Generalized Self-Organizing Mixture Autoregressive Model

Hujun Yin[1] and He Ni[2]

[1] School of Electrical and Electronic Eng., The University of Manchester, Manchester, UK
h.yin@manchester.ac.uk
[2] School of Finance, Zhejiang Gongshang University, Hangzhou, China
nihe@mail.zjgsu.edu.cn

Abstract. The self-organizing mixture autoregressive (SOMAR) model regards a time series as a mixture of regressive processes. A self-organizing algorithm is used with the LMS algorithm to learn the parameters of these regressive models. The self-organizing map is used to simplify the mixture as a winner-take-all selection of local models, combined with an autocorrelation coefficient based measure as the similarity measure for identifying correct local models. The SOMAR has been shown previously being able to uncover underlying autoregressive processes from a mixture. This paper proposes a generalized SOMAR that fully considers the mixing mechanism and individual model variances that make modeling and prediction more accurate for non-stationary time series. Experiments on both benchmark and financial time series are presented. The results demonstrate the superiority of the proposed method over other time-series modeling techniques on a range of performance measures.

1 Introduction

Time series modeling and forecasting is an active, challenging and reoccurring topic in statistics and signal processing owing to their wide use in real-world applications such as communications, speech processing, finance, astronomy and neurophysiology. Linear regression and autoregressive models such as autoregressive (AR), moving average (MA) and autoregressive moving average (ARMA), are commonly used methods. Most linear models assume that the time series being dealt with is stationary and uni-modal [3, 8] and assume a structured linear relationship of constant coefficients between the current value of the time series and its previous values and the error terms. Such conditions are not often met in practice. They are the pitfall of (linear) regressive models when the time series is non-stationary. Developing methods for modeling non-stationary and multimodal time series has become an active area of research. The autoregressive integrated moving average (ARIMA) [3], a generalized ARMA model, can better handle slow changing non-stationary time series by modeling the difference of the consecutive time series values instead of the value itself. The generalized autoregressive conditional heteroscedastic (GARCH) model [2] models the variance of the residual as a linear function of the previous variances, along with the autoregressive model of the time series. It has been a benchmark model for financial data, which exhibits varying volatilities from time to time. The mixture

J.C. Príncipe and R. Miikkulainen (Eds.): WSOM 2009, LNCS 5629, pp. 353–361, 2009.

autoregressive (MAR) model [17] represents another approach that considers the process as a mixture of regressive models and is a generalized Gaussian mixture transition distribution. It can handle non-stationary cycles and conditional heteroscedasticity and is often solved by the expectation and maximization (EM) method.

Various adaptive neural networks have been adopted to extend linear regressive models such as multilayer perceptron (MLP), radial basis functions (RBFs), support vector machines (SVM) and recurrent networks [9]. Nonstationarity implies that the time series change their dynamics in different time regions. It is unreasonable for a single model to capture the dynamics of the entire series. A potential solution is to use a mixture model approach to divide the entire model into several smaller ones. Then regression and prediction are made by the local models. The self-organizing map (SOM) can be used to partition time series. For instance, Dablemont et al. [6] applied SOM-based local models with RBF networks as regressors. Cao [4] used SVM regressors on SOM-clustered local segments. However, these models are two-stage modeling. Both clustering and local modeling may not be jointly optimized.

There were two early approaches to analyzing temporal signals or sequences with the SOM. One is to train a SOM on static states (i.e. time series values), and then temporal patterns or sequences of states can be identified by marking sequential locations of the state on the trained map. Such approaches can be used to monitor dynamic processes or trajectories of a temporal process such as industrial plants [1]. Another approach, which is often found in the literature, is to group consecutive time points into segments (using a sliding window). Then these segments are used as the input vectors to train the SOM. We term this method as vector SOM or simply SOM.

Several variants have since been proposed to extend SOM's ability for temporal modeling such as the recurrent SOM (RSOM) [10] and the recursive SOM (RecSOM) [16,15]. These 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. Neural gas (NG) [12] is another variant of SOM. Instead of having a fixed network topology throughout, NG can dynamically deploy its resources to suit varying topology of the data and has been applied to tasks including temporal modeling [12] and has been enhanced by merge NG (MNG) [15].

