International Journal of Forecasting 32 (2016) 303–312

Contents lists available at ScienceDirect

International Journal of Forecasting

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

Bagging exponential smoothing methods using STL decomposition and Box–Cox transformation

Christoph Bergmeir[a,*], Rob J. Hyndman[b], José M. Benítez[c]

[a] Faculty of Information Technology, Monash University, Melbourne, Australia

[b] Department of Econometrics & Business Statistics, Monash University, Melbourne, Australia

[c] Department of Computer Science and Artificial Intelligence, E.T.S. de Ingenierías Informática y de Telecomunicación, University of Granada, Spain

ARTICLE INFO

Keywords:
Bagging
Bootstrapping
Exponential smoothing
STL decomposition

ABSTRACT

Exponential smoothing is one of the most popular forecasting methods. We present a technique for the bootstrap aggregation (bagging) of exponential smoothing methods, which results in significant improvements in the forecasts. The bagging uses a Box–Cox transformation followed by an STL decomposition to separate the time series into the trend, seasonal part, and remainder. The remainder is then bootstrapped using a moving block bootstrap, and a new series is assembled using this bootstrapped remainder. An ensemble of exponential smoothing models is then estimated on the bootstrapped series, and the resulting point forecasts are combined. We evaluate this new method on the M3 data set, and show that it outperforms the original exponential smoothing models consistently. On the monthly data, we achieve better results than any of the original M3 participants.
© 2015 International Institute of Forecasters. Published by Elsevier B.V. All rights reserved.

1. Introduction

After more than 50 years of widespread use, exponential smoothing is still one of the most practically relevant forecasting methods available (Goodwin, 2010). This is because of its simplicity and transparency, as well as its ability to adapt to many different situations. It also has a solid theoretical foundation in ETS state space models (Hyndman & Athanasopoulos, 2013; Hyndman, Koehler, Ord, & Snyder, 2008; Hyndman, Koehler, Snyder, & Grose, 2002). Here, the acronym *ETS* stands both for ExponenTial Smoothing and for Error, Trend, and Seasonality, which are the three components that define a model within the ETS family.

Exponential smoothing methods obtained competitive results in the M3 forecasting competition (Koning, Franses, Hibon, & Stekler, 2005; Makridakis & Hibon, 2000), and the `forecast` package (Hyndman, 2014; Hyndman & Khandakar, 2008) in the programming language R (R Core Team, 2014) means that a fully automated software for fitting ETS models is available. Thus, ETS models are both usable and highly relevant in practice, and have a solid theoretical foundation, which makes any attempts to improve their forecast accuracy a worthwhile endeavour.

Bootstrap aggregating (bagging), as proposed by Breiman (1996), is a popular method in machine learning for improving the accuracy of predictors (Hastie, Tibshirani, & Friedman, 2009) by addressing potential instabilities. These instabilities typically stem from sources such as data uncertainty, parameter uncertainty, and model selection uncertainty. An ensemble of predictors is estimated on bootstrapped versions of the input data, and the output of the ensemble is calculated by combining (using

* Correspondence to: Faculty of Information Technology, P.O. Box 63 Monash University, Victoria 3800, Australia. Tel.: +61 3 990 59555.
E-mail address: christoph.bergmeir@monash.edu (C. Bergmeir).

http://dx.doi.org/10.1016/j.ijforecast.2015.07.002
0169-2070/© 2015 International Institute of Forecasters. Published by Elsevier B.V. All rights reserved.

the median, mean, trimmed mean, or weighted mean, for example), often yielding better point predictions. In this work, we propose a bagging methodology for exponential smoothing methods, and evaluate it on the M3 data. As our input data are non-stationary time series, both serial dependence and non-stationarity have to be taken into account. We resolve these issues by applying a seasonal-trend decomposition based on loess (STL, Cleveland, Cleveland, McRae, & Terpenning, 1990) and a moving block bootstrap (MBB, see, e.g., Lahiri, 2003) to the residuals of the decomposition.

Specifically, our proposed method of bagging is as follows. After applying a Box–Cox transformation to the data, the series is decomposed into trend, seasonal and remainder components. The remainder component is then bootstrapped using the MBB, the trend and seasonal components are added back in, and the Box–Cox transformation is inverted. In this way, we generate a random pool of similar bootstrapped time series. For each of these bootstrapped time series, we choose a model from among several exponential smoothing models, using the bias-corrected AIC. Then, point forecasts are calculated using each of the different models, and the resulting forecasts are combined using the median.

The only related work that we are aware of is the study by Cordeiro and Neves (2009), who use a sieve bootstrap to perform bagging with ETS models. They use ETS to decompose the data, then fit an AR model to the residuals, and generate new residuals from this AR process. Finally, they fit the ETS model that was used for the decomposition to all of the bootstrapped series. They also test their method on the M3 dataset, and have some success for quarterly and monthly data, but overall, the results are not promising. In fact, the bagged forecasts are often not as good as the original forecasts applied to the original time series. Our bootstrapping procedure works differently, and yields better results. We use STL for the time series decomposition, MBB to bootstrap the remainder, and choose an ETS model for each bootstrapped series. Using this procedure, we are able to outperform the original M3 methods for monthly data in particular.

The rest of the paper is organized as follows. In Section 2, we discuss the proposed methodology in detail. Section 3 presents the experimental setup and the results, and Section 4 concludes the paper.

2. Methods

In this section, we provide a detailed description of the different parts of our proposed methodology, namely exponential smoothing, and the novel bootstrapping procedure involving a Box–Cox transformation, STL decomposition, and the MBB. We illustrate the steps using series M495 from the M3 dataset, which is a monthly series.

2.1. Exponential smoothing

The general idea of exponential smoothing is that recent observations are more relevant for forecasting than older observations, meaning that they should be weighted more highly. Accordingly, simple exponential smoothing, for example, uses a weighted moving average with weights that decrease exponentially.

