Forecast Analysis Based On Pour Time Series

Emanuele Sansebastiano*

European Master on Advanced Robotics Plus (EMARO+)
Jaume I University, Spain
**emanuele.sansebastiano@outlook.com*

January 2017

Abstract

From the very beginning of the human civilization, men and women tried to know the future spanning every possible path: invoking gods, observing the bird flight and analysing animal's bodies. However, forecasting became strictly scientific just in the recent centuries thanks to mathematicians. In this paper we preform a forecast analysis based on a very pour tourism dataset, which is composed by independent time series having various sizes. Every time series requires its own forecast. In order to do it, many model types are used: polynomial function fitting, autoregressive model (AR) and autoregressive moving average model (ARMA). At first, we optimize automatically the complexity of every model type on a training set. Then, we evaluate every model type forecasting on the whole dataset. The MASE error, used as accuracy index, establishes which model performs the best time series forecast.

Keywords: Time series forecast, AR model, ARMA model, MASE error.

1 Introduction

This project deals with the necessity of forecasting time series using a very pour dataset and without knowing the system at all. Åström and Eykhoff [1] resumed in their survey many techniques and the protocol to follow in order to identify a system. Systems can be grouped in three categories: "white box" systems, where the system components are well known; "grey box" systems, where the system components are partially known; "balck box" systems, where the system components are not known at all. Sjöberg and al. [2] gave a general overview on identifying black box systems. The dataset used in this paper comes from one of the Kaggle competitions[1]; it is composed by 518 independent time series, having non-constant size and scale. The shortest time series counts just 7 values, while the largest 43. For every time series 4 consecutive values have to be forecast. Such a pour dataset leads to small complexity models not to over fit data. Box and Jerkins [3] and Brockwell and Davis [4] wrote two books about time series analysis,which guided the approach used in this paper.

The aim of this paper is:

1) Optimizing the complexity of every model kind to identify the black box system.

[1] https://www.kaggle.com/c/tourism1

2) Comparing various model criteria using their optimal complexity.

3) Defining the best model criteria to forecast based on pour time series.

There are many algorithms based on the forecast error, but, because the dataset has non-constant scale, a scaled error algorithm must be used to evaluate the model. Hyndman and Koehler [5] describe in their paper that the best forecast accuracy index is the mean absolute scaled error (MASE). Because the 518 time series are independent, the mean value of all the 518 MASE values would be used to evaluate the model accuracy. The technique to optimize the model complexity implemented in this paper is pretty simple: all the possible and reasonable model complexity are brutally tested using the 70% of the dataset as training set. The accuracy of the model is tested on the remain 30% of the dataset using the mean MASE. The model complexity having the lowest mean MASE is the complexity chosen for that model type. Van Mulders and al. citecomparison proposed two nonlinear optimization methods for black box identification, but they appears not to be faster than the brute force method for small dataset.

In section 2 a short overview on the model types is presented. In section 3 the detailed protocol used to achieve the optimal complexity of every model type and compare the optimized models so far defined is described. The section 4 deals with the criteria used to define the optimal complexity for every model type. In sections 5 and 6 the results are commented and future works proposed.

All the function are implemented in Matlab [7] and are available on-line[2].

2 Model Types

In order to forecast 4 consecutive values the time series trend has to be defined. Since the dataset deals with tourism data, a market approach appears to be interesting, but, as Öller [8] wrote, it is at odds with the required forecast nature. Economists use to predict values one by one, assuming that the market is locally stable: the first forecast value corresponds to the last value of the time series. Obviously, it imposes that every forecast value is equivalent to the previous one and so to the last of the time series. Following a similar concept we can say that the first predicted value is a weighted combination of the last j values:

$$v_{m+1} = \frac{\sum_{j=1}^{h} w_j v_{m+1-j}}{h} \tag{1}$$

where m is the size of the single time series; h is the number of the last time series values involved in forecasting; w_j is the weight assigned to the value v_{m+1-j}. According to the size of the time series, h may change its range. However, h should never tend to m not to over fit data.

Generally, finding a curve to fit the time series sounds to be more helpful than the previous approaches. In this paper a polynomial function has been used to fit the time series:

$$y = \sum_{j=0}^{q} a_j x^j \tag{2}$$

where y is the tourism value at the moment x, with $x \in \mathbb{N}$.