Earlier, Lampinen and Oja proposed a self-organizing map of spatial and temporal AR models [11], where each unit represents an AR model with its reference vector as the model parameters. The method in fact is a multiple AR model with the component models forming a spatial topology. However, the model has difficulties to converge to the underlying regressive models due to the simple error-based similarity measure. We have extended it to a mixture regressive model, termed the self-organizing mixture autoregressive (SOMAR) model [13,14], with a different partition mechanism and similarity measure to reflect the characteristics of homogeneous time series. Both the mixture and local models are jointly trained, and thus it offers better modeling performance [13,14]. Here the SOMAR model is further analyzed in light of the MAR model and generalized to a full mixture model.

The remainders of the paper are as follows. Section 2 briefly describes various regressive models. Section 3 presents SOM-based autoregressive models and the proposed generalized SOMAR model, followed by experimental results on both benchmark data and real-world data and comparisons with several methods in Section 4. Finally, conclusions are given in Section 5.

2 Regressive Time Series Models

2.1 Autoregressive Models: AR, ARMA, GARCH and ARIMA

Linear regressive models have been the primary tool in modeling time series. An autoregressive model of order *p*, denoted as AR(*p*), can be described as,

$$x_t = c + \sum_{i=1}^{p} \phi_i x_{t-i} + \varepsilon_t = c + \Phi^T \mathbf{x}_{t-1}^{(p)} + \varepsilon_t \qquad (1)$$

where $\Phi = [\phi_1, ..., \phi_p]^T$ are the parameters, $\mathbf{x}_{t-1}^{(p)} = [x_{t-1}, ..., x_{t-p}]^T$ is the concatenated input vector, *c* is a constant and ε is white noise of zero mean and variance σ^2.

An ARMA model with *p*-order AR terms and *q*-order MA terms is called ARMA(*p*, *q*) model and can be written as,

$$x_t = c + \sum_{i=1}^{p} \phi_i x_{t-i} + \sum_{i=0}^{q} \mu_i \varepsilon_{t-1} \qquad (2)$$

where $\{\mu_0, ..., \mu_q\}$ are the parameters of the moving average. The error terms are assumed to be independent identically-distributed (i.i.d.) random variables sampled from a normal distribution with zero mean and variance σ^2. When this condition does not hold, the GARCH model provides a generalized alternative, in which the variance of the error terms is modeled by another regressive model.

A standard GARCH(θ; *q*) model is characterized by Eq. (1) and the following variance model,

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^{\theta} \alpha_i \varepsilon_{t-i}^2 + \sum_{i=0}^{\rho} \beta_i \sigma_{t-i}^2 \qquad (3)$$

where ε_t is the error term with the assumption $\varepsilon_t = \sigma_t v_t$ and v_t is i.i.d. with zero mean and unit variance. $\{\alpha\}$ and $\{\beta\}$ are the model parameters of the variance.

ARIMA model uses lags or differencing of the time series in the ARMA model. ARIMA(*p*,*d*,*q*) model is characterized by the following equation,

$$(1 - \sum_{i=1}^{p} \phi_i L^i)(1 - L)^d x_t = (1 + \sum_{i=1}^{q} \mu_i L^i)\varepsilon_t \qquad (4)$$

where *L* is the lag operator, i.e. $Lx_t = x_{t-1}$ and *p*, *d*, and *q* are the orders of the autoregressive, integrated, and moving average parts of the model respectively. Note that ARMA(*p*,*q*), i.e. Eq. (1), can be expressed as $(1 - \sum_{i=1}^{p} \phi_i L^i) x_t = (1 + \sum_{i=1}^{q} \mu_i L^i)\varepsilon_t$.

As can be seen, the ARIMA model operates on the difference of the lagged time series. Such simple transformation can be effective in dealing with slow changes in non-stationarity. That is, the difference operator transforms a slow drift non-stationary process into a stationary process.

