International Journal of Neural Systems, Vol. 18, No. 6 (2008) 469–480
© World Scientific Publishing Company

SELF-ORGANISING MIXTURE AUTOREGRESSIVE MODEL FOR NON-STATIONARY TIME SERIES MODELLING

HE NI*,† and HUJUN YIN†,‡
*School of Finance, Zhejiang Gongshang University
HangZhou, 310018, P.R.China
†School of Electrical and Electronic Engineering
The University of Manchester, Manchester, M60 1QD, UK
*nihe@mail.zjgsu.edu.cn
‡h.yin@manchester.ac.uk

Modelling non-stationary time series has been a difficult task for both parametric and nonparametric methods. One promising solution is to combine the flexibility of nonparametric models with the simplicity of parametric models. In this paper, the self-organising mixture autoregressive (SOMAR) network is adopted as a such mixture model. It breaks time series into underlying segments and at the same time fits local linear regressive models to the clusters of segments. In such a way, a global non-stationary time series is represented by a dynamic set of local linear regressive models. Neural gas is used for a more flexible structure of the mixture model. Furthermore, a new similarity measure has been introduced in the self-organising network to better quantify the similarity of time series segments. The network can be used naturally in modelling and forecasting non-stationary time series. Experiments on artificial, benchmark time series (e.g. Mackey-Glass) and real-world data (e.g. numbers of sunspots and Forex rates) are presented and the results show that the proposed SOMAR network is effective and superior to other similar approaches.

Keywords: Time series; autoregressive models; non-stationarity; self-organising map; mixture of temporal models.

1. Introduction

Non-stationary time series modelling has attracted a great deal of attention and effort from researchers, because their wide existence in real-world applications such as finance, astronomy, seismology and neurophysiology. Most existing techniques developed in signal processing theory assume that the time series being dealt with are stationary or can be converted to stationary.[1] However, most of real-world signals are not in line with this assumption and their statistical properties change with time. Developing methods for modelling non-stationary signals has become an active area of research in both statistics and signal processing. Many adaptive neural network models have been adopted to extend linear regressive models.

For instance, autoregressive moving average (ARMA) models assume a structured linear relationship between the current value of the variable and previous values of the variable and error terms. The relationship is assumed to be described by a set of fixed parameters. Apparently this assumption is a pitfall of ARMA models when the time series is non-stationary. Although the autoregressive integrated moving average (ARIMA), a generalisation of the ARMA model, can better fit the non-stationary time series, it doesn't outperform neural network based methods in our experiments. According to the definition of ARIMA model,[1] the differenced FX rates or FX returns modelled by an ARMA model is theoretically equivalent to modelling the FX rates by an ARIMA model, where the order of integrated equals to 1.

Nonparametric models offer a good alternative to parametric ones. Not restricted by a fixed structure, a nonparametric model is relatively more flexible,

[‡] Corresponding author.

but at the cost of extensive computation. Due to the recent advances in computational intelligence and computer power, nonparametric models have been used extensively in the last few years with various success. For example, forecasting by neural networks provides strong evidence in terms of out-of-sample forecasting achievements.[3] Comparative studies show that adaptive methods significantly outperform the linear ARMA model and the naive random walk models.[3–5] The most widely used techniques so far are the multilayer perceptron (MLP), radial basis function (RBF) networks, support vector machines (SVM) and recurrent networks.

Non-stationarity implies that the time series switch their dynamics in different time regions. Empirical studies[6] show that the distribution of daily returns[a] is approximately symmetric and leptokurtic (i.e., fat tailed). One intuitive explanation for the fat-tailed distribution is that samples are independently distributed as a normal distribution whose mean and variance change over time. Alternatively, Clark and Mandelbrot argued that observed returns come from a mixture of normal distributions.[7,8] It is apparently not convincing for a single model to capture the dynamics of the whole time series. A potential solution can be found by using a mixture model approach, in which the entire model is divided into several smaller ones. Then regression and prediction are made by a dynamic set of best fit local models.

The self-organising map (SOM) can be used to divide a time series into segments. A number of regressive models can then be created by locally fitting to the corresponding parts of the time series. There have been a few successful applications. For instance, Dablemont[9] applied SOM-based local models with RBF network as regressors to predict the returns of the DAX30 index. Liu and Xu[10] used SOM-based local models to perform PCA on the data from multi-modes. Cao[5] proposed a SVM expert system, which is also based on SOM local models, to predict the future movement of a time series. However, these models are two-stage modelling. Both clustering and local modelling may not be optimal, or at least may not be jointly optimal.

In addition, SOM is a spatial model and does not naturally fit temporal signals. Although consecutive time points can be grouped into vectors to form temporal context by means of a sliding window, the vectors formed may not capture all temporal information of the time series. Several variants have since been proposed such as, the recurrent SOM (RSOM)[11] and the recursive SOM (RecSOM).[12] These SOM variants integrate the information of a sequence via recursive operations. As they differ in the notion of context, their efficiency in terms of representing temporal context are different.[13] Lampinen and Oja proposed a self-organising autoregressive (SOAR) model, where every unit represents an autoregressive (AR) model with its reference vector as the model parameters.[16] The experiments have shown that the model can learn to distinguish textures from images. The method in fact is a multiple AR model with the parameters of component models forming topological orders. However, the model is difficult to converge to the underlying regressive models due to the simple error-based similarity measure. Here, we extend such a model to a multi-regressive mixture model, termed self-organising mixture autoregressive (SOMAR) model, with a different partition mechanism and similarity measure to reflect the characteristics of homogeneous time series. It combines the strength of the SOM and simplicity of the AR model. Both the mixture and local models are jointly optimised, and thus it is better than two-stage modelling, e.g.[5,17] Good initial results have been reported in Ref. 18. In the two-stage process, the information that may be useful in the regression stage (second stage) can be inevitably lost in the clustering stage (first stage), since the two stages are separately optimized. Compare to SOAR model, the proposed method is more capable in modelling dynamic time series.

