Bulletin of the *Transilvania* University of Braşov • Vol 6(55), No. 2 - 2013
Series III: Mathematics, Informatics, Physics, 117-124

A QUANTITATIVE COMPARISON OF MODELS FOR UNIVARIATE TIME SERIES FORECASTING

Adela SASU[1]

Abstract

ARIMA is a popular method to analyze stationary univariate time series data, and nowadays it is considered the standard method for time series forecasting. We experimentally show that two machine learning based approaches can be used for the forecasting of univariate time series. The experiments are made on ten public time series datasets and we report the results obtained by ARIMA, linear regression and multilayer perceptrons networks. The quantitative results show that linear regresion and multi layer perceptrons obtain more acccurate predictions than the ones produced by ARIMA.

2000 *Mathematics Subject Classification:* 62M10, 60G25, 91B84, 97R40.
Key words: univariate time series forecasting, ARIMA, linear regression, multilayer perceptron.

1 Introduction

A time series is a sequence of observations $y_1, \ldots, y_t$ generated sequentially in time. Examples of time series include financial time series (stocks, indices, rates, etc.), physically observed time series (sunspots, weather, etc.), and mathematical time series (Fibonacci sequence, integrals of differential equations, etc.). The characteristic property of a time series is the fact that the data are not generated independently, their variance varies in time, they are often governed by a trend, and they have cyclic components. Statistical procedures that suppose independent and identically distributed data are, therefore, excluded from the analysis of time series.

Time series forecasting, or time series prediction takes an existing series of data $y_1, \ldots, y_t$ and forecasts the $y_{t+1}, y_{t+2}, \ldots$ data values. The goal is to observe or model the existing data series to enable future unknown data values to be forecasted accurately.

In order to better reveal certain non-seasonal features, a seasonal adjustment (SA) step is applied on the original data, to improve the forecasting accuracy. The mechanics of seasonal adjustment involve breaking down a series into trend-cycle, seasonal, and irregular components [11]:

- trend cycle: level estimate for each month (quarter) derived from the surrounding year-or-two of observations.

[1]Faculty of Mathematics and Informatics, *Transilvania* University of Braşov, Romania, e-mail: asasu@unitbv.ro

- seasonal effects: effects that are reasonably stable in terms of annual timing, direction, and magnitude. Possible causes include natural factors (the weather), administrative measures (starting and ending dates of the school year), and social/cultural/religious traditions (fixed holidays such as Christmas). Effects associated with the dates of moving holidays like Easter are not seasonal in this sense, because they occur in different calendar months depending on the date of the holiday.

- irregular components: anything not included in the trend-cycle or the seasonal effects (or in estimated trading day or holiday effects). Their values are unpredictable with regard to timing, impact, and duration. They can arise from sampling error, non-sampling error, unseasonable weather, natural disasters, strikes, etc.

The non-seasonally adjusted (NSA) time series is a sequence of successive measurements of any miscellaneous news activity obtained at regular time intervals (such as every month or every quarter). Some of the methods used in time series forecasting benefit from SA, while others are rather insensitive to it.

While Autoregressive Integrated Moving Average (ARIMA) model is adopted as de facto standard in time series forecasting applications, one may also consider some popular models used for regression problems. In machine learning, linear models and artificial neural networks are popular choices for inferential mechanisms. This paper quantifies the performance of ARIMA, linear regression and multilayer perceptron approaches for time series forecasting, on 10 time series datasets [14]–[23]. We experimentally show that the following conclusion from [10] page 285 is at least questionable: "In conclusion, one can say with certainty that for the immediate and short-term prediction, the Box-Jenkins method is the most accurate of all methods using time series models."

The ARIMA forecasting is made using the IBM SPSS Statistics analysis package software [13]. The linear regression and multilayer perceptron implementations are the ones provided by Weka [5].

The structure of this paper is as follows: section 2 contains a brief description of the three forecasting models which we compared and the formula used for model assessment; section 3 presents the datasets used for the experiments and the results produced by the three models. Section 4 draws the conclusions of the paper and sketches future work.

2 Time series forecasting models

2.1 ARIMA

ARIMA processes are a class of stochastic processes used in the area of time series modeling. The application of the ARIMA methodology for the study of time series analysis is due to Box and Jenkins [3]. Let us consider $\{y_t\}_t = y_1, y_2, \ldots$ the observations at equally spaced times and let $\{a_t\}_t = a_1, a_2, \ldots$ be a white noise series consisting of independent identically distributed random variables whose distribution is approximately normal with mean zero and variance σ_a^2. Assume that $E(y_t) = \mu_y$ and for simplicity let us note $y_t - \mu_y = \tilde{y}_t$. Therefore $E(\tilde{y}_t) = 0$.