2.2 Mixture Autoregressive (MAR) Model

A nonlinear or non-stationary time series can be regarded as a mixture of stationary processes characterized by the standard autoregressive models. The K-component MAR model is defined by [17],

$$F(x_t \mid \Gamma_{t-1}) = \sum_{k=1}^{K} \pi_k \varphi(\frac{x_t - \phi_{k0} - \phi_{k1}x_{t-1} - \ldots - \phi_{kp_k} x_{t-p_k}}{\sigma_k}) \qquad (5)$$

where $F(x_t \mid \Gamma_{t-1})$ is the conditional distribution of x_t given the past information up to t-1, Γ_{t-1}; $\varphi(.)$ is the standard normal distribution; $\{\pi_1,\ldots,\pi_K\}$ are the mixing parameters and $\pi_1+\ldots+\pi_K=1$, $\pi_k>0$, $k=1,\ldots, K$; p_k is the order of the k-th AR model; and σ_k^2 is the variance of the k-th distribution. This model is denoted as MAR(K; $p_1,\ldots,p_K$) model. The MAR has the ability to handle cycles and conditional heteroscedasticity in time series and its parameters are estimated via the EM algorithm and model selection by a Bayesian information criterion (BIC) [17].

3 Self-Organizing Mixture Autoregressive Models

3.1 Self-Organizing AR (SOAR) Model

Lampinen and Oja proposed a self-organizing AR (SOAR) network [11]. It is a map of neurons, each representing an AR model with its parameters as the reference vector $\mathbf{w}_i$. The experiment showed that the SOAR model can learn to distinguish texture images [11]. The method in fact is a multiple AR model. However the model has difficulties in converging to correct AR models. The training procedure is:

1) At each time step t, find the best matching unit by measuring the estimation error of each node, $e_{i,t} = x_t - \mathbf{w}_i^T \mathbf{x}_{t-1}^{(p)}$. In order to reduce the effect of the fluctuation or noise in the errors, an exponential average over the recent errors is used,

$$u_{i,t} = \lambda e_{i,t} + (1-\lambda)u_{i,t-1} \qquad (6)$$

where λ is a smoothing factor, $e_i(t)$ is the current error of node i and $u_i(t$-1) is the past averaged error.

2) Update the best matching unit as well as its neighborhood on the map by the recursive LMS or Widrow-Hoff rule,

$$\mathbf{w}_{i,t} = \mathbf{w}_{i,t-1} + \eta h(v,i) e_{i,t} \mathbf{x}_t^{(p)} \qquad (7)$$

where η is learning rate and $h(i,v)$ is the neighborhood function of indexes of node i and winner v.

However the performance of the SOAR model in finding the underlying AR processes in the mixture is poor [13]. Due to the stochastic nature of AR processes, although the overall MSE decreases, at each input, one can always expect large fluctuation even when the true model parameters are used and further smoothing is applied. In other words, this method has difficulties in converging to the true model parameters of the underlying AR processes. Nevertheless, the SOAR model localizes the time series by local models.

3.2 Self-Organizing MAR (SOMAR) Model

Based on the similar principle and the MAR model, the self-organizing MAR (SOMAR) was proposed [13]. It constitutes a simplified MAR model with the winner-take-all for local AR models. To ensure a robust learning, a new winner selection or similarity measure was proposed. A stochastic process is characterized by white noise residuals. As a sufficient condition, the modeling errors or the residuals should be or close to white noise if the modeling is following the correct path. Therefore, the autocorrelation of the error instead of the error itself is used to evaluate the similarity between the input vector and the neurons' weights representing the model parameters. To estimate the autocorrelation, a small batch of the errors is used,

$$R_{i,t}(k) = \frac{1}{m\sigma_e^2} \sum_{l=0}^{m-p-1} (e_{i,t-l} - \mu_e)(e_{i,t+k-l} - \mu_e) \tag{8}$$

where m is the length of the batch, μ_e and σ_e^2 are the mean and the variance of the errors in the batch respectively.

The winner is selected according to the sum of (absolute value of) autocorrelation coefficients (SAC),

$$v = \arg\min_i \left(\sum_{k=-m}^{m} \left| R_{i,t}(k) \right| \right) \tag{9}$$

The use of correlation measure for identifying local models is justified by the fact that a correct model produces white noise residuals. That is, if the model is correct or adequate, the residual is unpredictable or structure-less. Such effective correlation-based tests are often used in statistics and neural networks for checking the fitness of a model, e.g. [5], though there are other whiteness tests in the literature.

