Analysis and forecasting of a time series dataset with an ARIMA model

Pablo Torasso
Politecnico of Turin
Torino, Italy
S303196@studenti.polito.it

***Abstract*—In this paper we illustrate the process and results of an analysis conducted exploiting Autoregressive Integrated Moving Average (ARIMA) model on the BJsales dataset, natively available on R. We introduce the dataset doing some initial consideration on its properties, and then we investigate the optimal *p, d* and *q* parameters needed to represent the time series and conduct forecasting on the last ten observations unknown to the model. The source code of the proposed approach is available at : https://github.com/Elbarbons/Analysis-and-forecasting-of-a-time-series-dataset-with-an-ARIMA-model.**

***Index Terms*—ARIMA, BJsales, forecasting, time series, R**

I. Introduction

Over the past few decades, much effort has been devoted to the development and improvement of time series forecasting models aiming to predict future data points based on observed data over a known period. The major objective is obtain the best forecast, i.e., to ensure that the mean square of the deviation between the actual and the forecasted values is as small as possible.

Traditional models for time series forecasting, such as the Box–Jenkins or the Autoregressive Integrated Moving Average (ARIMA) model, assume time series data are generated by linear processes. However, these models may be inappropriate if the underlying mechanism is nonlinear. In fact, real-world systems are often nonlinear [4]. The ARIMA model is a stochastic process defined by three parameters, p, d, and q, where p stands for the Auto-Regressive AR(p) process, d is the integration (needed for the transformation into a stationary stochastic process), and q is the Moving Average MA(q) process [3].

II. Methodology

A. BJsales data

The BJsales data is a time series dataset available in R that describes the evolution over time of the number of sales. It contains 150 observations.

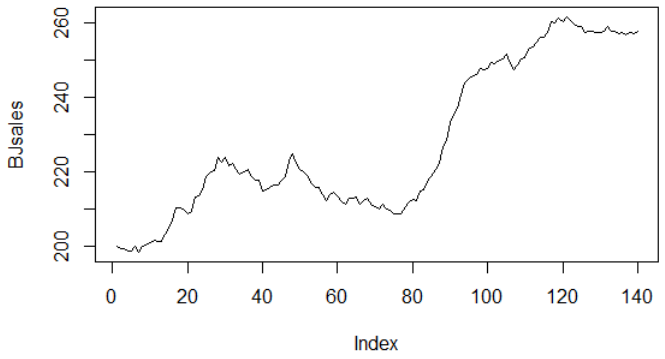

Figure 1. Data trajectory

By plotting the trajectory 1 of the population we can infer visually some properties. It is easy to see that data are not stationary: the mean is not stable and varies with time and has an exponential increase from index 80 to 100. Instead, the variance seems stationary across all the graph.

A more rigorous way to identify a non-stationary time series is the autocorrelation function plot, ACF. For a stationary time series, the ACF will drop to zero relatively quickly, while the ACF of non-stationary data decreases slowly [1]. Looking at

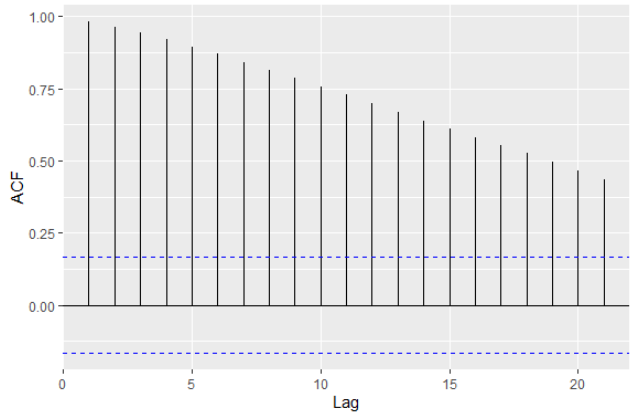

Figure 2. Autocorrelation plot of original data

the autocorelation graph 2 we confirm that the data is non-stationary. Given the non-stationarity of the data we apply a first order differencing defined by

$$\Delta Y_t = Y_t - Y_{t-1} \tag{1}$$

and we obtain a new stationary time series that need no further differentiation; as a matter of fact, looking at the autocorrelation function plot 3 of the first order differenced data we observe that the autocorrelation drops quickly.

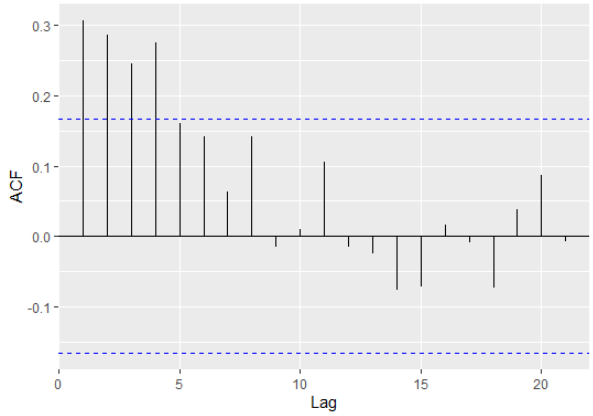

Figure 3. ACF plot on differenced data

Given what we inferred, from now on we will only consider ARIMA(p,1,q) models with fixed *d*, number of differentiation applied to the data.

B. Arima model

The main part of the ARIMA model combines AR and MA polynomials into a complex polynomial, as seen in below equation 2. The ARIMA (p, d, q) model is applied to all the data points.

$$y_t = \mu + \sum_{t=1}^{p}(\sigma_t y_{t-1}) + \sum_{t=1}^{q}(\theta_t \epsilon_{t-1}) + \epsilon_t \tag{2}$$

where the notation is as follows:
μ: the mean value of the time series data;
p: the number of autoregressive lags;
σ: autoregressive coefficients (AR);
q: the number of lags of the moving average process;
θ: moving average coefficients (MA);
ϵ: the white noise of the time series data;
d: the number of differences calculated in 1.