Neural gas (NG)[19] is another popular competitive vector quantization method in analog to the SOM. Instead of having a fixed network topology throughout, NG can dynamically deploy its resources to suit varying topology of the data, and therefore does not suffer from topological restrictions like the SOM. Apart from this advantage, the NG algorithm[20] is very much similar to the

[a] The distribution of values that are simple logarithmic differences between the daily closing prices of two consecutive trading days.

SOM algorithm, and is regarded as a variant of the SOM. In the proposed model, we replace SOM-based neighbourhood function with the NG-based updating function. The results show that the combination of NG-based clustering and AR regression further improves on the initial outcome in Ref. 18.

The remainder of the paper is organised as follows. In Section 2, we describe the proposed methodology. Section 3 will present the application of the proposed methodology for prediction of various non-stationary time series signals. Finally, conclusions are given in Section 4.

2. Methodology

The most straightforward way of predicting a future value of a time series from its past is by using a correctly tuned regressive model. Identifying a correct regressive model with its parameters is therefore the basis of the proposed prediction model. Apparently a non-stationary time series can hardly be modelled by a single AR model. Therefore one can assume that a non-stationary time series consists of several independent stationary AR processes[b] with the expectation that each AR process can be represented by a local AR model. This is thus in line with the argument that a non-stationary time series comes from a mixture of normal distributions.[7,8] The complete model can be considered as a mixture of these local AR models. The output of the SOMAR is the output of the best fitted local models.

$$\hat{x_t} = \sum_{i=1}^{K} \beta_{(i,\mathbf{x}(t))} \Phi_i(\mathbf{x}(t), \phi_i) \,. \tag{1}$$

where Φ_i represents the i-th local AR model and $\beta_{(i,\mathbf{x}(t))}$ are the mixing parameters. Since we define the prediction of the whole system as the prediction of the best fitting local model, $\beta_{(i,\mathbf{x}(t))}$ are then considered as gate functions, given as,

$$\beta_{(i,\mathbf{x})} = \begin{cases} 1 & \text{if } i = v \\ 0 & \text{else} \end{cases} \tag{2}$$

$\mathbf{x}(t) = [x(t-1), x(t-2), \ldots, x(t-m_i)]^T$ is the input vector, $\{\phi_i\}$ are model parameters, K is the number of AR processes, v is the index of the winning local model at each time step and m_i is the order of AR process i. The value of m_i can be determined by the Bayesian Information Criterion (BIC).[15] For example, in the experiment of Section 3, the order is shown by model validation to be a constant for each local model. In such a case, the proposed SOMAR network is a homogeneous mixture of local AR models.

2.1. *Lampinen and Oja's self-organising AR models*

Lampinen and Oja proposed a method called self-organising AR map (SOAR) based on a self-organising map of "neural" units for unsupervised segmentation and classification of 1D and 2D signals.[16] In the SOAR model, each neuron's weight vector $\mathbf{w}_i$ represents the parameters of a local AR model. The best matching unit is selected based on the smallest exponentially-smoothed error.

$$y_i^t = \alpha e_i(t) + (1-\alpha) y_i^{t-1} \,. \tag{3}$$

where α is a smoothing factor and $e_i(t) = x(t) - \mathbf{x}(t)^T \mathbf{w}_i$. The winner and its neighbouring units update their weights according to

$$\mathbf{w}_i(t) = \mathbf{w}_i(t-1) + g(r) e_i(t) \mathbf{x}(t) \,. \tag{4}$$

where $g(r)$ is a linear neighbourhood function and r is the distance between the best matching unit and the unit to be updated. The model has been shown to work in segmenting images into texture classes.

The SOAR model seamlessly links up the SOM clustering with the AR model. It yields positive experimental results on 2D textures.[16] However it can often fail in identifying and thus modelling the correct local AR models in a non-stationary time series. For example, Fig. 1 shows the divergence of the SOAR model in modelling a time series of a mixture of two AR(1) processes. The initial weights were set either randomly (upper figure) or to the true parameters (lower figure). The failure can be attributed to the fact that the weight updating of the SOAR local models is based on wrongly classified time series segments. That is, at each time step, the SOAR fails to identify the correct local model due to the stochastic nature of the error based similarity measure.

[b] This approach is different from the usual approach that transforms a non-stationary time series into a stationary one, in particular by removing linear trends.[14]

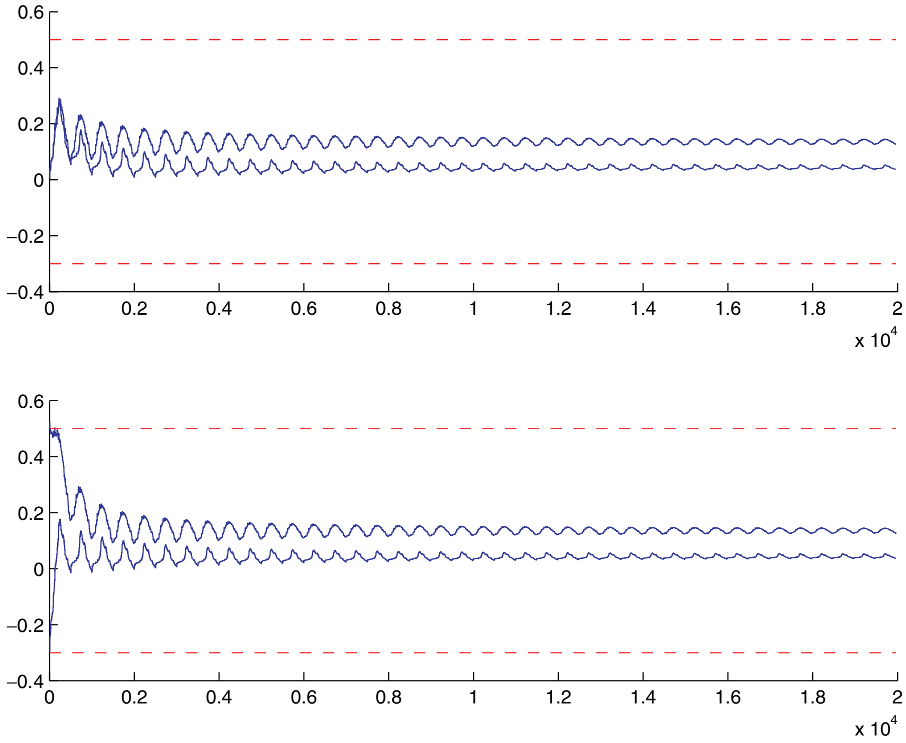

Fig. 1. Parameter estimation of two AR(1) processes by the SOAR with initial weights set either randomly (upper plot), and to the true parameters (lower plot). The dashed lines represent the true parameters.