3.3 Generalized SOMAR (GSOMAR) Model

Both the SOMAR and SOAR models represent a simplified, homescedastic and winner-take-all version of the MAR model. At any time, only one local AR model (the winner) is selected to represent the time series, all models are assumed of equal variance and the mixing factors are either unit for the winner or zero otherwise. Although some empirical use of neighboring nodes has been proposed for forecasting [13], the model is not a full mixture model. To fully employ the mixture model, all components will be required to contribute to the mixture coherently both in training and testing. The mixing factors and model variances have to be learnt as well. The SOM has been extended before to a mixture model. The self-organizing mixture network (SOMN) [18] is such an example, in which each node represents a conditional distribution. The SOMN has also been shown to converge faster and be more robust than the EM algorithm for heteroscedastic mixture distributions. To make the SOMAR a full mixture of AR models, the algorithm of the SOMN can be used to learn the mixing factors and variances. In addition to the weights (or AR model parameters), the mixing factors and model variances are updated in the training (modeling). Further assuming that the component models are uncorrelated, so their covariances are zeros. The variances of local models are scalar. Then the updating rules for the mixing weights and variances have the following simple forms,

$$\pi_{i,t} = \pi_{i,t-1} + \eta(\hat{P}_i - \pi_{i,t-1}) \tag{10}$$

$$\sigma_{i,t}^2 = \sigma_{i,t-1}^2 + \eta(\sigma_{i,e}^2 - \sigma_{i,t-1}^2) \tag{11}$$

where $\hat{P}_i$ is the winning frequency and $\sigma_{i,e}^2$ is the error variance of node *i*.

The trained mixture model, representing the MAR model, Eq. (5), can be fully used for forecasting the time series as well as model's volatility. In forecasting, the learnt mixing factors are further weighted by the neighborhood function of the SOM, acts as the posterior probability of a component class given an input sample [18].

4 Experimental Results and Comparisons

4.1 Artificial Data

As an illustrative example, a mixture two AR(2) processes was generated with their model parameters set to [0.2, -0.3] and [0.4, -0.1] and variances to 3 and 5, respectively. The learning process of the GSOMAR is shown in Fig. 1.

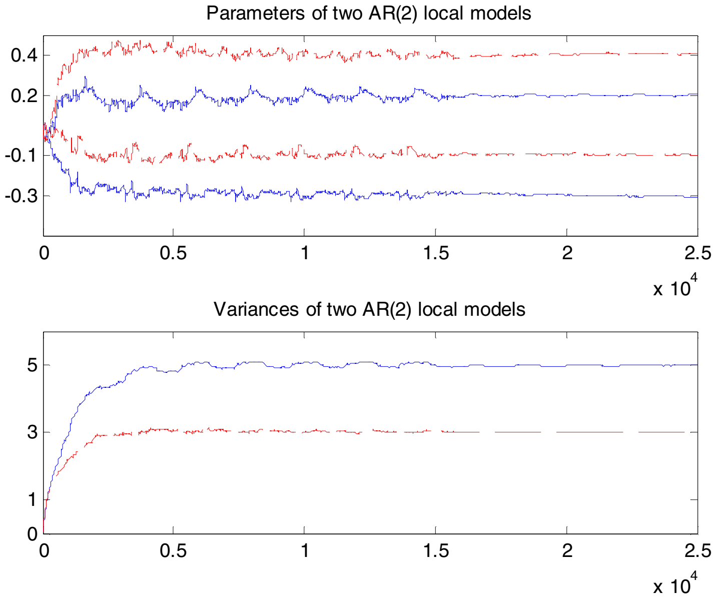

Fig. 1. Parameter and variance estimation of a mixture of two AR(2) processes. Fine tuning phase [13] starts at t=15000. Dashed (red) lines represent one process and solid (blue) lines the other.