Analysing the nature of the dataset, we can infer that every value may be influenced by the previous ones. A tourist location success is definitely related to its popularity, which is proportional to the number of persons that have been there previously. So, an autoregressive model approach appears to be interesting to model this system. The most generic model is the autoregressive moving average exogenous input (ARMAX) model:

$$\begin{cases} y(k) = \sum_{i=1}^{N} a_i y(k-i) + m(k) + e(k) \\ m(k) = \sum_{y=0}^{M} b_y u(k-y) \\ e(k) = \sum_{j=0}^{V} c_j n(k-j) \end{cases} \tag{3}$$

where $y(k)$ is the actual output based on the linear combination of the previous N outputs, previous M inputs, previous V white noises and the actual input

[2] All the functions are available at the repository: https://github.com/emanuelesansebastiano/Forecast_model

and the actual white noise. Generally, it is written in a more compact form:

$$A(z)y(k) = B(z)u(k) + C(z)n(k) \tag{4}$$

where $A(z)y(k)$ corresponds to the autoregressive part; $B(z)u(k)$ corresponds to the exogenous part; $C(z)n(k)$ corresponds to the moving average part. The exogenous variables are variables independent to the error term (noise), clearly the input is independent to the noise. The moving average part is considered null if it is defined as pure white noise, because it is equal to zero in average. Generally, there is "colored noise" (linear combination of white noises) in dynamic models. In this specific case, we already have all the time series values: the system is not dynamic. Finally, we know there is no point to keep the exogenous part in our model ($u(k) = 0$). We cannot tell the same for the moving average part: the white noise is assumed to be null in average, it is not zero as the input. So, both AR and ARMA models are included in the best model candidate list. However, we are expecting that keeping the moving average part does not really help.

We have not to forget that every model type could lead to conceptually wrong results: in this case, for example, all the forecast values must be positive. There are various techniques to adjust strictly negative values, the most common are:

1) Using the absolute values.

2) Saturating the negative values to zero.

In this paper the second technique has been used.

3 Protocol Description

As already mentioned in section 1, the dataset is composed by 518 independent time series having non-constant size and scale. For every time series 4 consecutive forecast values are requested. In section 2, many model types are described: the autoregressive models sound to be the most effective. However, according to the time series, a model type could work better than another; using the same model type for every time series does not look like being the best way to proceed, even if the time series are belonging to the same dataset. On the other hand, knowing which model type fits more a specific time series is impossible a priori. So, we should test every model type for every time series every time. It is computationally heavy and totally denies real-time applications. So, evaluating which model type fits better time series dataset in average appears to be the only way to size down the computational cost. Every model is evaluated according to a forecast accuracy index: the mean absolute scaled error (MASE). This algorithm provides an accuracy value for every time series. Then, the mean value of all the 518 MASE values is the actual value used to evaluate the model accuracy. Because of the non-constant scale, the mean absolute error (MAE) cannot be used directly; it must be scaled on the time series:

$$\begin{cases} \text{MASE} = \frac{\text{MAE}}{Q} \\ \text{MAE} = \frac{\sum_{i=1}^{H} |\text{err}_i|}{H} \\ \text{err}_i = Y_i - y_i \\ Q = \frac{1}{N-1} \sum_{j=2}^{N} |\hat{y}_j - \hat{y}_{j-1}| \end{cases} \tag{5}$$

where $\hat{y}_j$ is the j^{th} term of the time series; Y_i is the i^{th} real future value; y_i is the i^{th} forecast value; err_i is the i^{th} forecast error; Q is the scaled error performed on the N time series values; MAE is the mean absolute error performed on the H forecast values of the single time series.

In appendix A other possible algorithms to evaluate the error are presented. They follow the same trend of the MASE.

For most of the model types described, there are some terms to tune:

1) The number of values we want to consider (h) and the assigned weights (w_j) to perform the weighted combination (eq. 1);

2) The degree (q) of the polynomial function to fit the time series (eq. 2);

3) The complexity of $A(z)$ for the AR model, and $A(z)$ and $C(z)$ for the ARMA model (eq. 4).

Even for short time series, like the analysed ones, tuning these parameters is relevant and pretty difficult

to do manually. Talking about the AR and ARMA models, the best complexity does not generally overpass 6/7 because the time series are constantly updated; small complexity model fits better real-time systems. In order to find the best complexity, every time series has been split in two sub sets: a training set defined by the first 70% of the time series values, and a test set including the last 30% values. For the Polynomial, AR and ARMA model types many different criteria have been used to define the optimal complexity of the model (sec. 4). Looking at the pseudocode presented in section 3.1, we can clearly distinguish two main protocol part:

1) The search of the best complexity for every model criteria.