Consider the general ARMA(p, q) model [4]:

$$\tilde{y}_t = \phi_1 \tilde{y}_{t-1} + \cdots + \phi_p \tilde{y}_{t-p} + a_t - \theta_1 a_{t-1} - \cdots - \theta_q a_{t-q} \tag{1}$$

or

$$\phi(B)\tilde{y}_t = \theta(B)a_t \tag{2}$$

where,

$$\phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p$$

is the autoregression operator of order p, $\theta(B)$ is the moving average operator of order q

$$\theta(B) = 1 + \theta_1 B + \theta_2 B^2 + \cdots + \theta_q B^q$$

and B is the backward shift operator, $By_t = y_{t-1}$, $B^k y_t = y_{t-k}$.

The general ARIMA(p, d, q) model is defined as

$$\phi(B)(1 - B)^d \tilde{y}_t = \phi(B)\nabla^d \tilde{y}_t = \theta(B)a_t \tag{3}$$

where $\nabla = 1 - B$ is backward difference and $\nabla^d = (1 - B)^d$ is backward difference of order d.

A common assumption for many time series techniques is that the data are stationary. A stationary process has the property that the mean, variance and autocorrelation structure do not change over time. Stationarity can be defined in precise mathematical terms, but for our purpose we mean a flat looking series, without trend, constant variance over time, a constant autocorrelation structure over time and no periodic fluctuations. If the roots of polynomial $\phi(B)$ lie outside the unit circle, it may be shown that an ARMA(p, q) is stationary [10].

Many time series are not stationary. Most often, the series of first differences is stationary, *i.e.* $y_t - y_{t-1} = (1 - B)y_t$. If a series $\{y_t\}_t$ has to be differenced once to obtain stationarity, then the model corresponding to the original series is called an integrated ARMA model of order p, 1, q or an ARIMA$(p, 1, q)$. In practice, differencing on the first order is necessary, while second order differencing is rarely needed. If the original series $\{y_t\}_t$ is stationary, then it is not necessary to differentiate it. If the time series manifest a periodic fluctuation (seasonal pattern), then the general ARIMA model is defined as in [4]:

$$\Phi(B^s)\nabla_s^D y_t = \Theta(B^s)a_t \tag{4}$$

where s is the number of periods in a season. Let us note the seasonal ARIMA model ARIMA$(p, d, q)(P, D, Q)_s$, where P=number of seasonal autoregressive terms, D=number of seasonal differences, Q=number of seasonal moving average terms.

The implementation provided by IBM SPSS Statistics analysis package[2] version 21 was used for ARIMA. As ARIMA is delivered as an SPSS procedure at least from

[2] http://www-01.ibm.com/software/analytics/spss/

version 13 of this product[3], one can expect the current version to be quite error–free. The steps for producing the most appropriate ARIMA model are detailed in [12].

2.2 Weka for time series forecasting

Weka [5], which was initially conceived as a classification and regression tool, was further extended with forecasting capabilities[4]. The forecasting plugin allows the user to choose the model to be used for prediction, as in a regular classification or regression task. The user is allowed to specify the periodic attribute, or whether Weka should use lagged variables or overlay data. Through a feature selection step, Weka decides which of the initial features (in case of multivariate time series) and which derived features (*e.g.* lagged data) are to be used for prediction. The version of Weka used for this paper is 3.7.10, and the forecasting plugin is of version 1.0.14.

2.3 Linear regression

Linear regression is one of the most popular models for regression. The output provided by this model is computed as:

$$y(\mathbf{x}) = \mathbf{w}^T \cdot \mathbf{x} = \sum_{j=1}^{M} w_j x_j \tag{5}$$

Sometimes, a normally distributed difference ε between the predicted value $y(\mathbf{x})$ and the corresponding actual value t is considered for the sum in eq. (5), especially for further probabilistic treatments [9], [2]. The coefficients $\{w_j\}_{j=1,\ldots,M}$ are sought through least squares or iterative methods (*e.g.* gradient descent). The values $\{x_j\}_{j=1,\ldots,M}$ are called "features" or "attributes" and they are associated with the data to be processed.