Starting from this basic idea, exponential smoothing has been expanded to the modelling of different components of a series, such as the trend, seasonality, and remainder components, where the trend captures the long-term direction of the series, the seasonal part captures repeating components of a series with a known periodicity, and the remainder captures unpredictable components. The trend component is a combination of a level term and a growth term. For example, the Holt–Winters purely additive model (i.e., with additive trend and additive seasonality) is defined by the following recursive equations:

$$\begin{aligned}
\ell_t &= \alpha(y_t - s_{t-m}) + (1-\alpha)(\ell_{t-1} + b_{t-1}) \\
b_t &= \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1} \\
s_t &= \gamma(y_t - \ell_{t-1} - b_{t-1}) + (1-\gamma)s_{t-m} \\
\hat{y}_{t+h|t} &= \ell_t + hb_t + s_{t-m+h_m^+}.
\end{aligned}$$

Here, ℓ_t denotes the series level at time t, b_t denotes the slope at time t, s_t denotes the seasonal component of the series at time t, and m denotes the number of seasons in a year. The constants α, β^*, and γ are smoothing parameters in the $[0, 1]$-interval, h is the forecast horizon, and $h_m^+ = \lfloor (h-1) \bmod m \rfloor + 1$.

There is a whole family of ETS models, which can be distinguished by the type of error, trend, and seasonality each uses. In general, the trend can be non-existent, additive, multiplicative, damped additive, or damped multiplicative. The seasonality can be non-existent, additive, or multiplicative. The error can be additive or multiplicative; however, distinguishing between these two options is only relevant for prediction intervals, not point forecasts. Thus, there are a total of 30 models with different combinations of error, trend and seasonality. The different combinations of trend and seasonality are shown in Table 1. For more detailed descriptions, we refer to Hyndman and Athanasopoulos (2013), Hyndman et al. (2008), and Hyndman et al. (2002).

In R, exponential smoothing is implemented in the `ets` function from the `forecast` package (Hyndman, 2014; Hyndman & Khandakar, 2008). The different models are fitted to the data automatically; i.e., the smoothing parameters and initial conditions are optimized using maximum likelihood with a simplex optimizer (Nelder & Mead, 1965). Then, the best model is chosen using the bias-corrected AIC. We note that, of the 30 possible models, 11 can lead to numerical instabilities, and are therefore not used by the `ets` function (see Hyndman & Athanasopoulos, 2013, Section 7.7, for details). Thus, `ets`, as it is used within our bagging procedure, chooses from among 19 different models.

2.2. The Box–Cox transformation

This is a popular transformation for stabilizing the variance of a time series, and was originally proposed by Box and Cox (1964). It is defined as follows:

$$w_t = \begin{cases} \log(y_t), & \lambda = 0; \\ (y_t^\lambda - 1)/\lambda, & \lambda \neq 0. \end{cases}$$

Table 1
The ETS model family, with different types of seasonality and trend.

Trend component	Seasonal component		
	N (None)	A (Additive)	M (Multiplicative)
N (None)	N, N	N, A	N, M
A (Additive)	A, N	A, A	A, M
A_d (Additive damped)	A_d, N	A_d, A	A_d, M
M (Multiplicative)	M, N	M, A	M, M
M_d (Multiplicative damped)	M_d, N	M_d, A	M_d, M

Depending on the parameter λ, the transformation is essentially the identity ($\lambda = 1$), the logarithm ($\lambda = 0$), or a transformation somewhere between. One difficulty is the method of choosing the parameter λ. In this work, we restrict it to lie in the interval $[0, 1]$, then use the method of Guerrero (1993) to choose its value in the following way.

The series is divided into subseries of a length equal to the seasonality, or of length two if the series is not seasonal. Then, the sample mean m and standard deviation s are calculated for each of the subseries, and λ is chosen in such a way that the coefficient of variation of $s/m^{(1-\lambda)}$ across the subseries is minimized.

For the example time series M495, this method gives $\lambda = 6.61 \times 10^{-5}$. Fig. 1 shows the original series and the Box–Cox transformed version using this λ.

2.3. Time series decomposition

For non-seasonal time series, we use the loess method (Cleveland, Grosse, & Shyu, 1992), a smoothing method based on local regressions, to decompose the time series into trend and remainder components. For seasonal time series, we use STL, as presented by Cleveland et al. (1990), to obtain the trend, seasonal and remainder components.

In loess, a neighborhood is defined for each data point, and the points in that neighborhood are then weighted (using so-called *neighborhood weights*) according to their distances from the respective data point. Finally, a polynomial of degree d is fitted to these points. Usually, $d = 1$ and $d = 2$ are used, i.e., linear or quadratic curves are fitted. The trend component is equal to the value of the polynomial at each data point. In R, loess smoothing is available through the function `loess`. For the non-seasonal data in our experiments, i.e., the yearly data from the M3 competition, we use the function with a degree of $d = 1$. In this function, the neighborhood size is defined by a parameter α, which is the proportion of the overall points to include in the neighborhood, with tricubic weighting. To get a constant neighborhood of six data points, we define this parameter to be six divided by the length of the time series under consideration.

In STL, loess is used to divide the time series into their trend, seasonal, and remainder components. The division is additive, i.e., summing the parts gives the original series again. In detail, the steps performed during STL decomposition are: (i) detrending; (ii) cycle-subseries smoothing: series are built for each seasonal component, and smoothed separately; (iii) low-pass filtering of smoothed cycle-subseries: the subseries are put together again, and smoothed; (iv) detrending of the seasonal series; (v) deseasonalizing the original series, using the seasonal component calculated in the previous steps; and (vi) smoothing the deseasonalized series to get the trend component. In R, the STL algorithm is available through the `stl` function. We use it with its default parameters. The degrees for the loess fitting are $d = 1$ in steps (iii) and (iv), and $d = 0$ in step (ii). Fig. 2 shows the STL decomposition of series M495 from the M3 dataset, as an example.

Another possibility for decomposition is to use ETS modelling directly, as was proposed by Cordeiro and Neves (2009). However, the components of an ETS model are defined based on the noise terms, and evolve dynamically with the noise. Thus, "simulating" an ETS process by decoupling the level, trend and seasonal components from the noise and treating them as independent series may not work well. This is in contrast to an STL decomposition, in which the trend and seasonal components are smooth and the way in which they change over time does not depend on the noise component directly. Therefore, we can simulate the noise term independently in an STL decomposition using bootstrapping procedures.

