ISSN 1440-771X

MONASH University

Department of Econometrics and Business Statistics

http://www.buseco.monash.edu.au/depts/ebs/pubs/wpapers/

Bagging Exponential Smoothing Methods using STL Decomposition and Box-Cox Transformation

Christoph Bergmeir, Rob J Hyndman,
José M Benítez

March 2014

Working Paper 11/14

Bagging Exponential Smoothing Methods using STL Decomposition and Box-Cox Transformation

Christoph Bergmeir
Department of Computer Science and Artificial Intelligence,
E.T.S. de Ingenierías Informática y de Telecomunicación,
University of Granada, Spain.
Email: c.bergmeir@decsai.ugr.es

Rob J Hyndman
Department of Econometrics and Business Statistics,
Monash University, VIC 3800 Australia.

José M Benítez
Department of Computer Science and Artificial Intelligence,
E.T.S. de Ingenierías Informática y de Telecomunicación,
University of Granada, Spain.

21 March 2014

JEL classification: C22, C53, C63

Bagging Exponential Smoothing Methods using STL Decomposition and Box-Cox Transformation

Abstract

Exponential smoothing is one of the most popular forecasting methods. We present a method for bootstrap aggregation (bagging) of exponential smoothing methods. The bagging uses a Box-Cox transformation followed by an STL decomposition to separate the time series into 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. On the bootstrapped series, an ensemble of exponential smoothing models is estimated. The resulting point forecasts are averaged using the mean. We evaluate this new method on the M3 data set, showing that it consistently outperforms the original exponential smoothing models. On the monthly data, we achieve better results than any of the original M3 participants. We also perform statistical testing to explore significance of the results. Using the MASE, our method is significantly better than all the M3 participants on the monthly data.

Keywords: bagging, bootstrapping, exponential smoothing, STL decomposition.

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). Reasons for this are its simplicity and transparency, as well as its ability to adapt to many situations. It also has a solid theoretical foundation in ETS state-space models (Hyndman et al, 2002, 2008; Hyndman and Athanasopoulos, 2013).

In the M3 forecasting competition (Makridakis and Hibon, 2000; Koning et al, 2005), exponential smoothing methods obtained competitive results, and with the `forecast` package (Hyndman and Khandakar, 2008) in the programming language R (R Core Team, 2014), a fully automatic software for fitting ETS models is available. So, ETS models are usable, 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 to improve the accuracy of predictors (Hastie et al, 2009; Zhang and Zhang, 2009). An ensemble of predictors is estimated on bootstrapped versions of the input data, and the output of the ensemble is calculated by averaging (mean, median, trimmed mean, weighted averaging, etc.), 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 STL decomposition (Cleveland et al, 1990), and a moving block bootstrap (MBB, see, e.g., Lahiri, 2003) on 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, and the Box-Cox transformation is inverted. In this way, we generate a random pool of similar bootstrapped time series. For each one of these bootstrapped time series, we choose a model among several exponential smoothing models, using the bias-corrected AIC. Then, point forecasts are calculated using all the different models, and the resulting forecasts are averaged.

The only related work we are aware of is the work of Cordeiro and Neves (2009) who use a sieve bootstrap to perform bagging with ETS models. They fit an ETS model to the data, then fit an AR model to the residuals, and generate new residuals from this AR process. They also test their method on the M3 dataset. They have some success on quarterly and monthly data, but the overall 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. In particular, we are able to outperform the original M3 methods in monthly data.

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 describe in detail the different parts of our proposed methodology, which are 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

Exponential smoothing divides the time series into seasonal, trend, and error component. For example, the Holt-Winters purely additive model 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 $h_m^+ = [(h-1) \bmod m] + 1$.

Nowadays, there is a whole family of ETS models, with a solid state space foundation. The models can be distinguished by the type of error, trend, and seasonality they use. Distinguishing between additive and multiplicative error only has consequences for the prediction intervals, not for the point forecasts. In the example above, trend and seasonality are additive. In general, the trend can be non-existent, additive, multiplicative, or damped additive/multiplicative. The seasonality can be non-existent, additive, or multiplicative. So, in total there are 30 models with different combinations of error, trend and seasonality. For more detailed descriptions, we refer to Hyndman et al (2002) and Hyndman et al (2008).