For N data exemplars $\mathbf{x}^{(n)} = (x_1^{(n)}, \ldots, x_M^{(n)})$, $1 \leq n \leq N$ one can consider the matrix $\mathbf{X}$ defined as:

$$\mathbf{X} = \begin{pmatrix} x_1^{(1)} & \ldots & x_M^{(1)} \\ x_1^{(2)} & \ldots & x_M^{(2)} \\ \vdots & \ddots & \vdots \\ x_1^{(N)} & \ldots & x_M^{(N)} \end{pmatrix} \tag{6}$$

then the solution for which the least squared error is attained is:

$$\mathbf{w} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \tag{7}$$

Here, $\mathbf{y}$ is the column vector of the N actual values corresponding to the N data exemplars, to whom the predicted values $y(\mathbf{x})$ are compared.

[3]See for example the manual http://brain.mcmaster.ca/SPSS.manual/SPSS%20Trends%2013.0.pdf, for SPSS 13, published in 2004.

[4]http://wiki.pentaho.com/display/DATAMINING/Time+Series+Analysis+and+Forecasting+with+Weka

2.4 Multilayer perceptron

The multilayer perceptron are graphical models for function approximation, based on nonlinear transformations on the input data. They use function composition and linear combinations of nonlinear differentiable functions and they are proved universal approximators [8]. By gradient descent-based methods the coefficients are adjusted in order to decrease a loss function, which measures the discrepancy between the values produced by the neural network and the actual values [7]; some second order approaches for numerical optimization of loss function were also developed [7]. During training, some parameters like learning rate and momentum coefficient determine the magnitude and direction of change for the coefficients. To avoid large values for the coefficients, one may include the weight decay strategy, which acts as a regularization mechanism [6].

2.5 Model assessment

The predicted outputs were compared to the actual values by the accuracy function MAPE:

$$MAPE = \frac{1}{l}\sum_{t=1}^{l}\frac{|Y_t - \hat{Y}_t|}{|Y_t|} \cdot 100,\ (Y_t \neq 0), \tag{8}$$

where Y_t is the actual value, $\hat{Y}_t$ is the estimated value, and l is the number of predictions made. Lower values for MAPE means a better fit of the predicted data.

3 Experimental results

To compare the results obtained by ARIMA, linear regression, and multilayer perceptron, the following 10 publicly available datasets were used: All Employees: Mining and Logging: Oil and Gas Extraction [14], All Employees: Education and Health Services: Health Care [15], Civilian Unemployment Rate [16], Total Checkable Deposits [17], All Employees: Goods-Producing Industries [18], All Employees: Construction [19], All Employees: Total Private [20], All Employees: Manufacturing [21], Employment-Population Ratio - Men [22], and Employment-Population Ratio - Women [23]. We considered the recorded data up to (and including) June 2013. The enumerated datasets consist of monthly measurements, with both seasonal and non-seasonal adjustment. The number of values for each time series are given in Table 1. For the considered datasets, the last 12 values (i.e. the measurements for July 2012—June 2013) were kept away during developing the predictive models, to assess the accuracy of the predictions made by each model.

For the ARIMA model, the values for the coefficients p, d, q, P, D, and Q are automatically determined by the IBM SPSS software, through its internal routines. For the linear regression-based model, the so–called M5 heuristic method was used for automatic feature selection. As described on Weka's documentation page[5], "M5

[5] http://wiki.pentaho.com/display/DATAMINING/LinearRegression

steps through the features removing the one with the smallest standardised coefficient until no improvement is observed in the estimate of the error given by the Akaike information criterion [1]"[6]. Other feature preprocessing options, like greedy method or preserving all the available features lead to inferior results. For the multilayer perceptron, we decided for 10 neurons inside the hidden layer, learning rate 0.3, momentum 0.2, at most 500 training epochs; values close to the ones reported did not significantly change the final results, and the ones mentioned are obtained through a trial and error approach. We allowed for learning both with and without weight decay.