2.2. *Self-organising mixture AR model (SOMAR)*

Improving on the SOAR model, the proposed self-organising mixture AR (SOMAR) network not only combines topological clustering and linear autoregression, but also uses a better model-matching method for identifying local regressive models to ensure the convergence of the network. As a sufficient condition, the modelling error should be gradually close to white noise if the modelling is following a "correct" path. Therefore, we employ the autocorrelation of the error instead of the error itself to measure the fitness of a local model to an input segment. The size of the segments can vary to suit the correct orders of the local models. Though modelling errors of a local model may not decrease monotonically in its values (due to the stochastic gradient descent nature on the modelling error generated by a recursive LMS algorithm), the error should become closer to white noise when the structure of the data is gradually learnt. The autocorrelation of a time series is used as an efficient measure for its whiteness. The more white an error series is, the smaller the sum of its autocorrelation coefficient; and vice versa. Assume that we have a consecutive set of p modelling errors, $\{e(1), e(2), \ldots, e(p)\}$, with mean μ and variance σ^2. p is an integer larger than one, which is the number of consecutive modelling errors used for finding the winning local model. At the time when a segment is fed to the SOMAR, the winning local model is the one which generates the smallest sum of (the absolute value of) autocorrelation coefficients (SAC) of the modelling errors,

$$R_i(k) = \frac{1}{(p-k)\sigma^2} \sum_{t=1}^{p-k} (e_i(t) - \mu_i)(e_i(t+k) - \mu_i)\,, \quad (5)$$

where k is the number of lags and i is the index of a local regressive model. The winner is then,

$$v = \arg\min_i \left(\sum_{j=-k}^{k} |R_i(k)| \right) \quad i = 1, 2, \ldots, N, \quad (6)$$

where the absolute autocorrelation coefficient is used as non-white errors may be either positively or

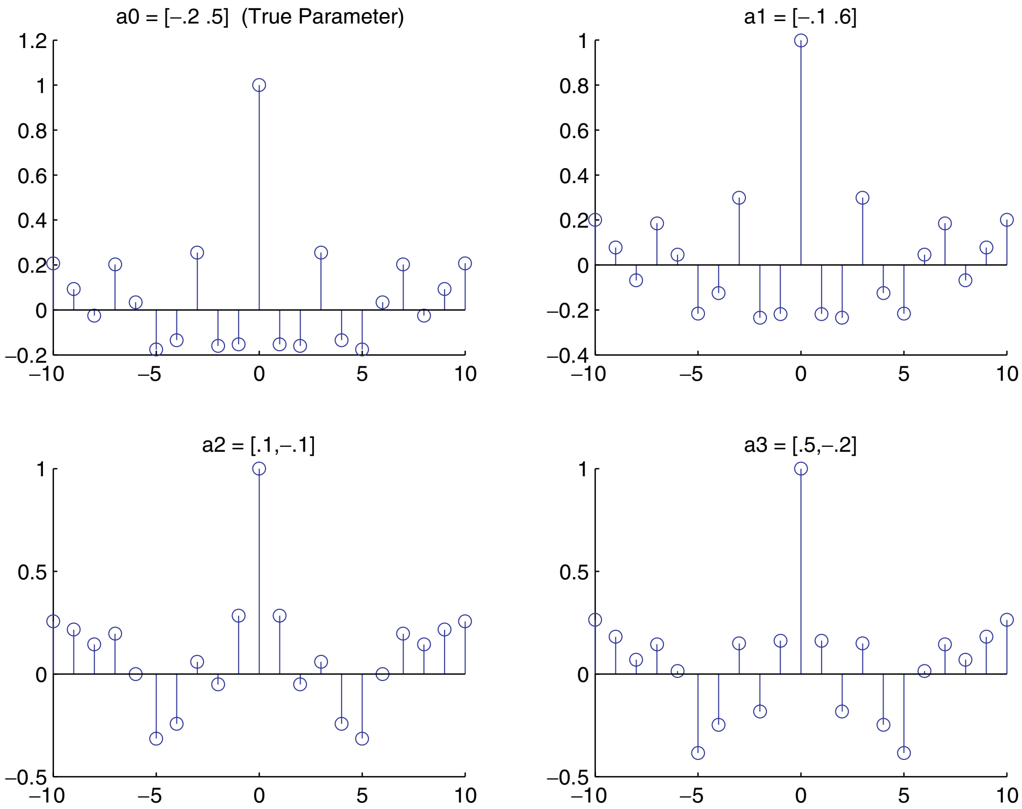

Fig. 2. Autocorrelation of the modelling errors for models $\mathbf{a}_0$ to $\mathbf{a}_3$. Corresponding SAC values are 3.8832, 4.3504, 4.5224 and 4.5963 respectively.

negatively correlated, and N is the number of local models.

Figure 2 shows the autocorrelation coefficients of the modelling errors of a patch of 20 points. The generating parameters of the AR(2) process are $\mathbf{a}_0 = [-.2, .5]$. We test the SAC value on the true parameters and three randomly chosen sets of parameters, $\mathbf{a}_1 = [-.1, .6]$, $\mathbf{a}_2 = [.1, -.1]$ and $\mathbf{a}_3 = [.5, -.2]$. Their correlations are plotted in Fig. 2. The SAC values for these four cases are 3.8832($\mathbf{a}_0$), 4.3504($\mathbf{a}_1$), 4.5224($\mathbf{a}_2$), and 4.5963($\mathbf{a}_3$) respectively. As we can see that, $\mathbf{a}_0$ is the best fit model, and model $\mathbf{a}_1$ is closer to model $\mathbf{a}_0$ than models $\mathbf{a}_2$ and $\mathbf{a}_3$, as are their SAC values. Therefore a better way to identify the correct local model is to use the SAC value instead of the modelling error itself.