2.4. Bootstrapping the remainder

As time series data are typically autocorrelated, adapted versions of the bootstrap exist (see Gonçalves & Politis, 2011; Lahiri, 2003). One prerequisite is the stationarity of the series, which we achieve by bootstrapping the remainder of the STL (or loess) decomposition.

In the MBB, as originally proposed by Künsch (1989), data blocks of equal size are drawn from the series until the desired series length is achieved. For a series of length n, with a block size of l, $n - l + 1$ (overlapping) possible blocks exist.

We use block sizes of $l = 8$ for yearly and quarterly data, and $l = 24$ for monthly data, i.e., at least two full years, to ensure that any remaining seasonality is captured. As the shortest series for the yearly data has a total of $n = 14$ observations, care must be taken to ensure that every value from the original series could possibly be placed anywhere in the bootstrapped series. To achieve this, we draw $\lfloor n/l \rfloor + 2$ blocks from the remainder series, then discard a random number of values, between zero and $l - 1$, from the beginning of the bootstrapped series. Finally, to obtain a series with the same length as the original series, we discard as many values as necessary to obtain the required length. This processing ensures that the bootstrapped series does not necessarily begin or end on a block boundary.

There are various other methods in the literature for bootstrapping time series, such as the tapered block

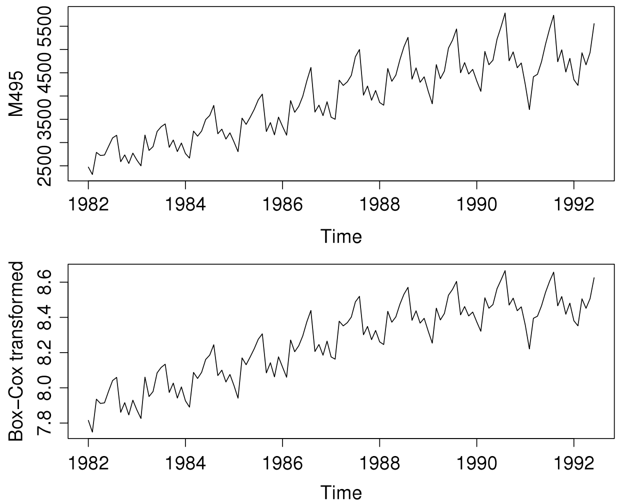

Fig. 1. Series M495 of the M3 dataset, which is a monthly time series. Above is the original series, below the Box–Cox transformed version, with $\lambda = 6.61 \times 10^{-5}$.

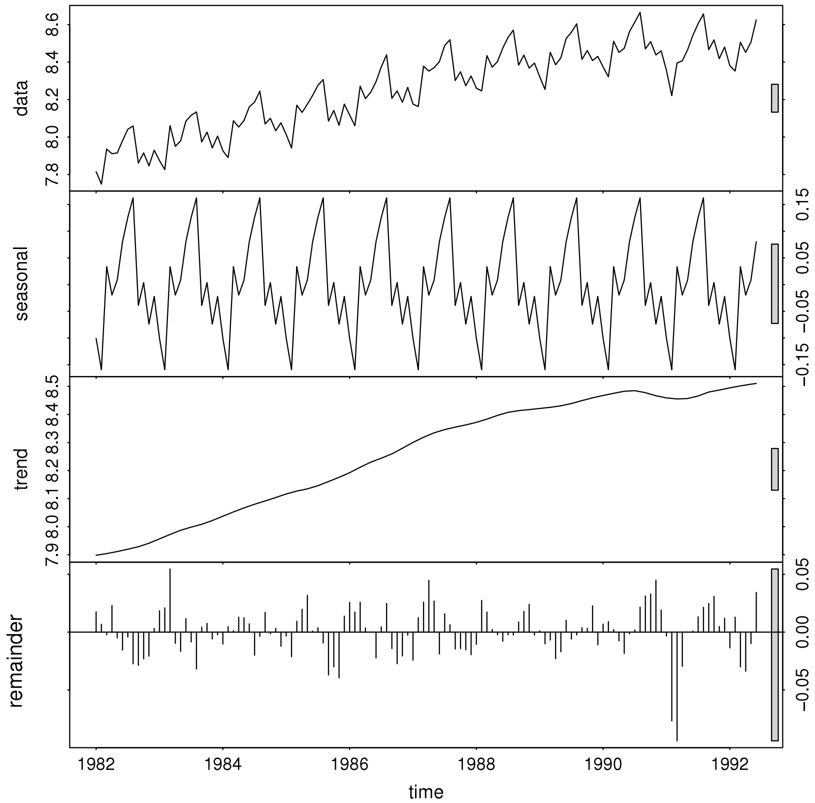

Fig. 2. STL decomposition into the trend, seasonal part, and remainder, of the Box–Cox transformed version of series M495 from the M3 dataset.

bootstrap (Paparoditis & Politis, 2001), the dependent wild bootstrap (DWB, Shao, 2010a), and the extended tapered block bootstrap (Shao, 2010b). However, Shao (2010a) concludes that, "for regularly spaced time series, the DWB is not as widely applicable as the MBB, and the DWB lacks the higher order accuracy property of the MBB". Thus, "the DWB is a complement to, but not a competitor of, existing block-based bootstrap methods". We performed preliminary experiments (which are not reported here) using the tapered block bootstrap and the DWB, but use only the MBB in this paper, as the other procedures did not provide substantial advantages.