In R, exponential smoothing is implemented in the ets function from the forecast package (Hyndman and Khandakar, 2008). All the different models are fitted automatically to the data; i.e., the smoothing parameters and initial conditions are optimized using maximum likelihood with a simplex optimizer (Nelder and Mead, 1965). Then, the best model from the model family is chosen using the bias-corrected AIC.

2.2 The Box-Cox transformation

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

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

Depending on a parameter λ, the transformation is essentially the identity ($\lambda = 1$), the logarithm, ($\lambda = 0$), or a transformation somewhere between. A difficulty is the way to choose the parameter λ. In this work, we restrict it to lie in the $[0, 1]$-interval, and then use the method of Guerrero (1993) to choose its value. For the example time series M495, this method gives $\lambda = 6.61 \times 10^{-5}$. Figure 1 shows the original series and the Box-Cox transformed version using this λ.

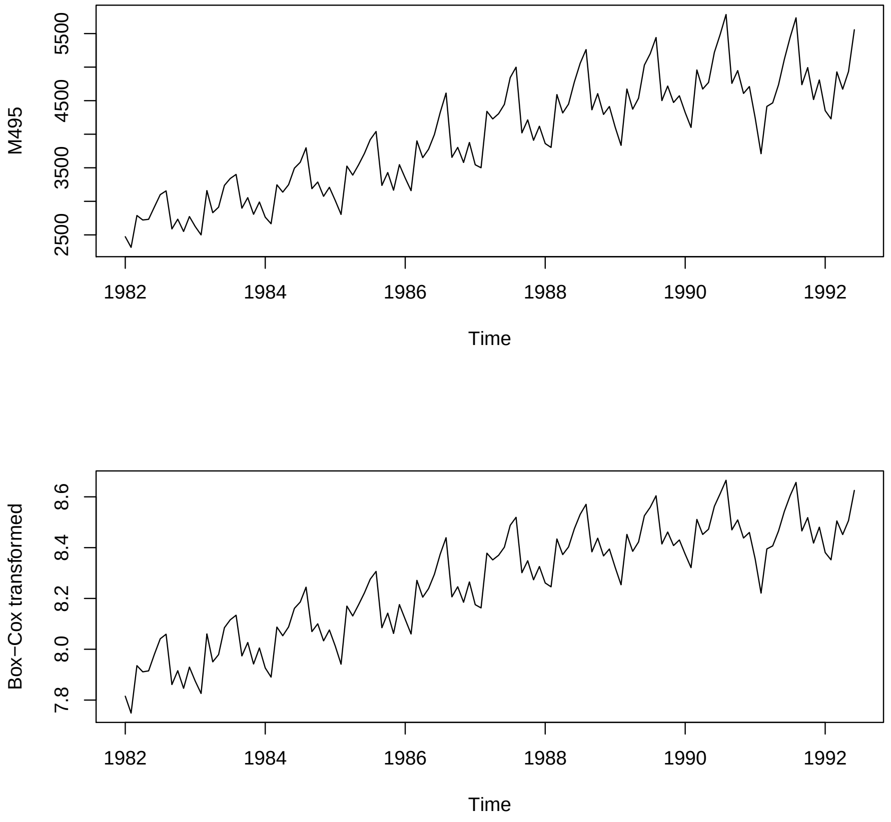

Figure 1: *Series M495 of the M3 dataset, which is a monthly time series. Above is the original series, below the Box-Cox transformed version, with* $\lambda = 6.61 \times 10^{-5}$.

2.3 Time series decomposition

For non-seasonal time series, we use the loess method (Cleveland et al, 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 decomposition (seasonal-trend decomposition based on loess), as presented by Cleveland et al (1990), to obtain trend, seasonal and remainder components.

In loess, for each data point a neighborhood is defined. The points in that neighborhood are then weighted (using so-called *neighborhood weights*) according to their distance 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 by the function `loess`. For the non-seasonal data of our experiments, i.e., the yearly data of the M3 competition, we use the function with a degree of $d = 1$. The neighborhood size is defined in this function by a parameter α, which is the proportion of overall points to include in the neighborhood, with tricubic weighting. To get a constant neighborhood of 6 data points, we define this parameter to be 6 divided by the length of the time series in consideration.

