Locally Weighted Learning for ARMA Time Series

إستخدام أسلوب التعلم المحلي المرجح للتنبؤ بالسلاسل الزمنية

Dr. Gamal A. M. Al-Shawadfi and Dr. Hindi A. Al-Hindi
College of Business and Economics, King Saud University,
Al-Qasseem Branch, Al-Melaida
Saudia Arabia

الخلاصة: يقدم هذا البحث إستخدام أسلوب التعلم المحلي المرجح للتنبؤ بالسلاسل الزمنية التي تتــبع نماذج ARMA(*p,q*). ويهدف البحث إلى التعرف على إمكانيات هذا الأسلوب في التنبؤ بالسلاســل الزمنية. وقد تم عشوائيا توليد عينات مقابلة لنماذج عديدة من السلاسل الزمنية وتم تقســيم هــذه العينات إلى مجموعتين. المجموعة الأولى أستخدمت لتقدير معالم أسلوب التعلم المحلي المرجح في حين أن المجموعة الثانية أستخدمت لتقييم أداؤه. وتمت مقارنة نتائج أسلوب الــتعلم المحلي المرجح مع النتائج التي تم الحصول عليها من إستخدام نماذج بوكس وجينكنز لــتقدير السلاسل الزمنية. وأوضحت النتائج أن أسلوب التعلم المحلي المرجح تفوق على نماذج بوكس وجينكنز بناء على المعايير المستخدمة لتقييم الأسلوبين وهي متوسط الإنحرافات المربعة ومتوســط الإنحرافات المطلقة ونسبة البيانات المقدرة بإستخدام أسلوب التعلم المحلي المرجح والتي تكون أقرب إلى البيانات الفعلية من تقديرات نماذج بوكس وجينكنز.

Abstract

This paper deals with the application of locally weighted learning for forecasting time series corresponding to a wide range of ARMA(*p,q*) models. The objective of this paper is to explore the feasibility of locally weighted learning in time series forecasting. The study adopted a simulation approach to generate random samples corresponding to different time series models. The samples were divided into two sets: training and test sets. The training set was used to estimate the parameters of the locally weighted learning whereas the test set was used to test its performance. The results of the locally weighted learning were compared to those obtained from using Box-Jenkins modeling approach. The results of the study show that locally weighted learning outperforms Box-Jenkins modeling approach based on the criteria used which are mean squared error (MSE), mean absolute error (MAE) and ratio of the estimated data points closer to actual data points (Ratio).

Accepted May 24, 2003.

1. Introduction

Box and Jenkins [1] introduced a forecasting approach for autoregressive moving average (ARMA) models. Since their introduction, ARMA models have been used to model time series data in a wide range of fields. In recent years, several new modeling approaches for time series have emerged and the Box Jenkins models have been used as benchmarks to evaluate the new approaches [2]. The new approaches suitable for time series modeling include artificial neural networks (ANNs) [3][4][5], support vector machines (SVMs) [6] and locally weighted learning (LWL) [7]. Artificial neural networks received much attention in the area of forecasting and several studies proved their superiority in time series forecasting.

The locally weighted learning [7][8] is a memory-based technique that performs a prediction when a query is received. The prediction is made using the neighboring examples of the query, which are considered relevant according to a distance measure. It involves storing the training data in memory and making a prediction on a query by query basis where data points near the query point are given higher weights [8]. In most learning methods, a single global model is used to fit all the training data. On the other hand, locally weighted learning attempts to fit the training data only in a region around the location of the query. This results in eliminating the impact of irrelevant data points on forecasts.

The use of locally weighted learning has increased in many areas. Atkeson et al. [9] discussed the application of locally weighted learning in robot control. Zografski [10] applied locally weighted learning for nonlinear modeling control and forecasting. Lawrence et al. [11] compared locally weighted learning and neural networks in function approximation where locally weighted learning performed better on half of the test problems. Lejeune and Sarda [12] applied locally weighted learning to estimate density functions. Atkeson and Schaal [13] combined an artificial neural network with locally weighted learning for representing nonlinear functions used in robot control. More and Schneider [14] used locally weighted learning in stochastic optimization on randomly generated functions and a simulated manufacturing task. Wang and Zucker [15] used locally weighted learning to solve the multiple-instance problem where they found that this method is competitive with the best existing methods used to solve this problem. These applications and many more are indications of the increasing popularity of this method in solving a wide range of problems.

