Proceedings of the 17th International Conference on Computational and Mathematical Methods in Science and Engineering, CMMSE 2017
4–8 July, 2017.

A specialized lazy learner for time series forecasting

Francisco Martínez[1], María P. Frías[2], Francisco Charte[1] and Antonio J. Rivera[1]

[1] *Department of Computer Science, University of Jaén (Spain)*

[2] *Department of Statistics and Operations Research, University of Jaén (Spain)*

emails: `fmartin@ujaen.es`, `mpfrias@ujaen.es`, `fcharte@ujaen.es`, `arivera@ujaen.es`

Abstract

In a time series context the nearest neighbour algorithm looks for the historical observations most similar to the latest observations of the time series. However, some nearest neighbours can be misleading. In this paper we propose that, if prior information about the structure of the time series is known, the search space of possible neighbours can be narrowed so that some possibly misleading neighbours are avoided. This way a more effective forecasting method can be obtained.

Key words: time series forecasting, lazy learners

1 Introduction

Time series forecasting has been traditionally done using statistical models, such as ARIMA [1] or exponential smoothing [2]. However, the last decades have seen the widespread use of computational intelligence techniques, such as artificial neural networks [3], to forecast univariate time series.

Lazy learners [4], such as nearest neighbours, are one of the computational intelligence techniques applied in time series forecasting. As the next section will explain in more detail, a nearest neighbours algorithm tries to forecast the next future values of a time series looking for past realizations that are similar to the last observations in the time series. The subsequent observations of the past similar realizations are used to forecast the time series.

 ISBN: 978-84-617-8694-7

However, we think that in certain time series, some similar past realizations can be misleading and, therefore, produce bad forecasts. To solve this problem, we propose to narrow the search space of possible neighbours, excluding possibly misleading instances.

The remainder of this paper is structured as follows. Section 2 describes how the nearest neighbour algorithm can be applied to predict the future values of a time series. Section 3 explains why to narrow the search space of neighbours could be effective. Section 4 describes the experimental setup and how the meta parameters of the nearest neighbour algorithm have been chosen. Section 5 shows the results of our experimentation and Section 6 draws some conclusions.

2 Using nearest neighbours in time series forecasting

The nearest neighbour is a classical supervised algorithm in machine learning. Originally, it was applied in classification tasks. Given an unlabeled example, we look for the most similar labeled example, according to some features of the examples and using a distance function to express similarity. The label of the most similar example is used to classify the unlabeled example. The labeled examples can contain outliers and errors that can distort the prediction, so instead of looking for the nearest neighbour normally the k nearest neighbours are found and their majority class is used as prediction.

The nearest neighbours algorithm can be easily extended to a regression task. Instead of having labeled examples, the examples contain a numeric target value and the target value of the k nearest neighbours is combined—for example, it is averaged—to produce the predicted value. In a time series forecasting scenario, the target value of an example is a time series observation and the features describing the example are lagged values of the observation. This way, the next value of a time series is predicted looking for examples in the time series similar to the last observations—see Figure 1. Figure 1 shows an example of one-step ahead forecasting using lags 1 to 4 as features and two nearest neighbours. The last four values of the time series are the features of the example to be regressed on and the two sets of consecutive white squares the 2 nearest neighbours, whose targets are the black triangles. The mean of the two targets is the forecast—the asterisk.

The underlying reason to using nearest neighbours in a time series scenario is that a time series contains repetitive patterns. Hence, given the last behaviour of a time series we try to look for similar past behaviours in the hope that their subsequent values can be similar to the future values of the time series.

3 Why to narrow the search space of neighbours

When we do not know too much about the structure of a time series, we simply select the features—lagged values—of the future observation to be forecast and look for the most

 ISBN: 978-84-617-8694-7

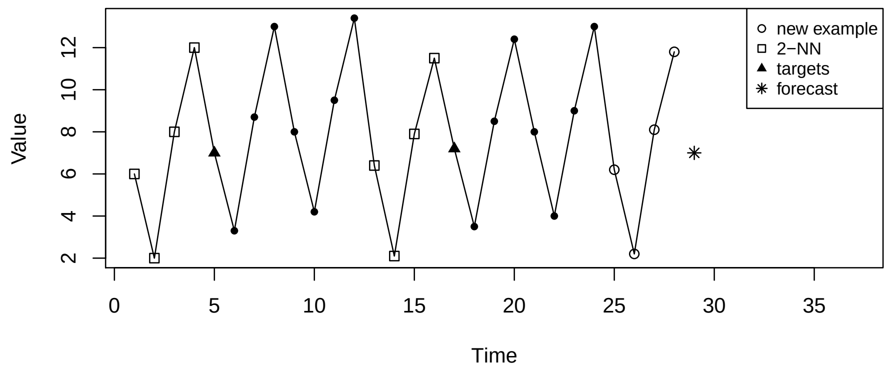

Figure 1: Example of 2 nearest neighbours for one-step-ahead forecasting.