In STL, loess is used to divide the time series into 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 sub-series 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 in steps (iii) and (iv) are $d = 1$, and $d = 0$ in step (ii). Figure 2 shows the STL decomposition of series M495 of the M3 dataset, as an example.

2.4 Bootstrapping the remainder

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

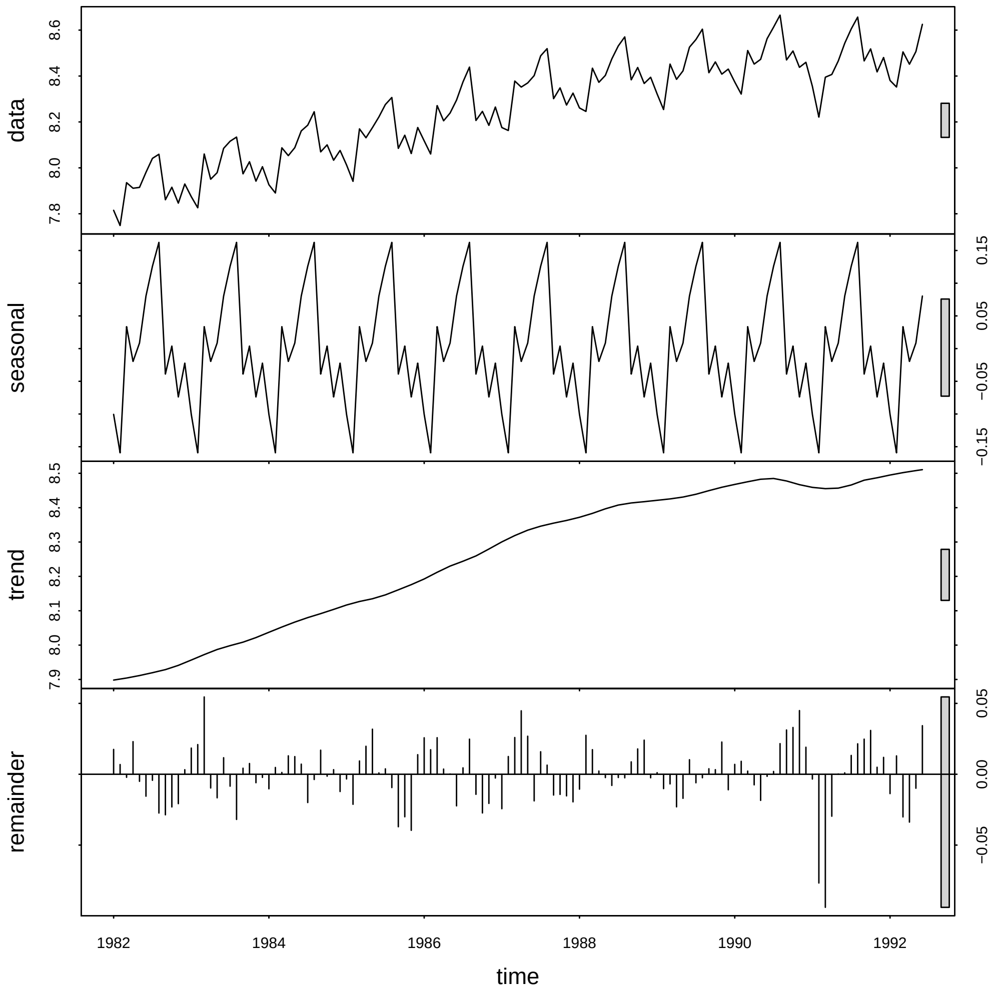

Figure 2: *STL decomposition into trend, seasonal part, and remainder, of the Box-Cox transformed version of series M495 of the M3 dataset.*

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 any remaining seasonality is captured. As the shortest series we use have $n = 14$ observations in total for the yearly data, care has to be taken that every value from the original series can possibly be placed anywhere in the bootstrapped series. To achieve this, we draw $\lfloor n/l \rfloor + 2$ blocks from the remainder series. Then, we discard from the beginning of the bootstrapped series a random amount between zero and $l-1$ values. Finally, to obtain a series with the same length as the original series, we discard the amount of values necessary to obtain the required length. This processing ensures that the bootstrapped series does not necessarily begin or end on a block boundary.

