International Journal of Forecasting 36 (2020) 110–115

Contents lists available at ScienceDirect

International Journal of Forecasting

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

A simple combination of univariate models

Fotios Petropoulos [a,*], Ivan Svetunkov [b]

[a] School of Management, University of Bath, UK
[b] Centre for Marketing Analytics and Forecasting, Lancaster University Management School, Lancaster, UK

ARTICLE INFO

Keywords:
M4-competition
ETS
ARIMA
Theta method
Complex exponential smoothing
Median combination

ABSTRACT

This paper describes the approach that we implemented for producing the point forecasts and prediction intervals for our M4-competition submission. The proposed simple combination of univariate models (SCUM) is a median combination of the point forecasts and prediction intervals of four models, namely exponential smoothing, complex exponential smoothing, automatic autoregressive integrated moving average and dynamic optimised theta. Our submission performed very well in the M4-competition, being ranked 6th for the point forecasts (with a small difference compared to the 2nd submission) and prediction intervals and 2nd and 3rd for the point forecasts of the weekly and quarterly data respectively.

© 2019 International Institute of Forecasters. Published by Elsevier B.V. All rights reserved.

1. The approach

Our approach is a simple combination of four models. All models were applied using existing implementations in the R statistical software. The following four models were considered (Table 1 summarises the R package and function used for each model and frequency).

- Exponential smoothing (ETS, Hyndman, Koehler, Snyder, & Grose, 2002), which selects the best model underlying one of the fifteen exponential smoothing methods based on the minimisation of a prespecified information criterion. For time series with frequencies lower than or equal to 24 (yearly, quarterly, monthly and daily), we used the `ets()` function of the *forecast* package (Hyndman, Athanasopoulos, Bergmeir, Caceres, Chhay, O'Hara-Wild, et al., 2017); for higher frequencies (weekly and hourly series), we used the `es()` function of the *smooth* package (Svetunkov, 2018), which selects the best model out of the possible 30. The state space model underlying the `es()` function differs from that underlying `ets()` mainly in the usage of the log normal distribution for multiplicative error models and a different transition matrix for the seasonal cases. In addition, `es()` is able to work on the data with frequencies greater than 24. By default, both functions use the corrected Akaike information criterion (AICc) for model selection. However, `es()` uses the "branch and bound" algorithm, which reduces the pool of models under considerations from 30 to a maximum of 10. This speeds up the selection process without significantly reducing the accuracy of the selected model.
- Complex exponential smoothing (CES, Svetunkov & Kourentzes, 2018), which sidesteps the ETS taxonomy and produces non-linear trends with a slope that depends on the data characteristics. There are both non-seasonal and seasonal versions of this model. The former allows one to slide between the level and the trend without the need for a dichotomic selection of components that is appropriate for the time series. The latter captures the type of seasonality (additive or multiplicative) and produces the appropriate forecasts, once again without the need to switch between the two option. The combination of these two models allows us to capture complex dynamics in the data, sidestepping the ETS

* Correspondence to: Fotios Petropoulos, School of Management, University of Bath, Claverton Down, Bath, BA2 7AY, UK.
E-mail addresses: f.petropoulos@bath.ac.uk (F. Petropoulos), i.svetunkov@lancaster.ac.uk (I. Svetunkov).

https://doi.org/10.1016/j.ijforecast.2019.01.006
0169-2070/© 2019 International Institute of Forecasters. Published by Elsevier B.V. All rights reserved.

taxonomy. Thus, CES is a data-driven model that is supposed to do a good job of capturing the wide variety of tendencies. We used the `auto.ces()` function of the *smooth* package, which makes a selection from among two seasonal models and one non-seasonal model using AICc (see Svetunkov & Kourentzes, 2018, for details).

