Computers and Operations Research 000 (2017) 1–8

Contents lists available at ScienceDirect

Computers and Operations Research

journal homepage: www.elsevier.com/locate/cor

Forecasting for big data: Does suboptimality matter?

Konstantinos Nikolopoulos [a,*], Fotios Petropoulos [b]

[a] forLAB, Bangor Business School, Bangor University, UK
[b] School of Management, University of Bath, UK

ARTICLE INFO

Article history:
Received 12 September 2016
Revised 16 January 2017
Accepted 8 May 2017
Available online xxx

Keywords:
Forecasting
Big data
Optimisation
Retail

ABSTRACT

Traditionally, forecasters focus on the development algorithms to identify optimal models and sets of parameters, optimal in the sense of within-sample fitting. However, this quest strongly assumes that optimally set parameters will also give the best extrapolations. The problem becomes even more pertinent when we consider the vast volumes of data to be forecast in the big data era. In this paper, we argue if this obsession to optimality always bares the respective fruits or do we spend too much time and effort in the pursuit of it. Could we better off by targeting for faster and robust systems that would aim for suboptimal forecasting solutions which, in turn, would not jeopardise the efficiency of the systems under use? This study throws light to that end by means of an empirical investigation. We show the trade-off between optimal versus suboptimal solutions in terms of forecasting performance versus computational cost. Finally, we discuss the implications of suboptimality and attempt to quantify the monetary savings as a result of suboptimal solutions.

1. Introduction

Every forecasting story starts with the same ritual: an excerpt from the renowned M-Competitions (Makridakis et al., 1982; 1993; Makridakis and Hibon, 2000). The forecasting competitions that from the early 80s to late 90s road-mapped the basic principles of forecasting; with the first one being: "statistically sophisticated or complex methods do not necessarily provide more accurate forecasts than simpler ones".

The story remains by and large the same up to today - with even the latest state of the art research supporting the assertion. Ghandara et al. (2016) provided further empirical evidence that nature-inspired optimization routines embedded in complex models do not necessarily lead to any performance improvement, if any. They demonstrate that under the volatility and uncertainty met in most financial markets, complex prediction models are on par or worse than more simple models in out-of-sample forecasting evaluation and they urge for future research to focus on the conditions under which computer intelligence optimization methods are being utilized in practice.

In fact forecasting as a discipline has not moved forward much since these research milestones were achieved back in the 80s and 90s. And that despite the call for action from the very originator of the field, Professor Spyros Makridakis. In an interview for the International Journal of Forecasting (Fildes and Nikolopoulos, 2006), he urged for seizing the power provided from super-intelligent and super-fast ICT systems in order to see, analyse and forecast data in a much better way. In a way, he opened the "forecasting for big data" agenda much earlier and asked the pure ICT potential to be harnessed for better forecasting capabilities in practice. The theory, however, was there anyway for many decades in the form of advanced data mining and knowledge extraction algorithms (Härdle, 1992; Haykin, 1998; 2008; Heaton, 2012).

This lack of progress is not attributed to neither the lack of IT/ICT power nor the (non-) advance of respective algorithms: it is all down to 21st-century business environment being so volatile that only robust and fast adaptive methods can provide good forecasts over a long period of time. This last point is very important as we need accurate forecasts for each and every decision (and thus) forecasting period. So, one-off "wonder" forecasting methods are not good in real life; robustness is a key element.

Another key point for methods to be successful is to be simple, as per the opening quote, but also being adaptive and able to be tuned fast for the respective performance. Methods that over learn and overoptimise training data sets are not good enough in real life contexts (Haykin, 1998), as the in-sample learning follows a U-shape function so that after a point over-training leads to negative in-sample and even worse out-of-sample performance.

This is exactly what our contribution from this research is aspiring to corroborate. What is the extent of optimality that we should aim for when training and selecting respective parameters in fore-

* Corresponding author.
E-mail address: k.nikolopoulos@bangor.ac.uk (K. Nikolopoulos).

http://dx.doi.org/10.1016/j.cor.2017.05.007
0305-0548/

Table 1
Length of the available monthly industry time series.

Number of observations	96	122	128	133	134	136	137	139	140	141	142	143	144
Number of time series	1	1	1	58	46	2	1	9	6	12	5	7	185

casting methods? We want to explore in-sample optimised suboptimal parameter selection of forecasting models, and sequentially quantify the impact, if any, on out-of-sample forecasting accuracy metrics.

We consider as an illustrative example the context of retails operation management: retailers handle from a few hundred products (in a local store), to a few thousands in a local Tesco Express store, to 100,000 SKUs (Stock Keeping Units) in a large Sainsbury's store in UK. The replenishment frequency can be from several hours for fast moving consumer goods like milk and vegetables, to weekly for stationery etc. The hierarchy dictates that orders are set at local shop level but supply is decided at the distribution centre level and the method to provide forecasts range from very basic extrapolations methods like Naive and Moving Averages (Makridakis and Hibon, 2000), to very computational intensive methods with ANNS, Genetic Algorithms and swarm intelligence (Haykin, 1998). This is an example of "big data" in terms of more the sheer volume of information that has to be handled, rather than the "richness" of it - in terms of exploratory variables that can drive demand. In such contexts any savings that can be achieved is important, and to that end suboptimal parameter selection could save a lot of computational time in forecasting support systems, as will be evidenced in our study.