In the literature, there exist various other methods for bootstrapping time series, such as the tapered block bootstrap (Paparoditis and 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." So that "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 finally decided to use the MBB only, as the other procedures where not giving substantial advantages.

Another type of bootstrap is the sieve bootstrap as 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 model and later bootstrapping the residuals, assuming they are uncorrelated. This bootstrapping procedure has the disadvantage that it has to be assumed that the model captures all relevant information in the time series. We performed experiments (also unreported) using an ETS model instead of the STL procedure, and using an AR process for bootstrapping instead of the MBB. However, the results were consistently worse than with the procedure presented here.

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 bootstrapped sample. Figure 3 gives an illustration of bootstrapped versions of the example series M495.

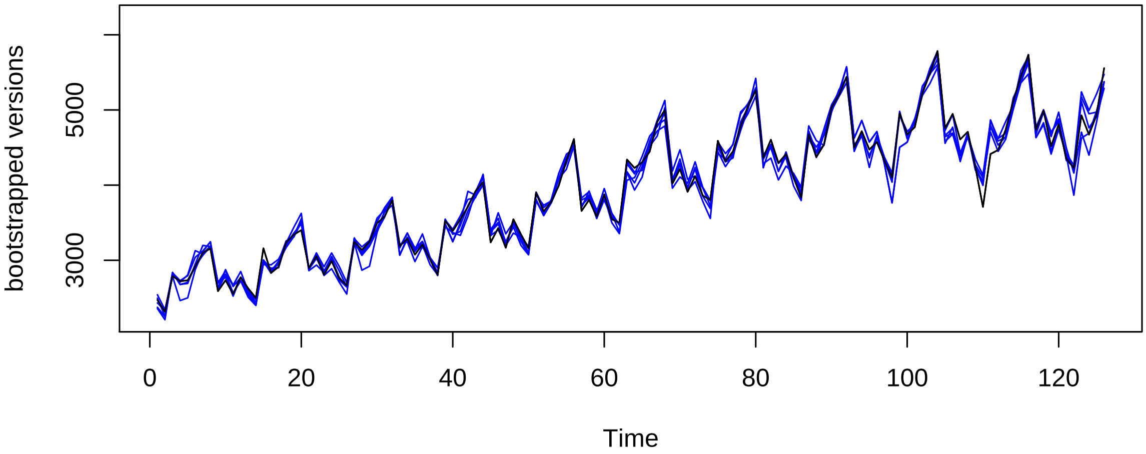

Figure 3: *Bootstrapped versions (blue) of the original series M495 (black). Five bootstrapped series are shown. It can be seen that the bootstrapped series resemble the behavior of the original series quite well.*