This study is designed to investigate locally weighted learning to forecasting time series corresponding to a wide range of ARMA(p,q) structures. Training and test samples, for various ARMA(p,q) structures, were randomly generated to estimate the parameters of the locally weighted learning model and evaluate its performance. The rest of this paper is organized in the following manner. Section 2 presents a brief overview of ARMA models. Section 3 describes the locally weighted learning approach. Section 4 describes the procedure used to generate the data and the measures used to evaluate the performance of the proposed approach. Section 5 discusses the results of the study. Section 6 concludes this paper.

2. ARMA(p,q) models

A linear stochastic model whose input is white noise a_t can be expressed as follows [16][17]:

$$y_t = \mu + a_t + \varphi_1 a_{t-1} + \varphi_2 a_{t-2} + \ldots.. \quad (1)$$

where μ is the mean of a stationary process and φ_t , $t=1,2,\ldots$ are coefficients which satisfy $\sum_{j=0}^{\infty}\varphi_j^2$, at is an uncorrelated random variables with means zero and variance σ_a^2. Equation (1) can be expressed in terms of a finite number of autoregressive (AR) and/or moving average components. Let $\tilde{y}_t = y_t - \mu$, an AR(p) process can be generally expressed as follows:

$$\tilde{y}_t = \phi_1 \tilde{y}_{t-1} + \phi_2 \tilde{y}_{t-2} + \cdots + \phi_p \tilde{y}_{t-p} + a_t \quad (2)$$

An MA(q) process can be expressed as follows:

$$\tilde{y}_t = a_t - \theta_1 a_{t-1} - \cdots - \theta_q a_{t-q} \quad (3)$$

A mixed ARMA(p,q) process can be expressed as:

$$\tilde{y}_t = \phi_1 \tilde{y}_{t-1} + \phi_2 \tilde{y}_{t-2} + \cdots + \phi_p \tilde{y}_{t-p} + a_t - \theta_1 a_{t-1} - \cdots - \theta_q a_{t-q} \quad (4)$$

As it is well known, Box and Jenkins [1] have presented the most popular procedure to analyze the ARMA models. The procedure consists of four phases: the identification, estimation, diagnostic checking and forecasting phases. The identification phase determines the initial number of autoregressive (AR) and moving average (MA) parameters using the autocorrelation and partial autocorrelation functions. The estimation phase is based on the maximum likelihood or nonlinear least squares estimates. The diagnostic checking phase examines the residuals to see the adequacy of the identified model. The last phase is the forecasting of future observations using their conditional expectations. Box and Jenkins procedure has been extensively explained by Harvey [16], Choi [18], Lutkepohl [19], Pankratz [20] and Wei [21].

The Yule-Walker equations may help in the identification and estimation phases of Box-Jenkins analysis. For example, Box and Jenkins [1] used the Yule Walker equations for obtaining initial estimates of the parameters of a time series model . The Yule Walker system of equations is:

$$\begin{aligned}
\rho_1 &= \Phi_1 + \Phi_2\rho_1 + \Phi_3\rho_2 + \cdots + \Phi_p\rho_{p-1} \\
\rho_2 &= \Phi_1\rho_1 + \Phi_2 + \Phi_3\rho_1 + \cdots + \Phi_p\rho_{p-2} \\
&\cdots \\
\rho_p &= \Phi_1\rho_{p-1} + \Phi_2\rho_{p-2} + \Phi_3\rho_{p-3} + \cdots + \Phi_p
\end{aligned} \tag{5}$$

where ρ_k represents the autocorrelation and Φ_k are the model parameters. $k = 1, 2, \cdots, p$.