Another type of bootstrap is the sieve bootstrap, which was proposed by Bühlmann (1997) and used by Cordeiro and Neves (2009) in an approach similar to ours. Here, the dependence in the data is tackled by fitting a

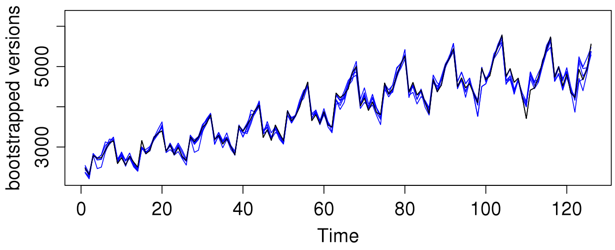

Fig. 3. Bootstrapped versions (blue) of the original series M495 (black). Five bootstrapped series are shown. It can be seen that the bootstrapped series resemble the behavior of the original series quite well. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

model and then bootstrapping the residuals, assuming that they are uncorrelated. This bootstrapping procedure has the disadvantage that one must assume that the model captures all of the relevant information in the time series. The MBB has the advantage that it makes no modelling assumptions other than stationarity, whereas the sieve bootstrap assumes that the fitted model captures all of the serial correlation in the data.

After bootstrapping the remainder, the trend and seasonality are combined with the bootstrapped remainder, and the Box–Cox transformation is inverted, to get the final bootstrappedsample. Fig. 3 gives an illustration of bootstrapped versions of the example series M495.

2.5. The overall procedure

To summarize, the bootstrapping procedure is given in Algorithm 1. Initially, the value of $\lambda \in [0, 1]$ is calculated according to Guerrero (1993). Then, the Box–Cox transformation is applied to the series, and the series is decomposed into the trend, seasonal part, and remainder, using STL or loess. The remainder is then bootstrapped using the MBB, the components are added together again, and the Box–Cox transformation is inverted.

Algorithm 1 Generating bootstrapped series

procedure BOOTSTRAP(ts, num.boot)
    $\lambda \leftarrow$ **BoxCox.lambda**(ts, min=0, max=1)
    ts.bc $\leftarrow$ **BoxCox**(ts, λ)
    if ts is seasonal **then**
        [trend, seasonal, remainder] $\leftarrow$ **stl**(ts.bc)
    else
        seasonal $\leftarrow 0$
        [trend, remainder] $\leftarrow$ **loess**(ts.bc)
    end if
    recon.series[1] $\leftarrow$ ts
    for i in 2 to num.boot **do**
        boot.sample[i] $\leftarrow$ **MBB**(remainder)
        recon.series.bc[i] $\leftarrow$ trend + seasonal + boot.sample[i]
        recon.series[i] $\leftarrow$ **InvBoxCox**(recon.series.bc[i], λ)
    end for
    return recon.series
end procedure

After generating the bootstrapped time series, the ETS model fitting procedure is applied to every series. As was stated in Section 2.1, we use the `ets` function from the `forecast` package (Hyndman, 2014; Hyndman & Khandakar, 2008). The model fits all possible ETS models to the data, then chooses the best model using the bias-corrected AIC. By applying the entire ETS fitting and model selection procedure to each bootstrapped time series independently, we address the issues of data uncertainty, parameter uncertainty, and model selection uncertainty.

For each horizon, the final resulting forecast is calculated from the forecasts from the single models. We performed preliminary experiments using the mean, trimmed mean, and median. However, we restrict our analysis in this study to the median, as it achieves good results and is less sensitive to outliers than the mean, for example, and we also take into account the results of Kourentzes, Barrow, and Crone (2014).

3. Experimental study

In this section, we describe the forecasting methods, error measures, and statistical tests that were used in the experiments, together with the results obtained for the M3 dataset, separately for yearly, quarterly, and monthly data.

3.1. Compared methods

In what follows, we refer to the decomposition approach proposed in this paper, namely the Box–Cox transformation and STL or loess, as *Box–Cox and loess-based decomposition (BLD)*. Bootstrapped versions of the series are generated as was discussed in Section 2, i.e., BLD is followed by the MBB, to generate bootstrapped versions of the series. We use an ensemble size of 100, so that we estimate models on the original time series and on 99 bootstrapped series.

We compare our proposed method both to the original ETS method and to several variants, in the spirit of Cordeiro and Neves (2009). Specifically, we consider all possible combinations of using BLD or ETS for decomposition, and the MBB or a sieve bootstrap for bootstrapping the remainder. Here, the sieve bootstrap is implemented as follows: an ARIMA model is fitted to the remainder of the method used for decomposition (BLD or ETS) using the `auto.arima` function from the `forecast` package (Hyndman, 2014; Hyndman & Khandakar, 2008), which selects a model automatically using the bias-corrected AIC, with model orders of up to five. Then, a normal bootstrapping procedure is applied to the residuals of this

ARIMA model. In particular, the following procedures are employed:

ETS The original exponential smoothing method applied to the original series, selecting one model from among all possible models using the bias-corrected AIC.

Bagged.BLD.MBB.ETS Our proposed method. Specifically, the bootstrapped time series are generated using BLD and MBB. For each of the series thus generated, a model is selected from all exponential smoothing models using the bias-corrected AIC. Then, the forecasts from each of the models are combined using the median.

Bagged.ETS.Sieve.ETS ETS is used for decomposition and the sieve bootstrap, as presented above, is used for bootstrapping the remainder. This approach is very similar to the approach of Cordeiro and Neves (2009). The main differences are that (i) we choose an ETS model for each bootstrapped series, so that this approach accounts for model uncertainty, and (ii) we use an ARIMA process instead of an AR process for the sieve bootstrap.

Bagged.BLD.Sieve.ETS BLD is used for decomposition, and the sieve bootstrap is used for bootstrapping the remainder.

Bagged.ETS.MBB.ETS ETS is used for decomposition, and MBB for bootstrapping the remainder.

3.2. Evaluation methodology

We use the yearly, quarterly, and monthly series from the M3 competition. There are 645 yearly, 756 quarterly, and 1428 monthly series, so that a total of 2829 series are used. We follow the M3 methodology, meaning that we forecast six periods ahead for yearly series, eight periods ahead for quarterly series, and 18 periods ahead for monthly series. The original data, as well as the forecasts of the methods that participated in the competition, are available in the R package `Mcomp` (Hyndman, 2013).

Although the M3 competition took place some time ago, the original submissions to the competition are still competitive and valid benchmarks. To the best of our knowledge, the only result in the literature that reports a better performance than the original contest winners is the recent work of Kourentzes, Petropoulos, and Trapero (2014).