Series	Values	MAPE			
		ARIMA	MLP with weight decay	MLP without weight decay	LR
[14]	498	3.2170	*1.8943*	3.1518	**0.5731**
[15]	282	*0.1973*	**0.1666**	0.3132	0.2573
[16]	786	7.0232	**4.4881**	12.8869	*4.8702*
[17]	654	6.4377	**4.0230**	*4.1402*	12.1636
[18]	894	*0.4098*	**0.3949**	6.5459	0.9609
[19]	894	*1.4730*	**1.2136**	3.5296	7.8756
[20]	894	*0.7690*	**0.5034**	3.4623	0.8690
[21]	894	1.1420	**0.7524**	*0.7910*	2.7882
[22]	786	*0.5308*	**0.5059**	1.4459	1.1893
[23]	786	*0.5030*	**0.3369**	1.8222	0.7422

Table 1: Time series datasets characteristics and prediction results for ARIMA, multilayer perceptron network (MLP) with and without weight decay, and linear regression (LR), quantified by the MAPE score. The "Values" column contains the number of values for the corresponding series, including the 12 values used for model assessment. On every line, the lowest and the lowest values are shown in bold and italic, respectively.

The numerical results from Table 1 are summarized as follows: in 9 datasets out of 10, the multilayer perceptron network with weight decay produced the most accurate predictions overall, with 4.6% to 34.5% smaller than the next accurate values. For the dataset for which the multilayer perceptron network with weight decay did not obtain the lowest MAPE score, it provided the next best result. Training the multilayer perceptron network without weight decay led to modest scores, producing the second best results for two datasets. The ARIMA provided the next best result for 6 cases out of 10. The linear regression produced the lowest MAPE score in one case, and the second next best result in another case.

[6] The citation refers to the bibliography of this paper.

4 Conclusions

This paper considered ten public time series datasets and reported the results obtained by ARIMA, linear regression and nonlinear models obtained through multilayer perceptron networks.

The quantitative results show supringsly accurate predictions obtained by machine learning based methods, better than the ones produced by ARIMA, which is extensively used for such tasks. The best results were obtained for multilayer perceptron with weigth decay, and with a small number of neurons in the hidden layer.

Although it is hard to extrapolate to other datasets, the presented tests show that nonlinear models produced by multilayer perceptron network offer a promising approach for time series forecasting.

References

[1] Akaike, H. *A new look at the statistical model identification*, IEEE Transactions on Automatic Control **19 (6)** (1974). 716723.

[2] Bishop C. M., *Pattern recognition and machine learning*, Springer, 2007

[3] Box, G.E.P., Jenkins, G.M., *Time series analysis: Forecasting and control*, 4th Edition, Wiley, 2008

[4] Chatfield, C., *The analysis of time series. An introduction*, Chapman & Hall/CRC, New York, 1995;

[5] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H., *The WEKA Data mining software: An update*; SIGKDD Explorations, **11**, (2009), no. 1.

[6] Hastie, T., Tibshirani, R., Friedman, J., *The elements of statistical learning. data mining, inference, and prediction*, 2nd edition, Springer 2011.

[7] Haykin, S., *Neural networks and learning machines*, 3rd edition, Prentice Hall, 2008.

[8] Hornik, K., Stinchcombe, M., White, H., *Multilayer feedforward networks are universal approximators*, Neural Networks, **2** (1989), no. 5.

[9] Murphy, K.P., *Machine learning: A probabilistic perspective*, MIT Press, 2012

[10] Popescu, T., *Serii de timp – Aplicaţii în analiza sistemelor (Time series applications in system analysis)*, Technical Publishing House, 2000 (in Romanian)

[11] Sasu A., *Time series seasonal adjustment method*, Bulletin of the Transilvania University of Brasov Series B1 **14(49)** (2007), 91-94.

[12] Sasu, A., *Time series forecasting using ARIMA models*, 8-th European Conference E-COMM-LINE 2007, 20-22 Sept., Bucharest, ISBN-10: 973-88046-6-3, ISBN-13: 978-973-88046-6-1, 67e, 2007.

[13] Yaffee, R.,A., McGee, M., *An introduction to time series analysis and forecasting: with applications of SAS and SPSS*, Academic Press, 2000

[14] http://research.stlouisfed.org/fred2/series/CEU1021100001

[15] http://research.stlouisfed.org/fred2/series/CEU6562000101

[16] http://research.stlouisfed.org/fred2/series/UNRATENSA

[17] http://research.stlouisfed.org/fred2/series/TCDNS

[18] http://research.stlouisfed.org/fred2/series/USGOOD

[19] http://research.stlouisfed.org/fred2/series/USCONS

[20] http://research.stlouisfed.org/fred2/series/CEU0500000001

[21] http://research.stlouisfed.org/fred2/series/CEU3000000001

[22] http://research.stlouisfed.org/fred2/series/LNU02300001

[23] http://research.stlouisfed.org/fred2/series/LNS12300002