Replacing the theoretical autocorrelations ρ_k by the estimated autocorrelations r_k and solving the above system of equations, we obtain the Yule-Walker estimators:

$$\Phi = R_p^{-1} r_p \tag{6}$$

where $\Phi = [\Phi_1\ \Phi_2\ \cdots\ \Phi_p]'$ is the vector of estimated parameters, R is an estimate of the $p \times p$ matrix of correlations up to *p-1* and *r* is the vector $[\rho_1\ \rho_2\ \cdots\ \rho_p]'$.

The Yule-Walker estimator has the same asymptotic properties as the least squares estimator. However it may have less attractive small sample properties . Therefore the least squares estimator is usually used in practice, see Lutkepohl [19].

3. Locally Weighted Learning

Given input variables $x \in R^n$ and an output variable $y \in R$, the following mapping $f : R^m \to R$ is considered of which there is a set of n examples $\{(x_i, y_i)\}_{i=1}^n$ generated from the following model [7][9][22]:

$$y_i = f(x_i) + \varepsilon_i \tag{7}$$

where ε_i is a random variable such that $E(\varepsilon_i) = 0$ and $E(\varepsilon_i\varepsilon_j) = 0$ and $E(\varepsilon_i^m) = \mu_m(x_i)$, $\forall m \geq 2$. The goal is to estimate the value of $f(x)$ for a given query point x_q using information pertaining only to a neighborhood of x_q. The parameter β of the linear approximation of $f(\cdot)$ in a neighborhood of x_q can be obtained by solving the following local polynomial regression [22]:

$$\sum_{i=1}^{n}\left\{(y_i - x_i'\beta)^2 K\left(\frac{d(x_i, x_q)}{h}\right)\right\} \tag{8}$$

where $d(x_i, x_q)$ is the distance from the query point x_q to x_i, $K(\cdot)$ is a weight function, h is the bandwidth and a constant value 1 has been added to each vector x_i in order to consider a constant term in the regression function. The solution to the problem is given by [7][9]:

$$\hat{\beta} = (X'W'WX)^{-1}W'X'Wy = (Z'Z)^{-1}Z'Wy$$
$$\hat{\beta} = PZ'Wy \qquad (9)$$

where W is a diagonal matrix whose elements are $w_{ii} = \sqrt{K(d(x_i, x_q)/h}$, and the matrix $X'W'WX = Z'Z$ is assumed to be non-singular so that its inverse $P = (ZZ')^{-1}$ is defined.

A leave-one-out cross validation (CV) estimation of the error variance $E[(\hat{y}_q - y_q)^2]$ can be easily obtained using the PRESS statistic [22], by calculating the error $e_j^{cv} = y_j - x_j'\hat{\beta}_{-j}$, without the need to calculate the parameters $\hat{\beta}_{-j}$ from the examples available with the j^{th} observation removed as follows [22]:

$$e_j^{cv} = y_j - x_j'\hat{\beta}_{-j} = \frac{y_t - x_j(Z'Z)^{-1}Z'Wy}{1 - z_j'(Z'Z)^{-1}z_j} = \frac{y_j - x_j\hat{\beta}}{1 - h_{jj}} \qquad (10)$$

where z_j' is the j^{th} row of Z and therefore $z_j = w_{jj}x_j$, and where h_{jj} is the j^{th} diagonal element of the matrix $H = Z(Z'Z)^{-1}Z'$.

The solution to the problem in (6) depends on the selection of the weight function $K(\cdot)$. The widely used weight function takes the following form [22]:

$$K\left(\frac{d(x_i, x_q)}{h}\right) = \begin{cases} 1 & if\, d(x_i, x_q) \le h \\ 0 & otherwisw \end{cases} \qquad (11)$$