2) The comparison between every model optimized in terms of complexity.

Both parts use the mean MASE as forecast accuracy index. The first requires more time and defines the best complexity for every model criteria based on the 70% of the dataset (training set). The second part uses the model complexities found by the first one to model and perform the forecast on the whole dataset. Finally, the model criteria are sorted by the lower MASE value.

3.1 Pseudocode Algorithms

1. Main Algorithm

```
< start >
-Best complexity for every model kind-
for (every model kind) do
    for (every reasonable complexity) do
        Generate the model on the 70%;
        Forecast on the 70%;
        Test forecast on the 30% (MASE);
    end for
    Keep the lowest MASE model;
end for
-Best model kind evaluator-
for (every model kind) do
    Generate the model on the 100%;
    Forecast on the 100%;
    Adjust the forecast (negative to 0);
    Test forecast on the solution (MASE);
end for
Sort models by the lower MASE;
< end >
```

4 Model Criteria

This section deals with the criteria used to find the best complexity for every model type.

4.1 Average Models

Two criteria for the average approach:

1) All the forecast values are the same and they correspond to the mean value of the time series.

2) According to the equation 1, decreasing weight values w_j are defined following a second order curve ($w_1 = 100\% \dots w_{h+1} = 0\%$). Moreover, the forecast is performed value by value including them in the time series to predict the future ones. It means 4 forecast values are not the same.

Even if the time series have different size, these criteria generates the same model all of them.

From now we are going to refer to these two model criteria as “Aver” and “Aver_w” respectively.

These two criteria have not been investigated deeply. No automatic tuning has been implemented; the parameters used to tune them were chosen after some manual tests.

4.2 Polynomial Models

Two criteria for the polynomial function approach:

1) Every polynomial function is acceptable, even the ones producing negative forecast values during the complexity analysis on the 70% of the dataset.

2) Just the polynomial functions producing non-negative forecast values during the complexity analysis on the 70% of the dataset are acceptable.

For both criteria there is no common polynomial degree: every time series has its own model complexity. The polynomial degrees tested go from 1 to 10, or to the time series size if it is smaller than 10.

From now we are going to refer to these two model criteria as "Poly" and "Poly_p" respectively.

4.3 Autoregressive Models

As mentioned in section 2, there are two kind of autoregressive models that fit this problem: the pure autoregressive model (AR) and the autoregressive moving average model (ARMA). The variable criteria defining a specific model complexity for every time series has been implemented for both model kind. From now we are going to refer to them as "AR_var" and "ARMA_var" respectively. Due to complex computation, the ARMA model identification requires a lot of time. So, the maximum complexity has been imposed equal to 3 for both $A(z)$ and $C(z)$. The AR model, instead, has a lighter identification, that allowed to push the complexity investigation up the half of the whole time series size. Recall that the maximum time series size is 43.

Since the AR model identification is faster, we can push the investigation forward, trying to find a model having homogeneous complexity on the dataset:

1) Basing the complexity investigation range on the shortest time series and using the same complexity for every time series.

2) Using the same percentage complexity for every time series[3].

From now we are going to refer to these two model criteria as "AR_comm" and "AR_perc" respectively.

5 Results and Conclusions

The whole experiment required a bit less than 40 minutes to be performed. The results are presented in table 1:

1) The accuracy index of the model (MASE).

2) The mean complexity value of the model (Complexity).

3) The time required to find the best complexity of the model criteria (Time).

Model	MASE	Complexity	Time [s]
AR_comm	2.535	1	81
AR_var	2.666	1.32	220
AR_perc	2.745	1.65	274
ARMA_var	2.853	A:1.06 C:1.15	1729
Poly_p	3.874	1.91	8
Poly	3.876	1.94	7
Aver	6.217	none	0.004
Aver_w	7.509	none	0.015

Table 1: Model evaluation

As we expected, the mean complexity of all the models is not big, it is always lower than 2. The data set was so pour that we could not expect anything different. The autoregressive models confirm the expectations: they are holding the first positions. Moreover, the ARMA model appears to be the worst among them considering both MASE and time required. In particular, the MASE proved what said in section 2: the noise is better to be assumed being white if the system is stationary. Surprisingly, the best AR model criteria is the "AR_comm". In theory, the variable criteria ("AR_var") should fit better, but the best complexity analysis was performed on the 70% of the dataset, which could not correspond to the best complexity analysis performed on the whole dataset. Moreover, due to the pour nature of the dataset, lower complexity are always preferable. Larger datasets would probably more compatible with the model "AR_var". In the end of the list there are the two criteria of the average approach. Even if they are the faster models, the average approach is definitely not suggested in any case. Considering the speed, the polynomial functions impose their selves as the best ones: they are more than 10 times faster than the fastest autoregressive model. Anyway, the MASE is more than the 30% higher than the worst autoregressive model. Identifying a

[3] 10% complexity means the 10% of the half size of the time serie.

time series system using a polynomial function is interesting only to have a first idea of the trend; this approach is not recommend to perform a definitive forecast.