similar examples in the time series. However, frequently we *do* know about the structure. For example, in the data explained in more detail in the next section, we are going to work with daily time series of cash money withdrawals at cash-machines. Apart from other patterns, these series have a strong weekly seasonality. Concretely, the withdrawals on weekends are very different from the withdrawals on workdays. Suppose now that we have information until a certain Friday and we want to predict the withdrawals on the following day, i.e., a Saturday. Let us also suppose that we are using the nearest neighbour algorithm and we use as feature the first lagged value. That is, to predict a Saturday we will find the day more similar to its previous day—a Friday. The most similar day found could be any day, let us suppose that it is a Tuesday. So we are going to predict the next Saturday withdrawals using the withdrawals on a Wednesday. This is not a desirable situation, because the withdrawals on Saturdays have nothing to do with the withdrawals on Wednesdays.

In our proposal, for this case we propose to narrow the search space looking only on previous Fridays to produce a more sensible forecast. That is, the prediction on a Saturday will be the average of several Saturdays whose previous Fridays are similar to the last Friday in the time series.

 ISBN: 978-84-617-8694-7

4 Experimental setup

In order to assess the effectiveness of our proposal we have used data from the NN5 time series competition[1]. In this competition 111 time series of 2 years of daily cash money withdrawals at various automatic teller machines (ATMs, or cash machines) at different locations in England were used. The goal was to forecast the cash money demand for the next 56 days ahead of every one of the 111 time series.

To assess forecast accuracy the symmetric mean absolute percentage error—SMAPE—was used. Given a vector forecast f for the next 56 days of a time series and the vector y of actual future values, the SMAPE is computed as follows:

$$SMAPE = \frac{1}{56} \sum_{t=1}^{56} \frac{|y_t - f_t|}{(|y_t| + |f_t|)/2} 100$$

The SMAPE obtained for every one of the 111 time series by a given method is averaged to obtain a final SMAPE, which is used as the main forecast accuracy measure to compare the different methods.

Figure 2 includes a time series from the NN5 competition. Although these series contain multiple overlying seasonalities, in this work we are interested in experimenting with their strong weekly seasonality pattern.

4.1 The lazy learner meta parameters

In order to apply the nearest neighbours algorithm several meta parameters have to be chosen. Next, we describe our choices:

- As similarity function, i.e. to select how similar two examples are given their features, the Euclidian distance has been used.

- As combination function, i.e. to combine the target values of the k nearest neighbours, the arithmetic mean has been used.

- For k—the number of nearest neighbours—several values have been used to assess how robust the proposed method is.

- The final parameter is the lagged observations that are used as feature vector to compare the similarity among examples. We have used consecutive lags starting from lag 1. Concretely, we have experimented with lags 1, 1 to 7 and 1 to 14. We have chosen multiples of seven, because we are interested in the weekly seasonality.

[1] http://www.neural-forecasting-competition.com/NN5/

 ISBN: 978-84-617-8694-7

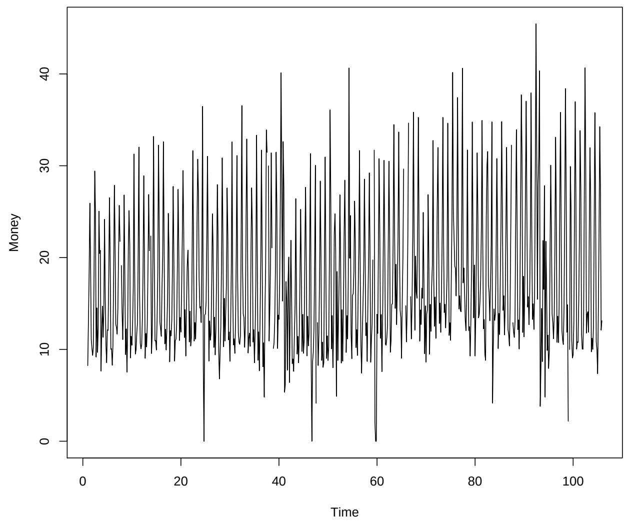

Figure 2: Example of time series from the NN5 competition.

Finally, we have to explain how the multiple—in this case 56—step-ahead forecasts are generated. When the forecast horizon is higher than 1 you have to chose among different strategies to generate the multiple forecasts, being the most used ones the direct, recursive and MIMO [5]. To date, there is no clear evidence that any strategy is superior than the others. Given this situation, we have chosen the recursive or iterative strategy because is quite straightforward. This strategy is used in the ARIMA and exponential smoothing methodologies. The recursive strategy uses a one-step-ahead model iteratively to get all the forecasts. That is, for forecasting horizon 1, the lagged observations of the time series are used as autoregressive variables for the model. For forecasting horizon 2, lacking a historical lagged one value to be used as autoregressive variable, the forecast for horizon 1 is used instead.