We produced and evaluated ARIMA(p,1,q) models, with $d = 1$ fixed for the reasons described in the previous subchapter II-A, with p and q both ranging from 0 to 3.

III. Results

We produced all the models and compared their AIC, AICc and BIC scores to determine the best parameters *p* and *q* used to forecast. The three top models and their scores are shown in table I.

ARIMA(p,d,q) model	**AIC**	**AICc**	**BIC**
ARIMA(1,1,1)	484.75	484.92	493.56
ARIMA(1,1,2)	486.47	486.77	498.20
ARIMA(2,1,1)	486.50	486.80	498.24

Table I
AIC, AICc and BIC scores of the 3 best ARIMA models

The ARIMA(1,1,1) model shows the best results across all three scores. Substituting in equation 2 the calculated parameters σ and θ, the equality takes the following form:

$$y_t* = 0.8874y*_{t-1} - 0.6550\epsilon_{t-1} + \epsilon_t \tag{3}$$

with $y*_t$ first differences of the original series y_t

Looking at the residuals plot 4 we observe they behave like white noises implying there is not significant autocorrelation: they follow approximately a $normal(0, \sigma^2)$, for some σ.

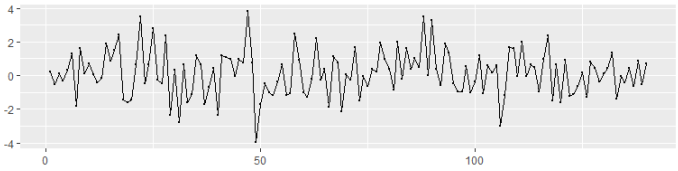

Figure 4. residuals plot

We then forecasted the future unknown data and plotted the 10 points forecasted, the 80% prediction interval and the 95% prediction interval for the next 10 steps. The plot 5 shows the training data, the forecasted points, their prediction intervals and the real future data.

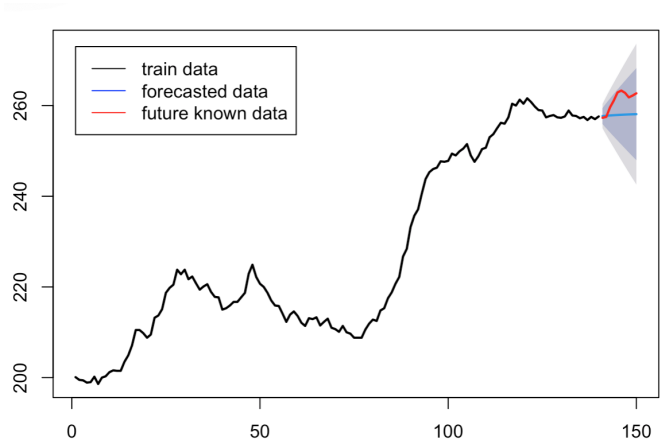

Figure 5. Forecast plot

We also compared our model against a baseline that predicts the next points as the mean of the stationary first order differenced data plus the last known observation. As judgment

parameter we used MSE, Mean Squared Error, revealing an MSE of $\sim$14.14 for the baseline and $\sim$14.23 for the ARIMA model

IV. Analysis of the results

As we said in chapter III all three scores agrees that the best parameters are $(p, d, q) = (1, 1, 1)$. With this setup we produce an 80% prediction interval that covers all ten unknown future points. We are confident in our model and the predictions produced by it because the residuals acted like white noise, demonstrating that the process acquired all the underlying informations about the given data.

Despite that, it is surprising that our model performed slightly worse respect to the naive baseline method according to MSE. This result is frequently observed in time series forecasting and many scientific papers demonstrated that in some circumstances is preferable to choose a simpler model rather than a complex one to achieve higher accuracy scores [5].

We believe that the lack of seasonality and the small sample cause this unexpected behaviour. We are led to think that greater samples bring the ARIMA model to outperform the baseline method and forecast more accurate future points.

V. Conclusions

In this paper we showed the steps to: analyze and prepare the time series data, to choose the right parameters of the ARIMA model, to produce forecasted data and how to evaluate the results. We suggest as further implementation to compare the ARIMA result with others methods results, like MOGA, AI based on the ARIMA model that could provide other possibilities for estimating the parameters (p ,d ,q) and improve data forecasting [2].

References

[1] Hyndman, R.J., & Athanasopoulos, G. (2018) Forecasting: principles and practice, 2nd edition, OTexts: Melbourne, Australia. OTexts.com/fpp2

[2] Hussan Al-Chalabi, Yamur K. Al-Douri & Jan Lundberg (2018) Time Series Forecasting using ARIMA Model: A Case Study of Mining Face Drilling Rig, Division of Operation and Maintenance Engineering ,Luleå, University of Technology

[3] G.E. Box, G.M. Jenkins, G.C. Reinsel and G.M. Ljung, Time series analysis: forecasting and control, 4th ed, John Wiley & Sons, 2016.

[4] J.V. Hansen, J.B. Mcdonald, and R.D. Nelson, Time Series Prediction With Genetic-Algorithm Designed Neural Networks: An Empirical Comparison With Modern Statistical Models, Computational Intelligence, vol. 15, 1999, pp. 171- 184.

[5] Kesten C. Green, and J. Scott Armstrong, Simple versus complex forecasting: The evidence, March 2015