To estimate the local regression model, we need to find the optimal number k of neighbors used in prediction. This is achieved by deriving k number of local models each with different number of neighbors and selecting the best model. The parameter $\hat{\beta}(k)$ of the model obtained using the k nearest neighbors is updated using the standard steps of recursive least squares algorithm as follows [9][22]:

$$\begin{cases} P(k+1) = P(k) - \dfrac{P(k)x(k+1)x'(k+1)P(k)}{1 + x'(k+1)P(k)x(k+1)} \\ \gamma(k+1) = P(k+1)x(k+1) \\ e(k+1) = y(k+1) - x'(k+1)\hat{\beta}(k) \\ \hat{\beta}(k+1) = \hat{\beta}(k) + \gamma(k+1)e(k+1) \end{cases} \qquad (12)$$

with $P(k) = (Z'Z)^{-1}$ and where $x(k+1)$ is the $(k+1)^{th}$ nearest neighbor of the query point. After obtaining the matrix $P(k+1)$, the cross validation (CV) errors can be calculated as follows [9][22]:

$$e_j^{cv}(k+1) = \frac{y_j - x'_j\hat{\beta}(k+1)}{1 - x'_j P(k+1) x_j}, \qquad \forall j : d(x_j, x_q) \le h(k+1) \qquad (13)$$

For a given query point x_q, a set of predictions $\hat{y}_q(k)$ each associated with a parameter estimate $\hat{\beta}(k)$ and an error vector $e^{cv}(k)$ derived using the k number of neighborhood points. The mean square error for each model is [22]:

$$MSE = \frac{\sum_{i=1}^{k} w_i (e_i^{cv}(k))^2}{\sum_{i=1}^{k} w_i} \qquad (14)$$

The prediction $\hat{y}_q$ is obtained by comparing the predictions obtained for each value k and selecting the one with lowest mean square errors as follows [9][22]:

$$\hat{y}_q = x'_q\hat{\beta}(\hat{k}), \qquad with \qquad \hat{k} = arg\,\min_k MSE(k) \qquad (15)$$

4. Experiment

To investigate the potential of locally weighted learning to modeling time series, simulated time series, generated from a wide range of coefficient values, were used in the study. Coefficient values were chosen from various sub-regions of the parameter space that satisfy the stationarity and invertability conditions of the generated time series. Each of these sub-regions represents a structure with similar autocorrelation functions and partial autocorrelation functions. The selected coefficient values ensure the extensive coverage of the parameter space.

The experiment comprises two stages, namely LWL training and LWL testing. Eight different ARMA(p,q) structures were considered each with different p and q values which are {(1,0), (2,0), (0,1), (0,2), (1,1), (1,2), (2,1), (2,2)}. Within each structure, four different values of the structure parameters were selected. For example, in case ARMA(1,0), the following values for ϕ_1 were selected {ϕ_1=0.3, ϕ_1 =0.5, ϕ_1=0.7, ϕ_1=0.9}. This resulted in a total of 32 models with each model defined in terms of its parameter set. For each coefficient set corresponding to an ARMA(p,q) model, 400 samples were randomly generated with each sample contains 20 data points. The first 200 samples were used for training the locally weighted learning model and the second 200 samples were used for testing its performance. The total number of samples is 12400. The locally weighted learning models were estimated using the Lazy Learning Toolbox [23][24], for MATLAB®. Three evaluation measures were used to evaluate the performance of the LWL model. The first is the mean squared error (MSE):

$$MSE = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n} \tag{16}$$

The second measure is the mean absolute error (MAE):

$$MAE = \frac{\sum_{i=1}^{n}| y_i - \hat{y}_i |}{n} \tag{17}$$

The third measure is the Ratio:

$$Ratio = \frac{r}{n} \tag{18}$$

where,
y_i : actual values
$\hat{y}_i$: forecasted values
n : sample size.
r : the number of points where LWL forecasts are closer to actual values than B-J forecasts.

To assess the performance of the proposed locally weighted learning, the results were benchmarked against those obtained using by the Box-Jenkins approach. Each of the training data sets used to train the LWL model was used again to estimate the parameters of the corresponding ARMA(p,q) model. The estimated ARMA model was then used for making forecasts using the data points in the test sample. Finally, the mean squared error, mean absolute error and Ratio were calculated for performance evaluation.