We use the symmetric MAPE (sMAPE) to measure the errors. The sMAPE is defined as

$$\text{sMAPE} = \text{mean}\left(200\frac{|y_t - \hat{y}_t|}{|y_t| + |\hat{y}_t|}\right),$$

where y_t is the true value of the time series y at time t, and $\hat{y}_t$ is the respective forecast. This definition differs slightly from that given by Makridakis and Hibon (2000), as they do not use absolute values in the denominator. However, as the series in the M3 all have strictly positive values, this difference in the definition should not have any effect in practice (except if a method produces negative forecasts).

Furthermore, we also use the mean absolute scaled error (MASE) proposed by Hyndman and Koehler (2006). It is defined as the mean absolute error on the test set, scaled by the mean absolute error of a benchmark method on the training set. The naïve forecast is used as a benchmark, taking into account the seasonality of the data. Thus, the MASE is defined as:

$$\text{MASE} = \frac{\text{mean}\left(|y_t - \hat{y}_t|\right)}{\text{mean}\left(|y_i - y_{i-m}|\right)},$$

where m is the periodicity, which is 1 for yearly data, 4 for quarterly data, and 12 for monthly data. The variable i runs over the training data, and t over the test data.

We calculate the sMAPE and MASE values as averages over all horizons for each series. Then, we calculate the overall means of these measures across series, as well as ranking the forecasting methods for each series and calculating averages of the ranks across series. Calculating the average ranks has the advantage of being more robust to outliers than the overall means.

3.3. Statistical tests of the results

We use the Friedman rank-sum test for multiple comparisons in order to detect statistically significant differences within the methods, and the post-hoc procedure of Hochberg and Rom (1995) for the further analysis of these differences (García, Fernández, Luengo, & Herrera, 2010).[1] The statistical testing is done using the sMAPE measure.

We begin by using the testing framework to determine whether the differences among the proposed and basic models are statistically significant. Then, in the second step, we use the testing framework to compare these models to the methods that originally participated in the M3 competition. A significance level of $\alpha = 0.05$ is used.

3.4. Results on the yearly data

Table 2 shows the results for all methods on the yearly data. The results are ordered by average sMAPE rank. It can be seen that the bagged versions that use BLD for decomposition perform better than the original ETS method, outperforming it consistently on all measures. The bagged versions that use ETS for decomposition perform worse than the original ETS method.

Table 3 shows the results of the first case of statistical testing, where we compare the bagged and ETS methods among themselves. The table shows the p-values adjusted by the post-hoc procedure. The Friedman test has an overall p-value of 5.11×10^{-5}, which is highly significant. The method with the best ranking, in this case Bagged.BLD.Sieve.ETS, is chosen as the control method. We can then see from the table that the differences from the methods using ETS for decomposition are significant at the chosen significance level.

Table 4 shows the results of the further statistical testing, where we compare the bagged and ETS methods with the methods from the M3 competition. The overall result of the Friedman rank sum test is a p-value of

[1] More information can be found on the thematic web site of SCI2S about *Statistical inference in computational intelligence and data mining* http://sci2s.ugr.es/sicidm.

Table 2
Results for the yearly series, ordered by the first column, which is the average rank of sMAPE. The other columns show the mean sMAPE, average rank of MASE, and mean of MASE.

	Rank sMAPE	Mean sMAPE	Rank MASE	Mean MASE
ForcX	12.458	16.480	12.437	2.769
AutoBox2	12.745	16.593	12.757	2.754
RBF	12.772	16.424	12.786	2.720
Flors.Pearc1	12.883	17.205	12.884	2.938
THETA	12.994	16.974	13.016	2.806
ForecastPro	13.050	17.271	13.064	3.026
ROBUST.Trend	13.118	17.033	13.147	2.625
PP.Autocast	13.223	17.128	13.205	3.016
DAMPEN	13.283	17.360	13.256	3.032
COMB.S.H.D	13.384	17.072	13.315	2.876
Bagged.BLD.Sieve.ETS	13.504	17.797	13.523	3.189
Bagged.BLD.MBB.ETS	13.588	17.894	13.601	3.152
SMARTFCS	13.755	17.706	13.783	2.996
ETS	13.867	17.926	13.935	3.215
HOLT	14.057	20.021	14.081	3.182
WINTER	14.057	20.021	14.081	3.182
ARARMA	14.462	18.356	14.551	3.481
B.J.auto	14.481	17.726	14.467	3.165
Flors.Pearc2	14.540	17.843	14.561	3.016
Bagged.ETS.Sieve.ETS	14.715	18.206	14.771	3.173
Auto.ANN	14.837	18.565	14.811	3.058
Bagged.ETS.MBB.ETS	15.051	18.685	15.128	3.231
AutoBox3	15.098	20.877	15.093	3.177
THETAsm	15.109	17.922	15.012	3.006
AutoBox1	15.444	21.588	15.426	3.679
NAIVE2	15.733	17.880	15.638	3.172
SINGLE	15.792	17.817	15.671	3.171

Table 3
Results of statistical testing for yearly data, using the original ETS method and bagged versions of it. Adjusted p-values calculated using the Friedman test with Hochberg's post-hoc procedure are shown. A horizontal line separates the methods that perform significantly worse than the best method from those that do not. The best method is Bagged.BLD.Sieve.ETS, which performs significantly better than either Bagged.ETS.Sieve.ETS or Bagged.ETS.MBB.ETS.

Method	p_{Hoch}
Bagged.BLD.Sieve.ETS	–
ETS	0.438
Bagged.BLD.MBB.ETS	0.438
Bagged.ETS.Sieve.ETS	0.034
Bagged.ETS.MBB.ETS	3.95×10^{-5}

1.59×10^{-10}, which is highly significant. We see that the ForcX method obtains the best ranking, and is used as the control method. The bagged ETS methods using BLD for decomposition are not significantly different, but ETS and the bagged versions using ETS for decomposition perform significantly worse than the control method.

3.5. Results on the quarterly data

Table 5 shows the results for all methods on the quarterly data, ordered by average sMAPE ranks. It can be seen that the proposed bagged method outperforms the original ETS method in terms of average sMAPE ranks, and average ranks and mean MASEs, but not in mean sMAPEs. This may indicate that the proposed method performs better in general, but that there are some individual series where it yields worse sMAPE results.