In order to verify that SAC has only one unique global minimum corresponding to the correct regressive model, we empirically calculate the SAC values of a number of models whose parameters are within the vicinity of the pre-set parameter $[-.2, .5]$. The SAC values with regard to model parameters are illustrated in a contour plot in Fig. 3.

The SOMAR network thus extends the conventional SOM-based local regression model, e.g. Refs. 13 and 17, and the SOAR model for better temporal modelling capability of (non-stationary) time series.

Analogous to the algorithm in Ref. 18, the winning local AR model and its neighbouring models update their model parameters by the ordinary recursive least-mean-squared method.[21] The adaption of the parameters $\mathbf{w}_i(t)$ are further weighted by the neighbourhood function $h(\lambda, k_i)$ and adaptation strength $\eta(t)$. Therefore the overall updating rule is,

$$\mathbf{w}_i(t) = \mathbf{w}_i(t-1) + h(\lambda, k_i)\eta(t)e_i(t)\mathbf{x}(t), \qquad (7)$$

where $e_i(t) = x(t) - \mathbf{x}(t)^T\mathbf{w}_i$.

Furthermore, neural-gas (NG) topology is used in stead of the common lattice structure. The

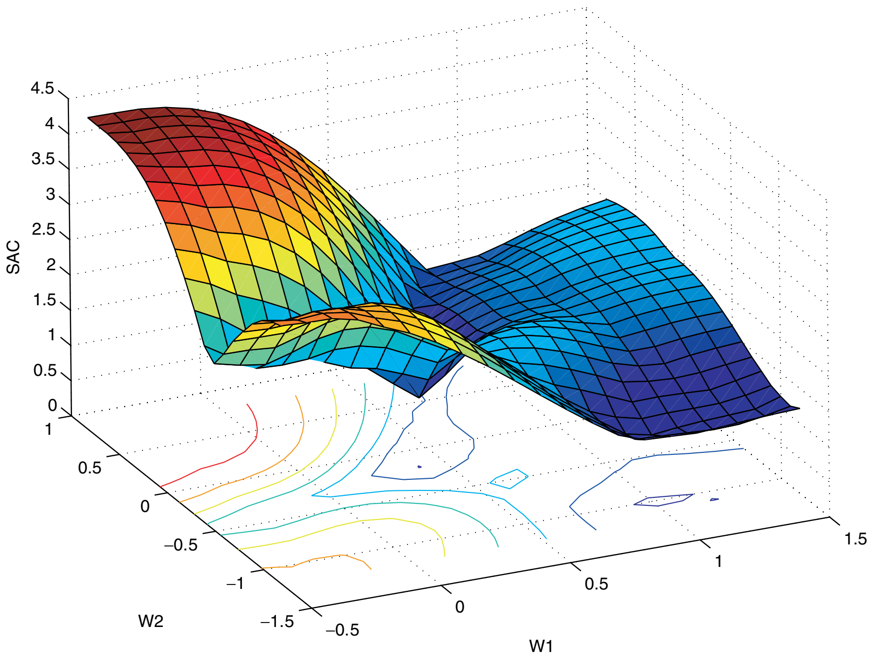

Fig. 3. Contour plot of SAC values around the true model parameters, $w = [-.2, .5]$.

neighbourhood function, which controls a certain number of units being adapted, is the "rank order" of the NG. Similar to the neighbourhood function of the SOM, $h(\lambda, k_i)$ sets a fixed schedule for decreasing the effective range of the adapting neighbourhood,

$$h(\lambda, k_i) = \exp(-k_i/\lambda(t)) \tag{8}$$

$$\lambda(t) = \lambda_i(\lambda_f/\lambda_i)^{t/t_{\max}} \tag{9}$$

where λ_i, λ_f are two parameters controlling the range of the neurons to be adapted. $t_{\max}$ is the maximum number of time steps of the training. k_i is the rank order of the unit: $\mathbf{w}_{i0}$ is the closest reference vector to the input $\mathbf{x}(t)$, $\mathbf{w}_{i1}$ is the second-closest reference vector to $\mathbf{x}(t)$, and so on.

The adaptation rate $\eta(t)$ is the same to the learning rate of the common SOM algorithm and is decreasing with time. It can be either linear or non-linear. A non-linear case, which associates with a better performance than a linear form, is used in training,

$$\eta(t) = \eta_i(\eta_f/\eta_i)^{t/t_{\max}} \tag{10}$$

again, η_i, η_f are two controlling parameters that can be set empirically.

Since some input patches may inevitably contain input segments generated by more than one underlying AR processes, it is necessary to filter out those segments when the parameters of local AR models have been roughly identified. The errors generated by these segments would result in small differences in the SAC values among the neighbouring local models. Therefore a fine tuning is introduced at that stage by skipping those segments that have similar SAC values to more than one local models. A threshold value is set to check whether a segment should be filtered out or should be used in the fine tuning stage. Intuitively the fine tuning can start when the learnt parameters have become generally stabilized or the oscillations of the parameters become small. A flowchart for training the SOMAR network is outlined in Fig. 4.

The local models effectively transform a non-stationary time series to a mixture of 'pieces' of stationary time series. The effectiveness can be shown by the unit root test. A widely used unit root test, the Augmented Dickey Fuller (ADF) test,[22] is used to check whether the time series localised by the SOMAR are intra-cluster stationary.