5. Results

Numerical results show that the proposed approach has a good performance in forecasting ARMA(p,q) models. A comparison of the MSE, MAE and Ratio for the locally weighted learning and the Box-Jenkins model is summarized in Table 1 and discussed below:

1. The mean square error (MSE) statistic measures the residual variance where smaller values for this statistic are better. The proposed LWL forecasting approach tends to have smaller MSE values compared to the Box-Jenkins method as can be seen in the

Table. The average value for the MSE for LWL forecasts equals to 0.1259, which is less than the MSE for Box-Jenkins models, which equals to 0.3076. Therefore, the LWL approach performs better than Box-Jenkins approach, as measured by this statistic. Notice that for all the eight models, MSE performance of the LWL is better than that of the B-J method. This may suggest that the LWL approach has some ability to forecast the behavior of the series in the future similar or better than B-J method.

Table 1: Moderate Behavior of LWL and B-J Forecasting Methods

Model	Method	MSE	MAE	RATIO
ARMA(1,0)	LWL	0.0926	0.2326	0.7065
	B-J	0.2408	0.3809	0.2935
ARMA(2,0)	LWL	0.1040	0.2570	0.7391
	B-J	0.2338	0.4091	0.2609
ARMA(0,1)	LWL	0.1729	0.3395	0.9130
	B-J	0.3729	0.5723	0.0870
ARMA(0,2)	LWL	0.1646	0.3243	0.8750
	B-J	0.3373	0.5382	0.1250
ARMA(1,1)	LWL	0.1483	0.3138	0.7880
	B-J	0.2716	0.4558	0.2120
ARMA(1,2)	LWL	0.1249	0.2865	0.7500
	B-J	0.3434	0.4738	0.2500
ARMA(2,1)	LWL	0.0965	0.2455	0.8804
	B-J	0.3640	0.4730	0.1196
ARMA(2,2)	LWL	0.1036	0.2617	0.8859
	B-J	0.2971	0.4966	0.1141
Average	LWL	0.1259	0.2826	0.8173
	B-J	0.3076	0.4750	0.1827

2. The MAE statistic measures the mean absolute error and is smaller for LWL forecasts for different ARMA(p,q) models than the corresponding Box-Jenkins models. On average, the MAE for the LWL approach is 0.2826 which is better than the MAE for the Box-Jenkins which is 0.4750.

3. The Ratio in Table 1 measures the percentage of data where the LWL forecasts closer to the actual values than Box-Jenkins forecasts. The Ratio values stay consistently better than that of Box-Jenkins models with average equals 81.73% as measured by this statistic. The average value of the Ratio for Box-Jenkins models is 0.1827.

4. When trying to arrive at an appropriate ARMA(p,q) forecasting model in Box-Jenkins method, the analyst makes decisions based on autocorrelation and partial autocorrelation plots. Because this process relies on human judgment to interpret the data, it can be slow and sometimes inaccurate. The analyst may turn to LWL approach as a quicker and more accurate alternative. LWL is an ideal approach for finding patterns in data without the need to rely on human judgment.

6. Conclusion

This study examined the feasibility of the locally weighted learning in time series forecasting. The study is based on a simulation approach in which training and test samples were randomly generated to train and test the performance of the proposed approach. In addition, the results obtained from the locally weighted learning were compared to those obtained from Box-Jenkins approach. The findings show that the locally weighted learning is able to obtain better results on all the three evaluation measures, which are MSE, MAE and Ratio. These findings added to the simplicity and objectivity of the locally weighted learning makes it a sound approach for time series forecasting.

References

1 Box, G. E. P. and Jenkins, G. M. (1976). *Time Series Analysis Forecasting and Control*. Holden Day, San Francisco, U.S.A.

2 Hwarng, H. B. and Ang, H. T. (2001), "A Simple Neural Network for ARMA(p,q) Time Series", *Omega*, Vol. 2, pp. 319-333.

3 Al-Shawadfi, Gamal A. (2002), "A Comparison Between Neural Network and Box-Jenkins Forecasting Techniques with Application to Real Data", *The Research Bulletin, Human Resources Development and Research Center*, College of Business and Economics , King Saud University P.O. Box 6033 – Al-Melaida, Saudi Arabia.