2.5 The overall procedure

To summarize, a scheme of the bootstrapping procedure is given in Algorithm 1. At first, λ is calculated according to Guerrero (1993). Then, the Box-Cox transformation is applied to the series, and the series is decomposed into 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)
    $\lambda$ ← BoxCox.lambda(ts)        ▷ use automatic procedure to determine $\lambda$
    ts.bc ← BoxCox(ts, $\lambda$)
    if ts is seasonal then
        [trend, seasonal, remainder] ← stl(ts.bc)
    else
        seasonal ← 0
        [trend, remainder] ← loess(ts.bc)
    end if
    recon.series[1] ← ts        ▷ add the original series as a sample
    for i in 2 to num.boot do
        boot.sample[i] ← MBB(remainder)
        recon.series.bc[i] ← trend + seasonal + boot.sample[i]
        recon.series[i] ← InvBoxCox(recon.series.bc[i], $\lambda$)
    end for
    return recon.series
end procedure
```

Then, to every bootstrapped version of the time series, an ETS model is fitted and used for forecasting. For each horizon, the final resulting forecast is calculated as the mean over the forecasts from the single models.

3 Experimental study

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

3.1 Compared methods

We use the `ets` function from the `forecast` package (Hyndman and Khandakar, 2008) for model building. Bootstrapped versions of the series are generated as discussed in Section 2.

We use an ensemble size of 30, so that we estimate models on the original time series and on 29 bootstrapped series. We compare two different model fitting approaches. The first one fits both additive and multiplicative models, the second one uses the Box-Cox transformation with the same value for λ as in the bagging procedure to handle non-additive aspects of the time series, and then the models can be restricted to additive models only. For comparison, both approaches are used in bagged and non-bagged versions. So, in total, the following procedures are employed:

ETS the original exponential smoothing method choosing among all possible models using the bias-corrected AIC.

AdditiveETS.BC exponential smoothing, choosing only from the additive models, and using a Box-Cox transformation with the same value for λ as in the bootstrapping.

BaggedETS for each of the 30 series, a model is chosen from all exponential smoothing models using the bias-corrected AIC. Then, for the forecasts, the outcome of all methods is averaged using the mean.

BaggedETS.BC the Box-Cox transformation with the same value for λ as in the bootstrapping is used during model estimation, and the models are restricted to additive models only.

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 in total 2829 series are used. As forecast horizons, we follow the M3 methodology, so that we forecast 6 values for yearly series, 8 values for quarterly series, and 18 values 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).

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 is slightly different from the definition given by Makridakis and Hibon (2000), as they do not use absolute values in the denominator. However, as the series in the M3 have all strictly positive values, this difference in the definition should not have any effect in practice (only if a method forecasts negative values).

Furthermore, we use also the mean absolute scaled error (MASE) as 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. As benchmark, the naïve forecast is used, taking into account the seasonality of the data. So, the MASE is defined as:

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

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, t over the test data.

We calculate the sMAPE and MASE as averages over all horizons per series. Then, we calculate overall means of these measures across series, and also rank the forecasting methods for each series and calculate 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 aligned Friedman rank-sum test for multiple comparisons to detect statistically significant differences within the methods, and the post-hoc procedure of Hochberg and Rom (1995) to further analyze those differences (García et al, 2010)[1]. The statistical testing is done using the sMAPE measure.