4.2 Mackey-Glass Data

The Mackey-Glass series has been widely used as a benchmark data for testing nonlinear models. The data set was generated by a dynamic system defined by the following differential equation,

$$\frac{dx}{dt} = \beta x_t + \frac{\alpha x_{t-\delta}}{1 + x_{t-\delta}^{10}}$$

with the parameter values set as δ=17, α=0.2, and β= -0.1. In total 2000 points were generated. The Mackey-Glass data is regarded as consisting of a number of unknown AR processes. In the experiment, the series was grouped into 12 consecutive values as the input vectors. The order of the AR processes was chosen by the BIC. The prediction result is shown in Table 1. The performance of statistical benchmark models GARCH and ARIMA are 4.48 and 4.35 respectively. The results are the average over 10 independent runs on the same data set. Different data set may lead to slightly difference performance. However, it can be seen that GSOMAR and SOMAR markedly outperform the others and GSOMAR further improves on SOMAR.

Table 1. Forecasting performance on Mackey-Glass data by various adaptive models

	GSOMAR	SOMAR	SOAR	SOM	RSOM	RecSOM	NeualGas	MNG	SOM+ SVM
MSE($^{-2}$)	**3.24**	3.62	4.29	4.48	4.32	4.10	4.38	4.35	4.52

4.3 Foreign Exchange Rates

The data was obtained from the PACIFIC Exchange Rate Service provided by W. Antwiler at UBCs Sauder School of Business. It consists of 15years' daily exchange rates (British pound vs. US dollar, Euro and HK dollar) excluding weekends and bank holidays when the currency markets were closed. In total 3200 consecutive points were used, in which the first 3,000 points were used as the training set, the next 100 points as the validation set, and the remaining 100 points as the test set. The training, validation and testing sets were windowed with the length of 15 points to form input vectors (again validated by the BIC).

To compare with other regressive models, the following commonly used performance measures have been calculated:

Predicted return (*%*): The percentage of correct prediction of the return ($\ln x_{t+1} / x_t$), which is also used as a criterion to check whether the prediction is made in the right direction. In other words, it shows how many percentages of the predicted returns have the same signs as their corresponding actual returns.

MSE of predicted rate ($^{-2}$): The MSE between the actual exchange rates and the predicted ones in the test set.

Accumulated profit (*P%*): The accumulated profit is the percentage gain of the accumulated profits over the testing period, say 100 trading days.

Table 2. Performance on FX rate prediction by various adaptive models. The best performances are marked in bold.

GBP vs	GSOMAR	SOMAR	SOAR	SOM	RSOM	RecSOM	NeualGas	MNG	SOM+ SVM
USD %	59.70	**59.73**	52.84	52.63	52.26	52.58	54.08	54.16	53.43
USD $^{-2}$	3.87	**3.80**	4.28	4.20	4.24	4.70	4.23	4.20	4.12
USD *P%*	**5.53**	5.15	4.78	4.80	4.98	5.12	5.33	5.35	4.82
EU %	**57.43**	56.42	52.62	52.12	53.05	53.17	54.24	54.27	54.09
EU $^{-2}$	**3.96**	4.11	4.73	4.32	4.64	4.95	4.51	4.50	4.62
EU *P%*	**5.41**	5.12	4.62	4.73	4.63	4.60	4.72	4.75	4.70
JPY %	**57.95**	57.30	53.22	54.29	52.48	52.33	53.46	53.47	52.10
JPY $^{-2}$	**4.23**	4.33	5.24	5.00	4.98	5.08	4.75	4.75	5.18
JPY *P%*	**5.32**	5.03	4.68	4.89	4.91	4.87	4.73	4.76	4.65
HKD %	**56.37**	56.31	53.50	53.95	53.88	54.02	54.21	54.22	54.13
HKD $^{-2}$	**4.11**	4.22	4.67	4.75	4.73	4.72	4.44	4.44	4.57
HKD*P%*	**5.32**	5.02	4.50	4.59	4.57	4.63	4.62	4.68	4.60

As reported before [13,14], the SOMAR model generally outperforms other adaptive methods as also shown in Table 2. The GSOMAR further improves on the SOMAR model in all these performance measures. As can be seen, both GSOMAR and SOMAR consistently outperform other methods by clear margins in the correct prediction percentages and modeling errors. The benefit of using the fuller GSOMAR model is that model variance parameters are readily available to indicate the volatility of the component regressive models and the mixture. Statistical model ARIMA performed the worse on these data sets with the *predicted return* (%) between 50-51% – only slightly better than random guess; while GARCH gave similar performances to SOM+SVM with the *predicted return* (%) between 53-54%.