4 Swanson, Norman R. and White, Halbert (1997), "A Model Selection Approach to Real-Time Macroeconomic Forecasting Using Linear Models and Artificial Neural Networks", *Review of Economics and Statistics*, Vol. 79, pp. 540-50.

5 Zhang, G. P. (2001), "An investigation of Neural Networks for Linear Time Series Forecasting", *Computers & Operations Research*, Vol. 28, pp. 1183-1202.

6 Tay, Fransis E. H. and Cao, Lijuan (2001), "Application of Support Vector Machines in Financial Time Series Forecasting", *Omega*, Vol. 29, pp. 309-317.

7 Atkeson, C. G.; Moore, A. W. and Schaal, S. (1997) "Locally Weighted Learning", *Artificial Intelligence Review*, Vol. 11, No. (1-5), pp. 11-73.

8 Bottou, L. and Vapnik, V. N. (1992), "Local Learning Algorithms", *Neural Computation*, Vol. 4, No. 6, pp. 888-900.

9 Atkeson, C. G., Moore, A. W. and Schaal, S. (1997), "Locally Weighted Learning for Control", *Artificial Intelligence Review*, Vol. 11, pp. 75-113.

10 Zografski, Z. (1992) "Geometric and Neuromorphic Learning for Nonlinear Modeling, Control and Forecasting", In *proceedings of the 1992 IEEE International Symposium on Intelligent Control*, Glasgow, Scotland, pp. 158-163.

11 Lawrence, S., Tsoi, A. C. and Back, A. D. (1996), "Function Approximation with Neural Networks and Local Methods: Bias, Variance and Smoothness", In *Australian Conference on Neural Networks (ACNN)*, Canberra, Australia, pp. 16-21.

12 Lejeune, M. and Sarda, P. (1992), "Smooth Estimators of Distribution and Density Functions", *Computational Statistics and Data Analysis*, Vol. 14, pp. 457-471.

13 Atkeson, C. G. and S. Schaal (1995), "Memory-Based Neural Networks for Robot Learning", *Neurocomputing*, vol. 9, no. 3, pp. 243-269, 1995.

14 Moore, A. W. and Schneider, J. (1996), "Memory-Based Stochastic Optimization", In D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, editors, *Neural Information Processing Systems 8*. MIT Press, 1996.

15 Wang, J. and Zucker, J.-D. (2000), "Solving the Multiple Instance Problem: A Lazy Learning Approach", In *Proceedings of the 17th International Conference on Machine Learning*, San Francisco, CA, pp. 1119-1125.

16 Harvey, A.C. (1990). *The Econometric Analysis of Time Series*. Philip Allan Publishers Limited, Market Place, Deddincton Oxford Ox 545 E, Great Britain.

17 Tashman, I. J. and Leach, M. L. (1991), "Automatic Forecasting Software: A Survey and Evaluation", *International Journal of Forecasting*, Vol. 7, pp.209-230.

18 Choi, Byoung Seon (1992). *ARMA Model Identification*, Springer-Verlag, New York, U.S.A.

19 Lutkepohl, Helmut (1993). *Introduction to Multiple Time Series Analysis*. Second edition , Springer-Verlag , Berlin, Germany.

20 Pankratz, Alan (1983). *Forecasting with Univariate Box-Jenkins Models: Concepts and Cases*. John Wiley and Sons, New York, USA.

21 Wei, William W.S. (1990). *Time Series Analysis Univariate and Multivariate Methods*. Addison Wisely Publishing Company, Inc. Redwood City, U.S.A.

22 Birattari, M., Bontempi, G. and Bersini, H. (1999), "Lazy Learning Meets Recursive Least Squares Algorithm", In *Advances in Neural Information Processing Systems, 11*, M. S. Kearns, S. A. Solla, and D. A. Cohn, Eds. MIT Press, Cambridge, MA.

23 Bontempi G., Birattari, M. and Bersini, H. (1999), "Lazy Learning for Local Modeling and Control", *International Journal of Control*, Vol. 72, No. 7/8, pp. 643-658.

24 *The Lazy Learning Toolbox For Use with MATLAB®*. Available for download at: http:/iridia.ulb.ac.be/~lazy/