At first, we use the testing framework to determine if the differences among the proposed models and the basis models are statistically significant. Then, in a second step, we use the bagged exponential smoothing method and compare it with the testing framework 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 1 shows the results for all methods on the yearly data. The results are ordered by average rank of sMAPE. It can be seen that the BaggedETS method performs better than the other

[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

	Rank sMAPE	Mean sMAPE	Rank MASE	Mean MASE
ForcX	12.022	16.480	12.007	2.769
AutoBox2	12.370	16.593	12.388	2.754
RBF	12.381	16.424	12.389	2.720
Flors.Pearc1	12.471	17.205	12.476	2.938
THETA	12.530	16.974	12.557	2.806
ForecastPro	12.615	17.271	12.636	3.026
ROBUST.Trend	12.769	17.033	12.809	2.625
PP.Autocast	12.800	17.128	12.788	3.016
DAMPEN	12.841	17.360	12.828	3.032
BaggedETS	12.853	17.492	12.840	2.973
COMB.S.H.D	12.960	17.072	12.910	2.876
SMARTFCS	13.310	17.706	13.333	2.996
AdditiveETS.BC	13.333	18.357	13.389	3.557
BaggedETS.BC	13.408	18.258	13.411	3.478
HOLT	13.554	20.021	13.565	3.182
WINTER	13.554	20.021	13.565	3.182
ETS	13.884	18.655	13.963	3.532
ARARMA	14.025	18.356	14.129	3.481
Flors.Pearc2	14.029	17.843	14.054	3.016
B.J.auto	14.068	17.726	14.041	3.165
Auto.ANN	14.420	18.565	14.394	3.058
AutoBox3	14.602	20.877	14.592	3.177
THETAsm	14.684	17.922	14.598	3.006
AutoBox1	14.931	21.588	14.930	3.679
NAIVE2	15.270	17.880	15.195	3.172
SINGLE	15.316	17.817	15.212	3.171

Table 1: *Results for the yearly series, ordered by the first column, which is the average rank of sMAPE. The other columns show mean sMAPE, average rank of MASE, and mean of MASE.*

methods, consistently outperforming them in all measures. The original ETS method does not perform particularly well on this dataset.

Table 2 shows the results of the first case of statistical testing, where we compare the ETS methods among themselves. The table shows the p-values adjusted by the post-hoc procedure. The aligned Friedman test has an overall p-value of 2.17×10^{-10}, which is highly significant. The method with the best ranking, BaggedETS in this case, is chosen as the control method. We can see then from the table that differences against all comparison methods are significant on the chosen significance level.

Table 3 shows the results of the second case of statistical testing, where we choose BaggedETS as the best method and compare it against the methods from the M3 competition. The overall result of the Friedman aligned rank sum test is a p-value of 2.46×10^{-10}, which is highly significant.

Method	p_{Hoch}
BaggedETS	–
AdditiveETS.BC	0.023
BaggedETS.BC	0.023
ETS	0.002

Table 2: *Results of statistical testing for yearly data, using only exponential smoothing methods. Adjusted p-values calculated from the aligned 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 the ones that do not.*

Method	p_{Hoch}
ForcX	–
AutoBox2	0.598
RBF	0.598
ROBUST.Trend	0.598
THETA	0.598
ForecastPro	0.598
Flors.Pearc1	0.598
PP.Autocast	0.574
DAMPEN	0.344
BaggedETS	0.344
COMB.S.H.D	0.085
SMARTFCS	0.056
ARARMA	0.001
B.J.auto	5.20×10^{-4}
WINTER	5.16×10^{-5}
HOLT	5.16×10^{-5}
Flors.Pearc2	5.16×10^{-5}
Auto.ANN	1.37×10^{-5}
AutoBox3	2.81×10^{-7}
SINGLE	7.79×10^{-8}
THETAsm	6.94×10^{-8}
NAIVE2	5.69×10^{-8}
AutoBox1	2.62×10^{-11}

Table 3: *Results of statistical testing for yearly data, using BaggedETS and the original results of the M3. Adjusted p-values calculated from the aligned 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 the ones that do not.*

	Rank sMAPE	Mean sMAPE	Rank MASE	Mean MASE
THETA	11.493	8.956	11.483	1.087
COMB.S.H.D	12.196	9.216	12.187	1.105
ROBUST.Trend	12.489	9.789	12.500	1.152
DAMPEN	12.692	9.361	12.697	1.126
ForcX	12.843	9.537	12.847	1.155
PP.Autocast	12.847	9.395	12.866	1.128
B.J.auto	13.161	10.260	13.169	1.188
ForecastPro	13.192	9.815	13.208	1.204
HOLT	13.236	10.938	13.171	1.225
RBF	13.267	9.565	13.251	1.173
WINTER	13.413	10.840	13.355	1.217
BaggedETS	13.425	10.030	13.438	1.192
ARARMA	13.426	10.186	13.393	1.185
BaggedETS.BC	13.493	10.059	13.523	1.209
Flors.Pearc1	13.500	9.954	13.511	1.184
AutoBox2	13.513	10.004	13.552	1.185
AdditiveETS.BC	13.671	9.895	13.704	1.224
ETS	13.713	9.838	13.697	1.214