3. Experiments

In this section, experiments on various data sets, including artificial data generated by mixed AR models, benchmark Mackey-Glass data, and real-world Sunspots observations and FX time series,

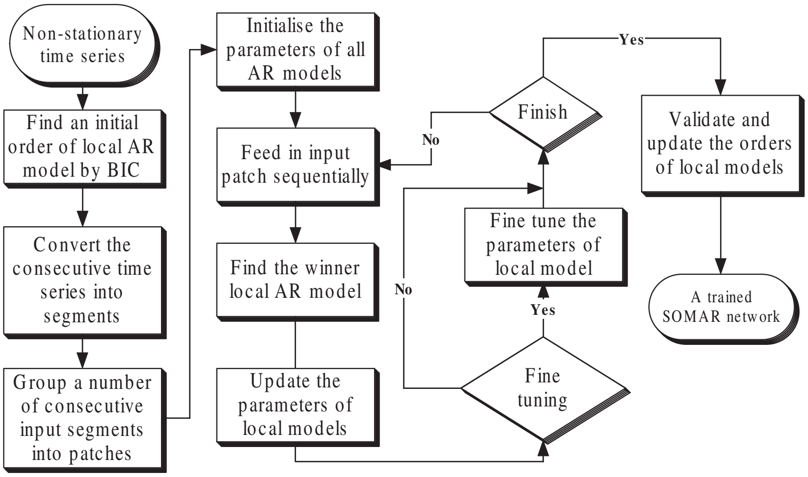
Non-stationary time series
Find an initial order of local AR model by BIC
Convert the consecutive time series into segments
Group a number of consecutive input segments into patches
Initialise the parameters of all AR models
Feed in input patch sequentially
Find the winner local AR model
Update the parameters of local models
Fine tuning
Yes
No
Fine tune the parameters of local model
Finish
No
Yes
Validate and update the orders of local models
A trained SOMAR network

Fig. 4. Flowchart of the SOMAR training procedure. The decision of the 'fine tuning' is made by whether the SAC value is smaller than the pre-determined threshold value. Stopping criterion is made on the convergence of the local models or the maximum number of iterations.

are presented. The proposed SOMAR network is compared with other SOM-based two-stage local regressive model approaches, such as vector SOM (VSOM), recurrent SOM (RSOM), recursive SOM (RecSOM), SOM with SVM regression, neural gas, the SOAR model, as well as the GARCH and ARIMA models.[c]

3.1. *Artificial data*

To illustrate the capability of SOMAR network in finding the underlying components of a mixture time series, we generate such a time series constructed from two AR(2) processes, $\mathbf{a}_1 = [.5, -.2]$ and $\mathbf{a}_2 = [.4, -.3]$. The signal consists of 60 consecutive 200-point long segments. Each segment is generated randomly by one of these two AR processes.

The SOMAR network is trained on the time series and the results, i.e. the weights learnt over time, are shown in the Fig. 5. This figure also shows that the oscillations of the learnt parameters become significantly smaller in the fine tuning phase as expected.

3.2. *Mackey-Glass data*

To further investigate the performance of the proposed SOMAR model in time series prediction, we apply it to 600 consecutive points of Mackey-Glass series, a dynamic system defined by the following differential equation:

$$\frac{dx}{dt} = \beta_{mg} x(t) + \frac{\alpha_{mg} x(t-\delta_{mg})}{1 + x(t-\delta_{mg})^{10}}\,, \tag{11}$$

with the parameters $\delta_{mg} = 17$, $\alpha_{mg} = 0.2$, $\beta_{mg} = -0.1$. The SOMAR network regards the Mackey-Glass data as of a number of unknown AR processes. In this experiment, the input is the Mackey-Glass series grouped by every 12 points $\mathbf{x}(i) = [x(i), x(i-1), \ldots, x(i-11)]$. The order of the AR processes is validated to 11 by the BIC. The results of prediction of the series are listed in Table 1 and a part of results are shown on Fig. 6.

3.3. *Number of sunspots*

Routinely collected sunspot number is one of the measurements of solar activity. The sunspot data is a benchmark data, which has been modelled by various regression and neural network models.[23] The 'cycles' in the number of sunspots, shown in Fig. 7, are not symmetrical, as the spot count takes on average around 5 years to rise from a minimum to a maximum and around 6 years to fall to a minimum. The data was collected from Ref. 24.

Again analogous to the Mackey-Glass data, we assume the sunspot process consists of a number of unknown AR processes. The sunspot series is grouped into every 8 points. The first 2,000 points are

[c] The performance of GARCH and ARIMA is quantified by the best average performance of repeated experiments on different model parameters, when $q = 1, 2, 3$ for GARCH(p, q) model and $d = 0, 1, 2$ for ARIMA(p, d, q) model.

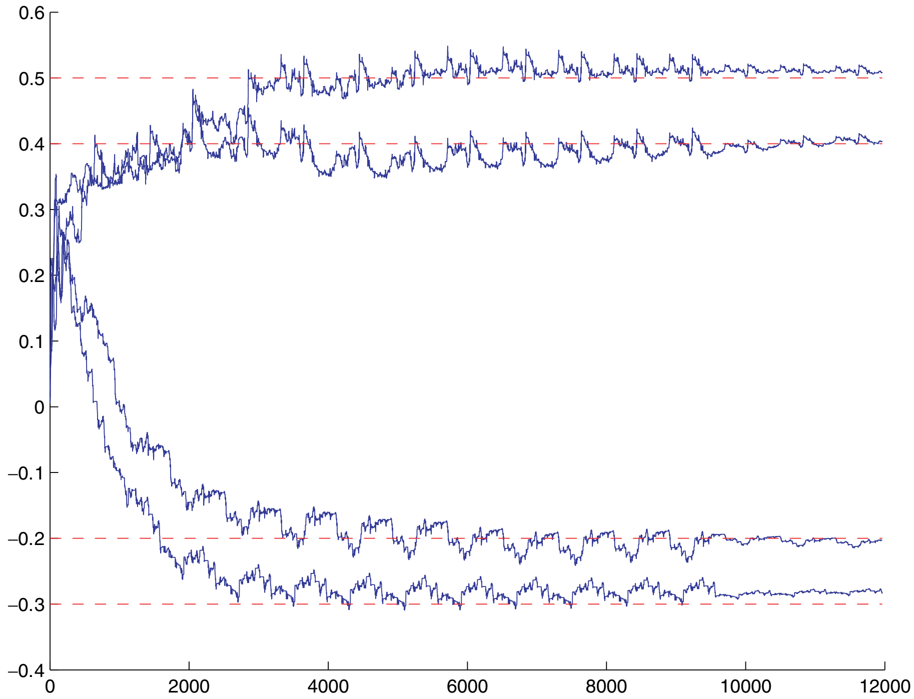

Fig. 5. Parameter estimation of two AR(2) processes $\mathbf{a}_1 = [0.5, -0.2]$ and $\mathbf{a}_2 = [0.4, -0.3]$, by SOMAR network. The solid lines represent the learning progresses and the dashed lines indicate the parameters of two AR(2) models.