Table 6 shows the results of statistical testing considering only the ETS and bagged methods. The Friedman test for multiple comparisons results in a p-value of 1.62×10^{-10}, which is highly significant. The method with the

Table 4
Results of statistical testing for yearly data, including our results (printed in boldface) and the original results of the M3. Adjusted p-values calculated using the Friedman test with Hochberg's post-hoc procedure are shown. A horizontal line separates the methods that perform significantly worse than the best method from those that do not.

Method	p_{Hoch}
ForcX	–
AutoBox2	0.516
RBF	0.516
Flors.Pearc1	0.516
THETA	0.516
ForecastPro	0.516
ROBUST.Trend	0.516
PP.Autocast	0.516
DAMPEN	0.496
COMB.S.H.D	0.325
Bagged.BLD.Sieve.ETS	0.180
Bagged.BLD.MBB.ETS	0.117
SMARTFCS	0.040
ETS	0.019
WINTER	0.004
HOLT	0.004
ARARMA	9.27×10^{-5}
B.J.auto	8.06×10^{-5}
Flors.Pearc2	4.44×10^{-5}
Bagged.ETS.Sieve.ETS	6.27×10^{-6}
Auto.ANN	1.47×10^{-6}
Bagged.ETS.MBB.ETS	9.33×10^{-8}
AutoBox3	5.10×10^{-8}
THETAsm	4.63×10^{-8}
AutoBox1	3.40×10^{-10}
NAIVE2	3.15×10^{-12}
SINGLE	1.19×10^{-12}

Table 5
Results for the quarterly series, ordered by the first column, which is the average rank of sMAPE.

	Rank sMAPE	Mean sMAPE	Rank MASE	Mean MASE
THETA	11.792	8.956	11.786	1.087
COMB.S.H.D	12.546	9.216	12.540	1.105
ROBUST.Trend	12.819	9.789	12.821	1.152
DAMPEN	13.067	9.361	13.050	1.126
ForcX	13.179	9.537	13.169	1.155
PP.Autocast	13.207	9.395	13.196	1.128
ForecastPro	13.544	9.815	13.571	1.204
B.J.auto	13.550	10.260	13.551	1.188
RBF	13.561	9.565	13.534	1.173
HOLT	13.575	10.938	13.513	1.225
Bagged.BLD.MBB.ETS	13.716	10.132	13.701	1.219
WINTER	13.723	10.840	13.665	1.217
ARARMA	13.827	10.186	13.786	1.185
AutoBox2	13.874	10.004	13.920	1.185
Flors.Pearc1	13.881	9.954	13.888	1.184
ETS	14.091	9.864	14.128	1.225
Bagged.BLD.Sieve.ETS	14.161	10.026	14.204	1.241
Auto.ANN	14.317	10.199	14.337	1.241
THETAsm	14.570	9.821	14.546	1.211
SMARTFCS	14.574	10.153	14.629	1.226
Flors.Pearc2	14.761	10.431	14.824	1.255
AutoBox3	14.823	11.192	14.763	1.272
AutoBox1	15.048	10.961	15.055	1.331
SINGLE	15.118	9.717	15.093	1.229
NAIVE2	15.296	9.951	15.290	1.238
Bagged.ETS.Sieve.ETS	15.687	10.707	15.706	1.351
Bagged.ETS.MBB.ETS	15.696	10.632	15.737	1.332

Table 6
Results of statistical testing for quarterly data, using the original ETS method and bagged versions of it. The best method is Bagged.BLD.MBB.ETS, which performs significantly better than either Bagged.ETS.Sieve.ETS or Bagged.ETS.MBB.ETS.

Method	p_{Hoch}
Bagged.BLD.MBB.ETS	–
ETS	0.354
Bagged.BLD.Sieve.ETS	0.147
Bagged.ETS.Sieve.ETS	6.94×10^{-7}
Bagged.ETS.MBB.ETS	5.00×10^{-8}

best ranking is the proposed method, Bagged.BLD.MBB.ETS. We can see from the table that the differences from the methods using ETS for decomposition are statistically significant, but those from the original ETS method are not.

Table 7 shows the results of further statistical testing of the bagged and ETS methods against the methods from the original M3 competition. The overall result of the Friedman rank sum test is a p-value of 1.11×10^{-10}, which is highly significant. We see from the table that the THETA method performs best and is chosen as the control method. It statistically significantly outperforms all methods but COMB.S.H.D.

3.6. Results on the monthly data

Table 8 shows the results for all methods on the monthly data, ordered by average sMAPE rank. The bagged versions using BLD for decomposition again outperform the original ETS method. Furthermore, Bagged.BLD.MBB.ETS also consistently outperforms all of the original methods from the M3 on all measures.

Table 9 shows the results of statistical testing considering only the bagged and ETS methods. The Friedman test gives a p-value of 5.02×10^{-10}, meaning that the differences are highly significant. The method with the best ranking is Bagged.BLD.MBB.ETS, and we can see from the

Table 7
Results of statistical testing for quarterly data, including our results (printed in boldface) and the original results of the M3. A horizontal line separates the methods that perform significantly worse than the best method from those that do not. We see that only the COMB.S.H.D does not have a worse statistical significance than the THETA method.

Method	p_{Hoch}
THETA	–
COMB.S.H.D	0.065
ROBUST.Trend	0.024
DAMPEN	0.005
ForcX	0.003
PP.Autocast	0.003
B.J.auto	1.07×10^{-4}
RBF	1.07×10^{-4}
HOLT	1.07×10^{-4}
Bagged.BLD.MBB.ETS	2.46×10^{-5}
WINTER	2.46×10^{-5}
ARARMA	7.50×10^{-6}
AutoBox2	4.46×10^{-6}
Flors.Pearc1	4.37×10^{-6}
ETS	2.68×10^{-7}
Bagged.BLD.Sieve.ETS	1.04×10^{-7}
Auto.ANN	1.06×10^{-8}
THETAsm	1.83×10^{-10}
SMARTFCS	1.81×10^{-10}
Flors.Pearc2	7.15×10^{-12}
AutoBox3	2.40×10^{-12}
AutoBox1	3.34×10^{-14}
SINGLE	8.57×10^{-15}
NAIVE2	2.25×10^{-16}
Bagged.ETS.Sieve.ETS	3.61×10^{-20}
Bagged.ETS.MBB.ETS	3.02×10^{-20}

Table 8
Results for the monthly series, ordered by the first column, which is the average rank of sMAPE.