5 Conclusions

A mixture model approach to tackling nonlinear and non-stationary time series has been proposed by using the generalized self-organizing mixture autoregressive (GSOMAR) model. It consists of a number of autoregressive models that are organized and learnt in a self-organized manner by the adaptive LMS algorithm. A correlation-based similarity measure is used for identifying correct AR models, thus making the model more effective and robust compared to the error-based measures. The GSOMAR further generalizes the winner-take-all SOMAR model by learning the mixing weights as well as the model variances. The experiments on various nonlinear, non-stationary time series show that the proposed model can correctly detect and uncover underlying regressive models. The results also show that the proposed method outperforms other methods in terms of modeling errors and prediction performances.

References

1. Allinson, N.M., Yin, H.: Interactive and semantic data visualization using self-organizing maps. In: Proc. IEE Colloquium on Neural Networks in Interactive Multimedia Systems (1998)
2. Bollerslev, T.: Generalized autoregressive conditional heteroskedasticity. Journal of Econometrics 31, 307–327 (1986)
3. Box, G., Jenkins, G.: Time Series Analysis: Forecasting and Control. Holden-Day, San Francisco (1970)
4. Cao, L.J.: Support vector machines experts for time series forecasting. Neurocomputing 51, 321–339 (2002)
5. Chen, S., Billings, S.A., Cowen, C.F.N., Grant, P.M.: Practical identification of NARMAX models using radial basis functions. Int. Journal of Control 52, 1327–1350 (1990)
6. Dablemont, S., Simon, G., Lendasse, A., Ruttiens, A., Blayo, F., Verleysen, M.: Time series forecasting with SOM and local non-linear models – Application to the DAX30 index prediction. In: Proc. of WSOM 2003, pp. 340–345 (2003)
7. Kohonen, T.: Self-Organizing Maps. Springer, Heidelberg (1997)
8. Enders, W.: Applied Econometric Time Series, 2nd edn. John Wiley & Sons, Chichester (2004)
9. Haykin, S.: Neural Networks – A Comprehensive Foundation, 2nd edn. Prentice-Hall, Englewood Cliffs (1998)
10. Koskela, T.: Time Series Prediction Using Recurrent SOM with Local Linear Models. Helsinki University of Technology (2001)
11. Lampinen, J., Oja, E.: Self-organizing maps for spatial and temporal AR models. In: Proc. of 6th SCIA Scandinavian Conference on Image Analysis, Helsinki, Finland, pp. 120–127 (1989)
12. Martinetz, T., Berkovich, S., Schulten, K.: Neural-gas network for vector quantization and its application to time-series prediction. IEEE Trans. Neural Networks 4, 558–569 (1993)
13. Ni, H., Yin, H.: Time-series prediction using self-organizing mixture autoregressive network. In: Yin, H., Tino, P., Corchado, E., Byrne, W., Yao, X. (eds.) IDEAL 2007. LNCS, vol. 4881, pp. 1000–1009. Springer, Heidelberg (2007)
14. Ni, H., Yin, H.: Self-organizing mixture autoregressive model for non-stationary time series modeling. International Journal of Neural Systems 18, 469–480 (2008)
15. Strickert, M., Hammer, B.: Merge SOM for temporal data. Neurocomputing 64, 39–72 (2005)
16. Voegtlin, T., Dominey, P.F.: Recursive self-organizing maps. Neural Networks 15, 979–991 (2002)
17. Wong, C.S., Li, W.K.: On a mixture autoregressive model. Journal of the Royal Statistical Society, Series B (Statistical Methodology), Part1 62, 95–115 (2000)
18. Yin, H., Allinson, N.M.: Self-organizing mixture networks for probability density estimation. IEEE Trans. on Neural Networks 12, 405–411 (2001)