Table 1. The performance of Mackey-Glass data predicted in terms of the mean-square-error (between the predicted values and the observed values) and the error variance by various methods.

Prediction model	MSE of prediction($^{-2}$)	MSE variance($^{-2}$)
SOMAR	**3.60**	**0.33**
SOAR	4.26	0.39
VSOM	4.46	0.48
RSOM	4.32	0.53
RecSOM	4.06	0.32
NeuralGas	4.40	0.43
SOM-SVM	4.52	0.46
GARCH	4.49	0.45
ARIMA	4.38	0.52

used to train the SOMAR model and the prediction is performed on the rest of the data. The prediction results are shown in Fig. 8 to illustrate the effect of SOMAR prediction. The results, in terms of the mean-square-error and the error variance, are listed in Table 2.

3.4. *Forex rate data*

The data was retrieved from the PACIFIC Exchange Rate Service provided by W. Antwiler at UBC's Sauder School of Business. It consists of 15 years of daily exchange rates (e.g. British pound vs. US dollar, Euro vs. HK dollar) excluding weekends and bank holidays when the currency markets were closed. The proposed SOMAR network was trained on the first 2,000 consecutive points and the performance was validated on the following 100 points and the prediction was tested on the last 100 points. Both the training and testing (including validation) sets were windowed with the length of 15 points to form input segments.[17,18] The validation of the model order was done by finding out the optimal order of each trained local model in repeated experiments by BIC. The results show that more than 80% of the resulting optimal orders are indifferent to the initially validated order, 14. This result suggests that the proposed SOMAR model implies a mixture of homogeneous local models for the data. The performance of the SOMAR in predicting FX returns and rates are shown in Fig. 9.

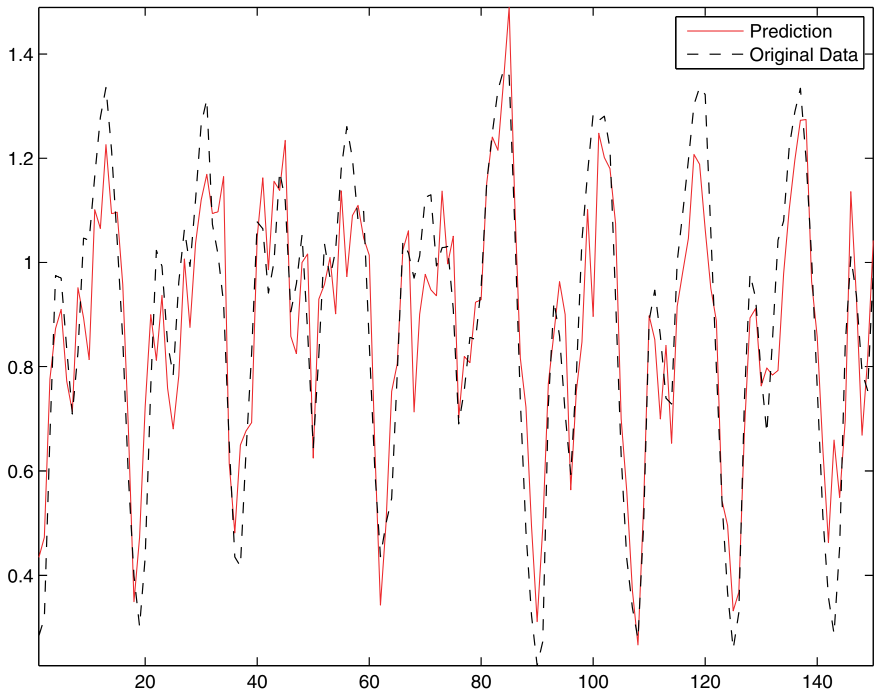

Fig. 6. The dashed line represents the original data points, the solid line represents the prediction by the SOMAR network on the Mackey-Glass data.

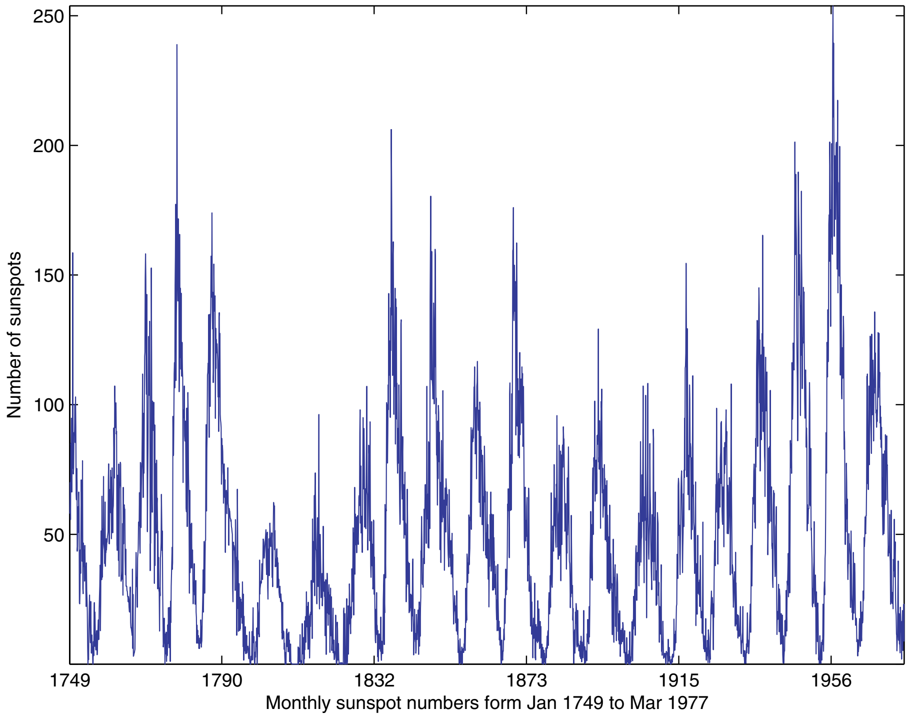

Fig. 7. Monthly means of daily relative sunspot numbers from Jan 1749 to Mar 1977.