	Rank sMAPE	Mean sMAPE	Rank MASE	Mean MASE
Bagged.BLD.MBB.ETS	11.714	13.636	11.725	0.846
THETA	11.992	13.892	11.932	0.858
ForecastPro	12.035	13.898	12.064	0.848
Bagged.BLD.Sieve.ETS	12.059	13.734	12.073	0.870
Bagged.ETS.Sieve.ETS	13.079	13.812	12.990	0.888
COMB.S.H.D	13.083	14.466	13.134	0.896
ETS	13.112	14.286	13.150	0.889
Bagged.ETS.MBB.ETS	13.180	13.873	13.116	0.870
HOLT	13.312	15.795	13.276	0.909
ForcX	13.374	14.466	13.415	0.894
WINTER	13.650	15.926	13.631	1.165
RBF	13.842	14.760	13.861	0.910
DAMPEN	14.118	14.576	14.175	0.908
AutoBox2	14.250	15.731	14.294	1.082
B.J.auto	14.278	14.796	14.290	0.914
AutoBox1	14.333	15.811	14.335	0.924
Flors.Pearc2	14.492	15.186	14.525	0.950
SMARTFCS	14.495	15.007	14.399	0.919
Auto.ANN	14.528	15.031	14.561	0.928
ARARMA	14.715	15.826	14.720	0.907
PP.Autocast	14.785	15.328	14.862	0.994
AutoBox3	14.892	16.590	14.801	0.962
Flors.Pearc1	15.213	15.986	15.211	1.008
THETAsm	15.292	15.380	15.285	0.950
ROBUST.Trend	15.446	18.931	15.353	1.039
SINGLE	15.940	15.300	16.004	0.974
NAIVE2	16.790	16.891	16.819	1.037

Table 9
Results of statistical testing for monthly data, using the original ETS method and bagged versions of it. The best method is Bagged.BLD.MBB.ETS, with performs significantly better than Bagged.ETS.Sieve.ETS, Bagged.ETS.MBB.ETS, and the original ETS method.

Method	p_{Hoch}
Bagged.BLD.MBB.ETS	–
Bagged.BLD.Sieve.ETS	0.338
Bagged.ETS.Sieve.ETS	3.70×10^{-6}
ETS	4.32×10^{-8}
Bagged.ETS.MBB.ETS	5.17×10^{-11}

table that it statistically significantly outperforms both the original method and methods using ETS for decomposition.

Table 10 shows the results of statistical testing of the bagged and ETS methods against the methods from the original M3 competition. The overall result of the Friedman rank sum test is a p-value of 2.92×10^{-10}, meaning that it is highly significant. We see from the table that the proposed method, Bagged.BLD.MBB.ETS, is the best method, and that only the THETA method, ForecastPro, and Bagged.BLD.Sieve.ETS are not significantly worse at the chosen 5% significance level.

4. Conclusions

In this work, we have presented a novel method of bagging for exponential smoothing methods, using a Box–Cox transformation, STL decomposition, and a moving block bootstrap. The method is able to outperform the basic exponential smoothing methods consistently. These results are statistically significant in the case of the monthly series, but not for the yearly or quarterly series. This may be because the longer monthly series allow for tests with greater power, while the quarterly and yearly series are too short for the differences to be significant.

Table 10
Results of statistical testing for monthly data, including our results (printed in boldface) and the original results of the M3. Bagged.BLD.MBB.ETS performs best, and only the THETA, ForecastPro, and Bagged.BLD.Sieve.ETS methods do not perform significantly worse.

Method	p_{Hoch}
Bagged.BLD.MBB.ETS	–
THETA	0.349
ForecastPro	0.349
Bagged.BLD.Sieve.ETS	0.349
Bagged.ETS.Sieve.ETS	1.71×10^{-5}
COMB.S.H.D	1.71×10^{-5}
ETS	1.50×10^{-5}
Bagged.ETS.MBB.ETS	5.56×10^{-6}
HOLT	5.93×10^{-7}
ForcX	2.03×10^{-7}
WINTER	7.05×10^{-10}
RBF	8.45×10^{-12}
DAMPEN	6.97×10^{-15}
AutoBox2	1.76×10^{-16}
B.J.auto	8.27×10^{-17}
AutoBox1	1.74×10^{-17}
Flors.Pearc2	1.37×10^{-19}
SMARTFCS	1.29×10^{-19}
Auto.ANN	4.81×10^{-20}
ARARMA	1.02×10^{-22}
PP.Autocast	9.18×10^{-24}
AutoBox3	2.15×10^{-25}
Flors.Pearc1	1.10×10^{-30}
THETAsm	4.70×10^{-32}
ROBUST.Trend	7.73×10^{-35}
SINGLE	1.50×10^{-44}
NAIVE2	4.53×10^{-64}

Furthermore, on the monthly data from the M3 competition, the bagged exponential smoothing method is able to outperform all methods that took part in the competition, most of them statistically significantly. Thus, this method can be recommended for routine practical application, especially for monthly data.

Acknowledgments

This work was performed while C. Bergmeir held a scholarship from the Spanish Ministry of Education (MEC) of the "Programa de Formación del Profesorado Universitario (FPU)" (AP2008-04637), and was visiting the Department of Econometrics and Business Statistics, Monash University, Melbourne, Australia. This work was supported by the Spanish National Research Plan Project TIN-2013-47210-P and the Andalusian Research Plan P12-TIC-2985. Furthermore, we would like to thank X. Shao for providing code for his bootstrapping procedures, and F. Petropoulos for helpful discussions.

References

Box, G. E. P., & Cox, D. R. (1964). An analysis of transformations. *Journal of the Royal Statistical Society: Series B*, *26*(2), 211–252.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, *24*(2), 123–140.