Auto.ANN	13.961	10.199	13.976	1.241
THETAsm	14.238	9.821	14.217	1.211
SMARTFCS	14.249	10.153	14.296	1.226
Flors.Pearc2	14.335	10.431	14.415	1.255
AutoBox3	14.458	11.192	14.386	1.272
SINGLE	14.637	9.717	14.610	1.229
AutoBox1	14.729	10.961	14.735	1.331
NAIVE2	14.823	9.951	14.814	1.238

Table 4: *Results for the quarterly series, ordered by the first column, which is the average rank of sMAPE.*

We see that the ForcX method obtains the best ranking and is used as the control method. Differences to the BaggedETS method are not significant.

3.5 Results on the quarterly data

Table 4 shows the results for all methods on the quarterly data, ordered by average rank of sMAPE. It can be seen that the two bagged method versions outperform the comparison methods in terms of average rank of sMAPE, and average rank and mean MASE, but not in mean sMAPE. This may indicate that in general the bagged versions perform better, but there are some single series where they yield worse sMAPE results.

Table 5 shows the results of statistical testing considering only the ETS methods. The aligned Friedman test for multiple comparisons results in a p-value of 2.04×10^{-10}, which is highly

Method	p_{Hoch}
BaggedETS.BC	–
BaggedETS	0.777
ETS	0.777
AdditiveETS.BC	0.777

Table 5: *Results of statistical testing for quarterly data, using only exponential smoothing methods. None of the results is statistically significant (so no horizontal line is drawn).*

Method	p_{Hoch}
THETA	–
COMB.S.H.D	0.148
DAMPEN	0.031
PP.Autocast	0.031
ROBUST.Trend	0.031
ForcX	0.018
RBF	0.002
ForecastPro	7.66×10^{-4}
ARARMA	1.19×10^{-4}
Flors.Pearc1	1.19×10^{-4}
B.J.auto	1.07×10^{-4}
AutoBox2	1.07×10^{-4}
BaggedETS	1.60×10^{-5}
HOLT	1.56×10^{-5}
WINTER	6.46×10^{-6}
THETAsm	3.54×10^{-6}
SINGLE	1.63×10^{-7}
Auto.ANN	1.16×10^{-7}
NAIVE2	9.64×10^{-9}
SMARTFCS	8.33×10^{-9}
Flors.Pearc2	3.12×10^{-10}
AutoBox3	5.43×10^{-11}
AutoBox1	1.91×10^{-11}

Table 6: *Results of statistical testing for quarterly data, using BaggedETS.BC and the original results of the M3. A horizontal line separates the methods that perform significantly worse than the best method from the ones that do not. We see that only the COMB.S.H.D is not worse with statistical significance than the THETA method.*

significant. The method with the best ranking is BaggedETS.BC. However, we can see from the table that no differences to the other methods are statistically significant.

Table 6 shows the results for statistical testing of the BaggedETS method against the methods from the original M3 competition (though BaggedETS.BC was chosen in the last test as the control method, we use BaggedETS here, as from Table 4 we can see that it has slightly better performance. Anyway, the methods achieve very similar results). The overall result of the

Friedman aligned rank sum test is a p-value of 2.62×10^{-10}, which is highly significant. We see from the table that the THETA method performs best and is chosen as the control method. It outperforms all methods but COMB.S.H.D with statistical significance.

3.6 Results on the monthly data

Table 7 shows the results for all methods on the monthly data, ordered by average rank of sMAPE. The bagged versions outperform again the other methods. Furthermore, BaggedETS.BC performs particularly well, also outperforming all the methods from the M3. Only ForecastPro achieves a slightly better mean MASE.

Table 8 shows the results of statistical testing considering only the ETS methods. The aligned Friedman test gives a p-value of $p < 10^{-10}$, so that differences are highly significant. The method

	Rank sMAPE	Mean sMAPE	Rank MASE	Mean MASE
BaggedETS.BC	11.178	13.739	11.148	0.852
THETA	11.474	13.892	11.405	0.858
ForecastPro	11.578	13.898	11.597	0.848
BaggedETS	12.047	14.347	11.925	0.854
AdditiveETS.BC	12.424	14.224	12.373	0.891
COMB.S.H.D	12.527	14.466	12.584	0.896
ETS	12.648	14.375	12.718	0.895
HOLT	12.834	15.795	12.794	0.909
ForcX	12.844	14.466	12.894	0.894
WINTER	13.117	15.926	13.100	1.165
RBF	13.302	14.760	13.323	0.910
DAMPEN	13.578	14.576	13.631	0.908
AutoBox2	13.677	15.731	13.738	1.082
B.J.auto	13.746	14.796	13.744	0.914
AutoBox1	13.790	15.811	13.800	0.924
Flors.Pearc2	13.915	15.186	13.947	0.950
SMARTFCS	13.933	15.007	13.833	0.919
Auto.ANN	13.975	15.031	13.992	0.928