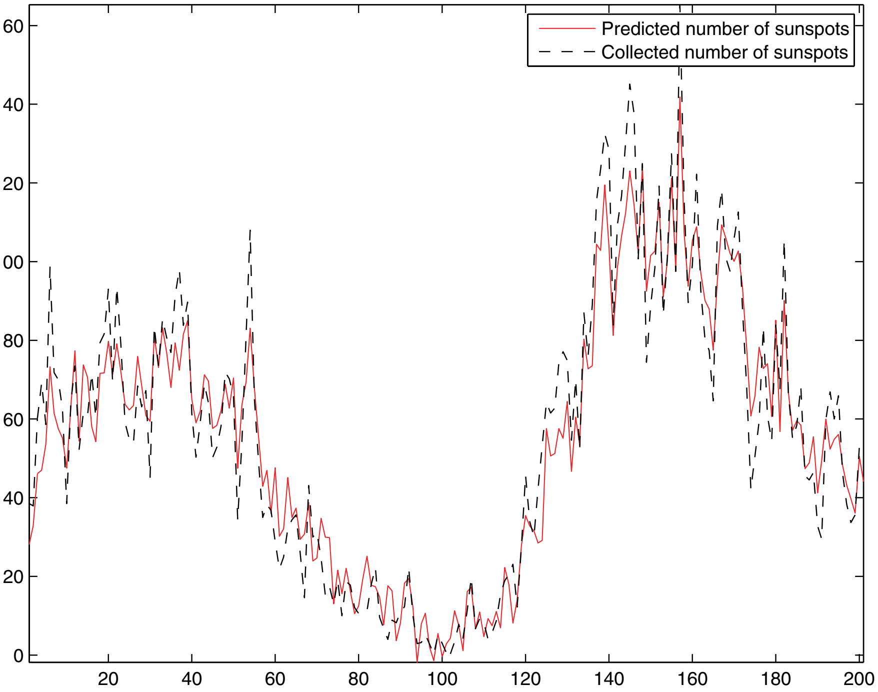

Fig. 8. The dashed line represents the collected sunspot number, the solid line represents the prediction by the SOMAR network.

Table 2. The performance of number of sunspots predicted in terms of mean-square-error (between the predicted values and the observed values) and error variance by various methods.

Prediction model	MSE of prediction	MSE variance
SOMAR	**14.75**	**2.30**
SOAR	16.46	3.56
VSOM	16.23	3.74
RSOM	16.16	3.36
RecSOM	16.12	2.71
NeuralGas	15.71	3.03
SOM-SVM	15.88	3.56
GARCH	16.61	3.53
ARIMA	16.72	3.60

To compare with other regressive models, we conducted the following two types of tests:

Predicted return ($x'_t = \ln \frac{x_{t+1}}{x_t}$) The correct prediction percentage is used as a criterion to check whether the prediction is made in the right direction. In other words, we calculate the percentage of the predicted returns that have the same 'direction' as their corresponding actual returns. The correct prediction percentages are shown in Table 3.

Predicted rate The mean-square-error between the actual FX rates and their predictions in the testing set.

The results of the SOMAR network from the two tests (predicted FX return and predicted FX price) are compared to that of vector SOM, SOAR, recurrent SOM, recursive SOM, neural gas, SOM with SVM regression, and GARCH model and are shown in Table 3. It can be seen that the SOMAR outperforms other temporal SOMs, other local regressive models and GARCH model, ARIMA model. It shows that SOMAR is an efficient method to accommodate the non-stationarity of the FX rates by means of multiple underlying linear processes. Furthermore, the ADF test shows that the localised FX rate segments in all clusters can reject the null hypothesis of the existence of a unit root.[d] The rejection of the existence of a unit root indicates that the SOMAR model

[d] The FX rate vectors are tested against the hypothesis of existence of a unit root using the Matlab function developed by Ludwig Kanzler in the Department of Economics, University of Oxford in March 1998.

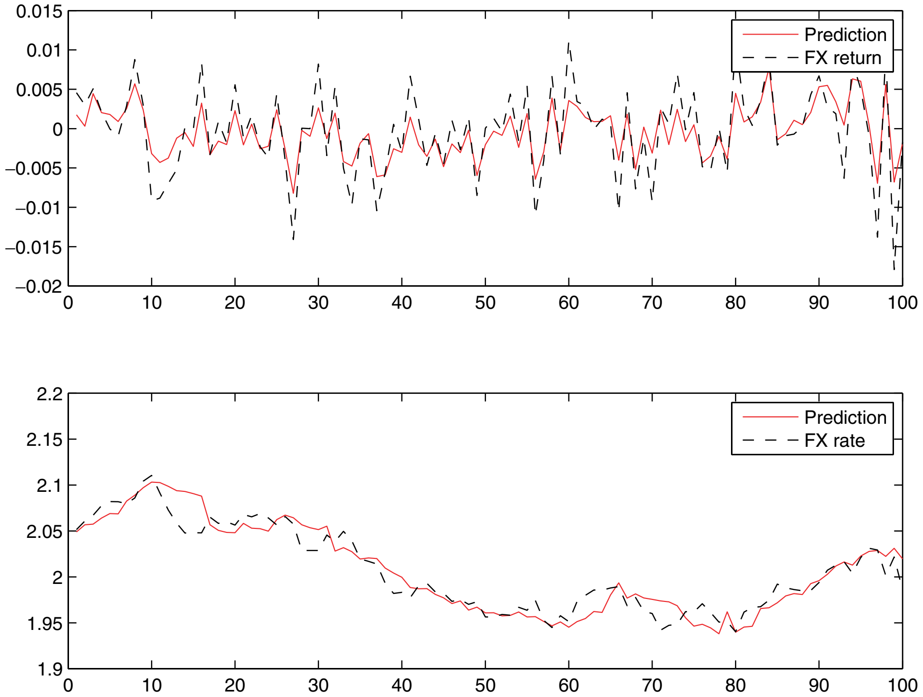

Fig. 9. Prediction of FX return and FX rate in comparing with the original time series on 100 trading days. The dashed lines represent the original data and the solid lines represent the predictions.