Bühlmann, P. (1997). Sieve bootstrap for time series. *Bernoulli*, *3*(2), 123–148.

Cleveland, R. B., Cleveland, W. S., McRae, J., & Terpenning, I. (1990). STL: A seasonal-trend decomposition procedure based on loess. *Journal of Official Statistics*, *6*, 3–73.

Cleveland, W. S., Grosse, E., & Shyu, W. M. (1992). Local regression models. In *Statistical models in S*. Chapman & Hall/CRC (Chapter 8).

Cordeiro, C., & Neves, M. (2009). Forecasting time series with BOOT.EXPOS procedure. *REVSTAT—Statistical Journal*, *7*(2), 135–149.

García, S., Fernández, A., Luengo, J., & Herrera, F. (2010). Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. *Information Sciences*, *180*(10), 2044–2064.

Gonçalves, S., & Politis, D. (2011). Discussion: Bootstrap methods for dependent data: A review. *Journal of the Korean Statistical Society*, *40*(4), 383–386.

Goodwin, P. (2010). The Holt–Winters approach to exponential smoothing: 50 years old and going strong. *Foresight: The International Journal of Applied Forecasting*, *19*, 30–33.

Guerrero, V. (1993). Time-series analysis supported by power transformations. *Journal of Forecasting*, *12*, 37–48.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: data mining, inference, and prediction*. Springer.

Hochberg, Y., & Rom, D. (1995). Extensions of multiple testing procedures based on Simes' test. *Journal of Statistical Planning and Inference*, *48*(2), 141–152.

Hyndman, R.J. (2013). Mcomp: Data from the M-competitions. URL: http://robjhyndman.com/software/mcomp/.

Hyndman, R.J. (2014). Forecast: Forecasting functions for time series and linear models. R package version 5.6. URL: http://CRAN.R-project.org/package=forecast.

Hyndman, R., & Athanasopoulos, G. (2013). *Forecasting: principles and practice*. URL: http://otexts.com/fpp/.

Hyndman, R., & Khandakar, Y. (2008). Automatic time series forecasting: The forecast package for R. *Journal of Statistical Software*, *27*(3), 1–22.

Hyndman, R., & Koehler, A. (2006). Another look at measures of forecast accuracy. *International Journal of Forecasting*, *22*(4), 679–688.

Hyndman, R. J., Koehler, A. B., Ord, J. K., & Snyder, R. D. (2008). *Forecasting with exponential smoothing: the state space approach*. Springer, URL: http://www.exponentialsmoothing.net.

Hyndman, R., Koehler, A., Snyder, R., & Grose, S. (2002). A state space framework for automatic forecasting using exponential smoothing methods. *International Journal of Forecasting*, *18*(3), 439–454.

Koning, A., Franses, P., Hibon, M., & Stekler, H. (2005). The M3 competition: Statistical tests of the results. *International Journal of Forecasting*, *21*(3), 397–409.

Kourentzes, N., Barrow, D., & Crone, S. (2014a). Neural network ensemble operators for time series forecasting. *Expert Systems with Applications*, *41*(9), 4235–4244.

Kourentzes, N., Petropoulos, F., & Trapero, J. (2014b). Improving forecasting by estimating time series structural components across multiple frequencies. *International Journal of Forecasting*, *30*(2), 291–302.

Künsch, H. R. (1989). The jackknife and the bootstrap for general stationary observations. *Annals of Statistics*, *17*(3), 1217–1241.

Lahiri, S. (2003). *Resampling methods for dependent data*. Springer.

Makridakis, S., & Hibon, M. (2000). The M3-competition: Results, conclusions and implications. *International Journal of Forecasting*, *16*(4), 451–476.

Nelder, J., & Mead, R. (1965). A simplex method for function minimization. *The Computer Journal*, *7*, 308–313.

Paparoditis, E., & Politis, D. (2001). Tapered block bootstrap. *Biometrika*, *88*(4), 1105–1119.

R Core Team (2014). *R: a language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing, URL: http://www.R-project.org/.

Shao, X. (2010a). The dependent wild bootstrap. *Journal of the American Statistical Association*, *105*(489), 218–235.

Shao, X. (2010b). Extended tapered block bootstrap. *Statistica Sinica*, *20*(2), 807–821.

Christoph Bergmeir received the M.Sc. degree in Computer Science from the University of Ulm, Germany, in 2008, and the Ph.D. degree from the University of Granada, Spain, in 2013. He is currently working at the Faculty of Information Technology, Monash University, Melbourne, Australia, as a Research Fellow in Applied Artificial Intelligence. His research interests include time series predictor evaluation, meta-learning and forecast combination, and time series complexity. He has published in journals such as *IEEE Transactions on Neural Networks and Learning Systems*, *Journal of Statistical Software*, *Computer Methods and Programs in Biomedicine*, and *Information Sciences*.

Rob J. Hyndman is Professor of Statistics in the Department of Econometrics and Business Statistics at Monash University and Director of the Monash University Business & Economic Forecasting Unit. He is also Editor-in-Chief of the *International Journal of Forecasting* and a Director of the International Institute of Forecasters. Rob is the author of over 100 research papers in statistical science. In 2007, he received the Moran medal from the Australian Academy of Science for his contributions to statistical research, especially in the area of statistical forecasting. For 30 years, Rob has maintained an active consulting practice, assisting hundreds of companies and organizations. His recent consulting work has involved forecasting electricity demand, tourism demand, the Australian government health budget and case volume at a US call centre.

José Manuel Benítez (M'98) received the M.S. and Ph.D. degrees in Computer Science both from the Universidad de Granada, Spain. He is currently an Associate Professor at the Department of Computer Science and Artificial Intelligence, Universidad de Granada. He is the head of the Distributed Computational Intelligence and Time Series (DiCITS) lab. His research interests include Cloud Computing and Big Data, Data Science, Computational Intelligence and Time Series. He has published in the leading journals of the "Artificial Intelligence" and Computer Science field. He has led a number of research projects funded by different international and national organizations as well as research contracts with leading international corporations.