In this research study, we explore suboptimality by considering a simple forecasting method (Simple Exponential Smoothing) and two optimisation approaches (grid-search and trial and error). Using a subset of the M3-competition data, we demonstrate the effects of suboptimality on forecast accuracy, namely the symmetric Mean Absolute Percentage Error and, consequently, the statistical differences in the performance rankings. We trade-off forecast accuracy against the computational time required for producing optimal versus suboptimal models. The next section discusses the data used in this study and the experimental design that was implemented. Section 3 presents the numerical results. Section 4 provides a short discussion of the results as well as implications for theory, practice and software vendors. Section 5 concludes the study.

2. Design

In order to explore the effects of optimality and suboptimality on the forecasting performance, we use the monthly industry subset from the M3-competition (Makridakis and Hibon, 2000). The M3-competition is the largest up-to-date forecasting competition, featuring a total of 3003 time series of various categories (micro, macro, demography, finance, industry and other) and frequencies (yearly, quarterly, monthly and other). In the original study (Makridakis and Hibon, 2000), 24 methods and commercial packages were compared with regards to their forecasting performance. Since then, the data has been used numerous times for research purposes, and the development of new forecasting methods. The industry monthly subset consists of 334 time series of varying lengths, however in all cases the available history spreads for at least eight years with a mean value of twelve years. The exact lengths and respective number of time series are presented in Table 1. In their majority, the data represent either sales or demands and, as such, can be considered a good proxy for retail data.

The forecasting function that is implemented in this study is the simplest form of the exponential smoothing family, the simple exponential smoothing (SES) method. SES is very widely used in practice and is suitable for data that do not exhibit trend or seasonality. It is based on an exponential smoothing average, where more recent observations are assigned larger weights. The degree of the smoothness is controlled via a smoothing parameter, α, which takes values in the range [0, 1]. The one-step-ahead forecast of the exponential smoothing method is calculated as $f_{t+1} = \alpha y_t + (1-\alpha) f_t$, where y_t represents the actual value at period t and f_t the forecast for the respective period. If forecasts for further horizons are required, these are equal to the one-step-ahead forecast, $f_{t+h} = f_{t+1}$, as SES produces flat forecasts.

In this work we study the effects of optimising (or suboptimising) the α smoothing parameter. The initial forecast (also called initial level) is not optimised, rather it is set equal to the initial actual observation, or $f_1 = y_1$. The algorithmic implementation (in R language) of SES that we used in this study is provided in Appendix A.

Two simple optimisation methods are considered. The first one is widely known as grid-search optimisation (also known as parameter sweep or exhaustive search). Keeping in mind that the parameter to be optimised can take values within a certain range (in our case α takes values in [0, 1]), the algorithm starts from the one end of the range and reaches the other end after n steps. Essentially, all possible values of the parameter within the range are considered with an updating interval that equals to $m = |max - min|/n$, where max and min correspond to the limits of the range. For example, if $n = 100$, then every α value with two decimal points is tested $(0, 0.01, 0.02, \ldots, 1)$. For each value of the smoothing parameter, the corresponding one-step-ahead forecasts are calculated and the model fit is measured by the means of the Mean Squared Error (MSE):

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - f_i)^2 \quad (1)$$

Other error measures could be considered (such as the Mean Absolute Error, or MAE), however the MSE is the most widely used in practice. Effectively, $n+1$ MSEs are calculated and the smoothing parameter with the lowest MSE is considered to be the optimal one. The algorithmic implementation of the grid-search optimisation is provided in Appendix A.

The second optimisation algorithm, that we consider in this study, is the trial and error algorithm, which is a fixed-step convergence procedure through a modified Luus–Jakola approach. The search of the optimal α smoothing parameter starts from the values 1/3 and 2/3 where the corresponding MSEs are calculated. The smoothing value with the lowest MSE is selected as the current optimal ($\dot{\alpha}$). For every subsequent step ($k = 2, 3, \ldots$), the algorithm calculates the MSE that corresponds to the smoothing values with distance $\frac{1}{3\times 2^{k-1}}$ from the current optimal, or $\dot{\alpha} \pm \frac{1}{3\times 2^{k-1}}$. Among the smoothing values $\dot{\alpha} - \frac{1}{3\times 2^{k-1}}$, $\dot{\alpha}$ and $\dot{\alpha} + \frac{1}{3\times 2^{k-1}}$, the one with the lowest MSE is selected as the new current optimal. This procedure is repeated for a pre-specified number of steps n, with $k \leq n$. Effectively, $2n$ MSEs are calculated, while the trial and error approach is expected to work well when MSE is a U-shape function of the α smoothing parameter. The algorithmic implementation of the trial and error optimisation is provided in Appendix A.

For each of the two optimisation methods discussed above, ten cases are considered with regards to the value of n. Table 2 provides the number of steps (n) considered in each case. It is worth

Please cite this article as: K. Nikolopoulos, F. Petropoulos, Forecasting for big data: Does suboptimality matter? Computers and Operations Research (2017), http://dx.doi.org/10.1016/j.cor.2017.05.007