ARARMA	14.155	15.826	14.164	0.907
PP.Autocast	14.256	15.328	14.340	0.994
AutoBox3	14.339	16.590	14.242	0.962
Flors.Pearc1	14.627	15.986	14.636	1.008
THETAsm	14.674	15.380	14.687	0.950
ROBUST.Trend	14.874	18.931	14.781	1.039
SINGLE	15.329	15.300	15.412	0.974
NAIVE2	16.157	16.891	16.190	1.037

Table 7: *Results for the monthly series, ordered by the first column, which is the average rank of sMAPE.*

Method	p_{Hoch}
BaggedETS.BC	–
BaggedETS	3.12×10^{-7}
AdditiveETS.BC	8.09×10^{-12}
ETS	1.04×10^{-17}

Table 8: *Results of statistical testing for monthly data, using only exponential smoothing methods. The BaggedETS.BC method clearly outperforms all other methods.*

Method	p_{Hoch}
BaggedETS.BC	–
THETA	0.087
ForecastPro	0.087
COMB.S.H.D	2.27×10^{-7}
ForcX	1.59×10^{-8}
HOLT	1.69×10^{-10}
RBF	1.36×10^{-10}
DAMPEN	9.02×10^{-12}
Auto.ANN	6.64×10^{-14}
WINTER	5.13×10^{-14}
B.J.auto	1.09×10^{-14}
Flors.Pearc2	6.43×10^{-15}
SMARTFCS	4.01×10^{-16}
AutoBox1	1.61×10^{-17}
AutoBox2	5.00×10^{-18}
PP.Autocast	2.12×10^{-18}
AutoBox3	3.46×10^{-24}
ARARMA	2.30×10^{-24}
THETAsm	6.77×10^{-25}
Flors.Pearc1	1.95×10^{-25}
SINGLE	9.72×10^{-35}
ROBUST.Trend	4.50×10^{-49}
NAIVE2	4.17×10^{-62}

Table 9: *Results of statistical testing for monthly data, using BaggedETS.BC and the original results of the M3. BaggedETS.BC performs best, and only the THETA and ForecastPro methods do not perform significantly worse. However, their p-values are close to 0.05 so that they are nearly significant.*

with the best ranking is BaggedETS.BC, and we can see from the table that it outperforms all the other methods with statistical significance.

Table 9 shows the results for statistical testing of the BaggedETS.BC method against the methods from the original M3 competition. The overall result of the Friedman aligned rank sum test is again a p-value of $p < 10^{-10}$, so that it is highly significant. We see from the table that BaggedETS.BC is the best method, and that only the THETA method and ForecastPro are not significantly worse on the chosen 5% significance level. However, their p-values are close to this level. Using MASE for the significance tests (results are not reported here), all methods perform significantly worse than BaggedETS.BC.

4 Conclusions

In this work, we have presented a novel method of bagging for exponential smoothing methods, using Box-Cox transformation, STL decomposition, and the moving block bootstrap. The method is able to consistently outperform the basic exponential smoothing methods. On the monthly data of the M3 competition, the bagged exponential smoothing method with Box-Cox transformation is able to outperform, with statistical significance, all methods that took part in the competition. So, especially for monthly data this method can be recommended to be used in practice.

Acknowledgements

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)", and he was visiting the Department of Econometrics & Business Statistics, Monash University, Melbourne, Australia. Furthermore, we would like to thank X. Shao for providing code of his bootstrapping procedures.

References

Box GEP, Cox DR (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 RB, Cleveland WS, McRae J, Terpenning I (1990) STL: A seasonal-trend decomposition procedure based on loess. Journal of Official Statistics 6:3–73

Cleveland WS, Grosse E, Shyu WM (1992) Local regression models. Statistical Models in S., Chapman & Hall/CRC, chap 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, 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, 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

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

Hyndman RJ, Koehler AB, Ord JK, Snyder RD (2008) Forecasting with Exponential Smoothing: The State Space Approach. Springer Series in Statistics, Springer, URL http://www.exponentialsmoothing.net

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

Künsch HR (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. Comput J 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. R Foundation for Statistical Computing, Vienna, Austria, 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

Zhang CX, Zhang JS (2009) A novel method for constructing ensemble classifiers. Statistics and Computing 19:317–327