5 Experimentation

The goal of our experimentation is to assess whether, knowing information about the structure of a time series, the use of a selected space of possible neighbours can improve the forecast accuracy of the nearest neighbours algorithm. To that end, we have experimented with forecasting the 111 time series of the NN5 competition. We try to take advantage of our prior knowledge of the weekly seasonality pattern in the time series of this competition.

 ISBN: 978-84-617-8694-7

The way in which we are going to use this knowledge is as follows: in order to forecast the money withdrawals on one day, we take into account the day of the week to be forecast and only use as possible nearest neighbours examples, those examples whose targets have the same day of the week as the day being forecast.

In order to assess whether this strategy is effective we are going to execute the following nearest neighbours instances to predict 56 days ahead of the 111 time series of the NN5 competition:

- Classical nearest neighbours algorithm with k ranging from 1 to 10 and autoregressive lagged values at consecutive lags: 1, 1 to 7 and 1 to 14.
- Nearest neighbours algorithm with the same meta parameters, but using a selected space of possible neighbours, so that only examples whose targets have the same day of the week as the day of the value being forecasted are included.

Table 1: Global SMAPE of different algorithms over the 111 NN5 time series.

	1	2	3	4	5	6	7	8	9	10
C - 1	49.04	42.52	40.27	39.36	39.04	38.30	38.98	39.20	38.77	39.00
SS - 1	32.87	27.66	26.32	25.27	24.60	24.51	24.18	24.09	24.14	24.13
C - 1:7	31.55	28.21	25.33	23.95	23.61	23.43	23.36	23.51	23.60	23.48
SS - 1:7	30.16	25.42	23.56	23.08	22.56	22.37	22.32	22.44	22.51	22.58
C - 1:14	32.19	26.63	24.77	23.97	23.39	23.06	22.99	22.81	22.71	22.76
SS - 1:14	31.28	25.33	24.02	23.04	22.93	22.46	22.35	22.15	22.12	22.15

The results of our experiment are shown in Table 1. The first row of this table indicates the numbers of neighbours—i.e., the k meta parameter. The first column indicates whether the algorithm uses the classical approach (C) or our proposal of a selected search space (SS). After the hyphen the number of lags used as autoregressive variables is shown—for example, C - 1:7 means the classical algorithm with lags from 1 to 7. For every combination of type of algorithm, lags and k the table shows its forecast accuracy using the SMAPE measure over the 111 time series from the NN5 competition. The computation of SMAPE was described in the previous section. The most outstanding result is that for every combination of k and lags, the new approach outperform the results of the classical algorithm. The improvement in forecast accuracy is especially evident when only one lag is considered. This is a expected result, as with only a lagged value is easier to find misleading neighbours.

6 Conclusions

In this paper we have proposed to reduce the search space of possible neighbours in a nearest neighbours algorithm applied to time series forecasting when prior information about the

 ISBN: 978-84-617-8694-7

structure of the time series is known. The goal is to look for only significant neighbours. Our preliminary experimentation on the time series of the NN5 competition seems to indicate that the proposal is quite effective.

Acknowledgements

This paper has been partially supported by the project TIN2015-68854-R (FEDER Founds) of the Spanish Ministry of Economy and Competitiveness.

References

[1] G. E. Box, G. M. Jenkins and G. C. Reinsel, *Time Series Analysis: Forecasting and Control, 4th edition*, John Wiley & Sons, Hoboken, 2008.

[2] R. J. Hyndman, A. B. Koehler, J. K. Ord and R. D. Snyder, *Forecasting with Exponential Smoothing: The State Space Approach*, Springer–Verlag, Berlin, 2008.

[3] S. F. Crone, M. Hibon, K. Nikolopoulos, *Advances in forecasting with neural networks? Empirical evidence from the NN3 competition on time series prediction*, International Journal of Forecasting **27(3)** (2011) 635–660.

[4] I. H. Witten, E. Frank and M A. Mark, *Data Mining: Practical Machine Learning Tools and Techniques, 3rd edition*, Morgan Kaufmann Publishers Inc., San Francisco, 2011.

[5] B. T. Souhaib, G. Bontempi, A. F. Atiya and A. Sorjamaa, *A Review and Comparison of Strategies for Multi-step Ahead Time Series Forecasting Based on the NN5 Forecasting Competition*, Expert Syst. Appl. **39(8)** (2012) 7067–7080.

 ISBN: 978-84-617-8694-7