143.

[7] K.H. Chan, J.C. Hayya, J.K. Ord, A note on trend removal methods: The case of polynomial regression versus variate differencing, Econometrica 45 (3) (1977) 737–744.

[8] F.X. Diebold, A.S. Senhadji, Deterministic vs. stochastic trend in US GNP, yet again, American Economic Review 86 (1996) 1291–1298.

[9] R.C. Eberhart, J. Kennedy, A new optimizer using Particle Swarm Theory, in: Proc. of the Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, 1995, pp. 39–43.

[10] P.S.A. Freitas, Combinação de Modelos Neuronais na Previsão de Séries Temporais, M.Sc. thesis, (in Portuguese), FCUL, University of Lisbon, Portugal, 1999.

[11] C.W.J. Granger, R. Ramanathan, Improved methods of combined forecasts, Journal of Forecasting 3 (1984) 197–204.

[12] A. Guay, P. St-Amant, Do the Hodrick–Prescott and Baxter–King Filters Provide a Good Approximation of Business Cycles? Cahiers de Recherche CREFE/CREFE Working Papers 53, Université du Québec à Montreal, 1997.

[13] L.K. Hansen, P. Salamon, Neural network ensembles, IEEE Transactions on Pattern Analysis and Machine Intelligence 12 (10) (1990) 993–1001.

[14] J.D. Hart, Differencing as an approximate de-trending device, Stochastic Processes and their Applications 31 (1989) 251–259.

[15] S. Hashem, Optimal linear combinations of neural networks, Neural Networks 10 (4) (1997) 599–614.

[16] L. Ljung, T. Söderstrom, Theory and Practice of Recursive Identification, MIT Press, Cambridge, Massachussets, 1983.

[17] J.R. Magnus, H. Neudecker, Matrix Differential Calculus with Applications in Statistics and Econometrics, John Wiley & Sons, 1988.

[18] O. Nelles, Nonlinear System Identification, Springer-Verlag, 2000.

[19] C.R. Nelson, H. Kang, Spurious periodicity in inappropriately detrended time series, Econometrica 49 (3) (1981) 741–751.
[20] P. Newbold, C.W.J. Granger, Experience with forecasting time series and the combination of forecasts, Journal of the Royal Statistical Society A 137 (2) (1974) 131–165.
[21] M.P. Perrone, L.N. Cooper, When networks disagree: Ensemble methods for hybrid neural networks, in: R.J. Mammone (Ed.), Artificial Neural Networks for Speech and Vision, Chapman-Hall, 1993, pp. 126–142.
[22] D.S.G. Pollock, Recursive estimation in econometrics, Computational Statistics & Data Analysis 44 (2003) 37–75.
[23] A.J.L. Rodrigues, Dynamic Regression and Supervised Learning Methods in Time Series Modelling and Forecasting, Ph.D. thesis, Lancaster University, England, 1996.
[24] P.X.G. Silva, Previsão de Séries Temporais Não Estacionárias por Redes Neuronais, M.Sc. thesis (in Portuguese), FCUL, University of Lisbon, Portugal, 1998.
[25] S. Waterhouse, G. Cook, Ensemble methods for phoneme classification, in: M.C. Mozer et al. (Eds.), Advances in Neural Information Processing Systems, 9, MIT Press, 1997, pp. 800–806.
[26] R.L. Winkler, S. Makridakis, The combination of forecasts, Journal of the Royal Statistical Society A 146 (2) (1983) 150–157.
[27] P.C. Young, Recursive Estimation and Time Series Analysis: An Introduction, Springer-Verlag, 1984.
[28] G. Zhang, B.E. Patuwo, M.Y. Hu, Forecasting with artificial neural networks: The state of the art, International Journal of Forecasting 14 (1) (1998) 35–62.
[29] G.P. Zhang, V.L. Berardi, Combining multiple neural networks for time series forecasting, in: Proceedings of the Decision Science Institute Annual Meeting, 2000, pp. 966–968.