6 Future Works

As an extension of this project, implementing some filter to improve the dataset at first can be done. However, the small size of the dataset would make this step challenging.

Appendix

A Other Forecast Accuracy Indicator

Root Mean Absolute Scaled Error (RMASE)

$$\begin{cases} \text{RMASE} = \frac{\text{RMSE}}{Q} \\ \text{RMSE} = \sqrt{\frac{\sum_{i=1}^{H} \text{err}_i^2}{H}} \\ \text{err}_i = Y_i - y_i \\ Q = \frac{1}{N-1} \sum_{j=2}^{N} |\hat{y}_j - \hat{y}_{j-1}| \end{cases} \tag{6}$$

where $\hat{y}_j$ is the j^{th} term of the known time series; Y_i is the i^{th} real future value; y_i is the i^{th} forecast value; err_i is the i^{th} forecast error; Q is the scaled error performed on the N time series values; RMSE is the root mean squared error performed on the H forecast values of the single time series.

Mean Absolute Percentage Scaled Error (MAPSE)

$$\begin{cases} \text{MAPSE} = \frac{\text{MAPE}}{Q_{\text{perc}}} \\ \text{MAPE} = \frac{\sum_{i=1}^{H} |\text{err}_{\text{perc},i}|}{H} \\ \text{err}_{\text{perc},i} = \frac{Y_i - y_i}{Y_i} 100 \\ Q_{\text{perc}} = \frac{1}{N-1} \sum_{j=2}^{N} |\frac{\hat{y}_j - \hat{y}_{j-1}}{\hat{y}_j} 100| \end{cases} \tag{7}$$

where $\hat{y}_j$ is the j^{th} terms of the known time series; Y_i is the i^{th} real future value; y_i is the i^{th} forecast value; $\text{err}_{\text{perc},i}$ is the i^{th} percentage forecast error; Q_{perc} is the percentage scaled error performed on the N time series values; and MAPE is the mean absolute percentage error performed on the H forecast values of the single time series.

Acknowledgments

I thank Prof. Maria Museros, Jaume I University, for providing valuable advices and support to complete this project. I also thanks Prof. Marco Baglietto, Università degli Studi di Genova, for taking "System Identification" course during the first year of the EMARO+ project. Even if they did not contributed directly to this project, I would like to thank Mr. Leslie Lamport, who invented LaTeX, for having produced a such amazing tool and Prof. Giorgio Quintana, Jaume I University, for providing a well done guide to use LaTeX during his lectures.

References

[1] Åström, K. and Eykhoff, P. (1971). System identification—A survey. Automatica, 7(2), pp.123-162.5.

[2] Sjöberg, J., Zhang, Q., Ljung, L., Benveniste, A., Delyon, B., Glorennec, P., Hjalmarsson, H. and Juditsky, A. (1995). Nonlinear black-box modeling in system identification: a unified overview. Automatica, 31(12), pp.1691-1724.

[3] Box, G. and Jenkins, G. (1976). Time series analysis. 1st ed. San Francisco: Holden-Day.

[4] Brockwell, P. and Davis, R. (1991). Time series. 1st ed. New York: Springer-Verlag.

[5] Hyndman, R. and Koehler, A. (2006). Another look at measures of forecast accuracy. International Journal of Forecasting, 22(4), pp.679-688.

[6] Van Mulders, A., Schoukens, J., Volckaert, M. and Diehl, M. (2009). Two Nonlinear Optimization Methods for Black Box Identification Compared. IFAC Proceedings Volumes, 42(10), pp.1086-1091.

[7] Ljung, L. (1997). System identification toolbox for use with MATLAB. 1st ed. Natick, MA: MathWorks, Inc.

[8] Öller, L. (2001). Forecasting Non-stationary Economic Time Series. International Journal of Forecasting, 17(1), pp.133-134.