Table 3. Overall predicted FX returns in percentage (%) and FX prices* (10^{-2}) of various methods on the FX rates.

GBPvs	USD %	USD $^{-2}$	Euro %	Euro $^{-2}$	HKD %	HKD $^{-2}$
SOMAR	**59.93**	**3.83**	**56.26**	**4.07**	**56.32**	**4.20**
SOAR	53.01	4.33	52.56	4.68	53.56	4.69
VSOM	52.65	4.25	51.98	4.42	53.96	4.75
RSOM	52.20	4.38	53.01	4.62	53.87	4.75
RecSOM	52.65	4.71	53.20	4.97	54.02	4.95
NeuralGas	54.23	4.21	54.25	4.52	54.21	4.45
SOM-SVM	53.52	4.04	54.23	4.68	54.14	4.58
GARCH	53.15	4.25	53.01	4.70	53.73	4.51
ARIMA	50.23	4.95	50.65	4.89	51.36	4.76

* The prices are divided by the mean exchange rate of that pair over the testing period for a fair comparison.

is efficient in transforming non-stationary FX rates into a number of stationary processes.

4. Conclusions

A semi-parametric, mixture approach to tackling non-stationarity of real-world time series has been proposed by using the self-organising mixture autoregressive (SOMAR) network. It consists of a number of local autoregressive models and is organised and learnt by the neural gas and adaptive LMS algorithms, so forming a topologically ordered mixture of local regressive models. The proposed autocorrelation-based similarity measure makes the network more effective and robust in identifying correct local models given input segments, compared to the error-based measures. The SOMAR network combines (local) regression and clustering in a joint estimation, and is proven better than the two-stage approaches. Furthermore, the joint estimation effectively reduces the associated computational cost. The experiments with various data sets show that the proposed model can correctly detect and uncover underlying regressive models. The results also show that the proposed method outperforms other SOM-based methods in modelling and prediction of non-stationary FX rates time series.

References

1. Enders, W., *Applied Econometric Time Series* (2nd edition), John Wiley & Sons. (2004)
2. Meese, R.A. and Rogoff, K., Empirical exchange rate models of the seventies: do they fit out of sample?, *Journal of International Economics* **14** (1983) 3–24.
3. Hann, T.H. and Steurer, E., Much ado about nothing? Exchange rate forecasting: Neural networks vs. linear models using monthly and weekly data, *Neurocomputing* **10** (1996) 323–339.
4. Chen, A.-S. and Leung, M.T., Regression neural network for error correction in foreign exchange forecasting and trading, *Computers and Operations Research* **31** (2004) 1049–1068.
5. Cao, L.J., Support vector machines experts for time series forecasting, *Neurocomputing* **51** (2002) 321–339.
6. Diebold, F.X., Empirical Modeling of Exchange Rate Dynamics (New York, Springer-Verlag) (1988).
7. Clark, P.K., A subordinate stochastic process model with finite variance for speculative price, *Econometrica*, (1973) 135–155.
8. Mandelbrot, B. and Taylor, H., On the distribution of stock price differences, *Operations Research*, **15** (1969) 1057–1062.
9. Dablemont, S., Simon, G., Lendasse, A., Ruttiens, A., Blayo, F. and Verleysen, M., Time series forecasting with SOM and local non-linear models — Application to the DAX30 index prediction, in *WSOM'2003 Proceedings — Workshop on Self-Organizing Maps Hibikino* (Japan) (2003) 340–345.
10. Liu, Z. and Xu, L., Topological local principal component analysis, *Neurocomputing* **55** (2003) 739–745.
11. Koskela, T., Time Series Prediction Using Recurrent SOM with Local Linear Models (Helsinki University of Technology), (2001).
12. Voegtlin, T. and Dominey, P.F., Recursive self-organising maps, *Neural Networks* **15** (2002) 979–991.
13. Strickert, M. and Hammer, B., Merge SOM for temporal data, *Neurocomputing*, **64** (2005) 39–72.
14. Gärdenfors, P. and Hansson, B., Forecasting non-stationary time series Some methodological aspects, Technological Forecasting and Social Change **18** (1980) 63–75.
15. Schwarz, G., Estimating the dimension of a model, *Annals of Statistics* **6** (1978) 461–464.
16. Lampinen, J. and Oja, E., Self-organizing maps for spatial and temporal AR models, in Proc. 6th SCIA, Scandinavian Conference on Image Analysis. Helsinki, Finland (1989) 120–127.
17. Ni, H. and Yin, H., Recurrent self-organising maps and local support vector machine models for exchange rate prediction, *in Proceedings of the Third International Symposium on Neural Networks '06*, LNCS 3973 (2006) 504–511.
18. Ni, H. and Yin, H., Time-series prediction using self-organising mixture autoregressive network, in *Proceedings of the Intelligent Data Engineering and Automated Learning '07*, LNCS 4881 (2007) 1000–1009.
19. Martinetz, T. and Schulten, K., A growing neural gas network learns topologies, *Advances in Neural Information Processing Systems* **7** (1995) 625–632.
20. Martinetz, T., Berkovich, S. and Schulten, K., Neural-gas network for vector quantization and its application to time-series prediction, *IEEE Transactions on Neural Networks*, **4** (1993) 558–569.
21. Haykin, S., Adaptive Filter Theory, Prentice Hall (2002).
22. Dickey, D.A. and Fuller, W.A., Distribution of the estimators for autoregressive time series with a unit root, *Journal of the American Statistical Association* **74** (1979) 427–431.
23. Cottrell, M., Girard, B., Girard, Y., Mangeas, M. and Muller, C., Neural modeling for time series: A statistical stepwise method for weight elimination, *IEEE Transactions on Neural Networks* **6** (1995) 1355–1364.
24. Hyndman, R.J., Time Series Data Library, Accessed by *http://www-personal.buseco.monash.edu.au/hyndman/TSDL/*