- An automatic autoregressive integrated moving average model (ARIMA, Hyndman & Khandakar, 2008), which identifies the best ARIMA model. The automated selection works as follows. First, the appropriate degree of differencing is determined by the Kwiatkowski-Phillips–Schmidt–Shin unit root test. Then, four simple models ($p = q = 0$, $p = q = 2$ or $p + q = 1$, where p and q refer to the autoregressive and moving average orders of the models respectively) are fitted, and the model with the lowest AICc is selected as the temporary best model. The search for a better model involves varying the values of p and q of the temporary best model by ± 1; if a better model (a model with a lower AICc) is indeed found, then this model becomes the temporary best model. Models both with and without the constant are considered when searching for the best model. The search continues until no better model can be found. The maximum orders of p and q that are tested are five. A similar approach is used for the seasonal ARIMA models, but restricting the maximum orders of the seasonal AR and MA to two. A detailed explanation of the order selection mechanism is given by Hyndman and Khandakar (2008). This mechanism is implemented in the `auto.arima()` function of the *forecast* package.
- The dynamic optimised theta model (DOTM, Fiorucci, Pellegrini, Louzada, Petropoulos & Koehler, 2016), which is an extension of the theta method for forecasting (Assimakopoulos & Nikolopoulos, 2000) that achieved the best performance in the M3 competition (Makridakis & Hibon, 2000). The data are first checked for seasonality using an autocorrelation function test of the lag that matches the frequency of the data; we opted for a 90% confidence level, similarly to the original implementation of Assimakopoulos and Nikolopoulos (2000). If the data are found to be seasonal, then the seasonality is removed as follows. The seasonal indices are calculated by applying the multiplicative classical decomposition method. Subsequently, the original data are divided by the respective seasonal indices to produce the seasonally adjusted data. The original theta method decomposed the seasonally-adjusted data into two "theta" lines, where the first line was the linear regression line on time and had no curvature ($\theta = 0$), while the second had double the curvature of the seasonally adjusted data ($\theta = 2$). According to Assimakopoulos and Nikolopoulos (2000), these two theta lines are able to capture the long- and short-term features of the data. For each series, DOTM optimises the θ value of the theta line that focuses on the short-term curvatures of the seasonally-adjusted data. We used the `dotm()` function of the *forecTheta* package (Fiorucci, Louzada & Yiqi, 2016). Note that for series which are longer than 5000 observations (D2047, D2194 and D4099), DOTM is applied only on the most recent 5000 observations.

Table 1
Forecasting models and the corresponding R functions.

Model	Frequency	R package	Function
ETS	≤ 24	*forecast* 8.2	`ets()`
	> 24	*smooth* 2.3.1	`es()`
CES	All	*smooth* 2.3.1	`auto.ces()`
ARIMA	All	*forecast* 8.2	`auto.arima()`
DOTM	All	*forecTheta* 2.2	`dotm()`

Table 2
Horizons required and frequencies assumed for each data set.

Data set	Horizon	Frequency
Yearly	6	1
Quarterly	8	4
Monthly	18	12
Weekly	13	52
Daily	14	7
Hourly	48	168

Table 2 shows the frequencies that we assumed for each data set, together with the forecasting horizons that were required by the organisers of the M4-competition. Even if some data, such as daily or hourly, could exhibit multiple seasonalities, we have opted for simplicity and modelled each series using only a single frequency that is suitable for the relatively short required forecasting horizons.

We use the median operator to combine the outputs of the four models. While more complicated methods of averaging the prediction interval information from the different models could have been considered, Lichtendahl, Grushka-Cockayne, and Winkler (2013) claim that a simpler method of averaging the quantiles seems to work quite well. As such, we opt for simplicity and take the median of the upper bounds and the median of the lower bounds for all models in order to obtain the combined intervals. After averaging the point forecasts and prediction intervals using the median operator, we set any negative values to zero, as the M4-competition data set consists of non-negative values.

A flowchart of the proposed simple combination of univariate models (SCUM) approach for producing the point forecasts is depicted in Fig. 1, where y denotes the in-sample time series data, *freq* refers to the frequency of this series (for example, 12 for monthly data), *obs* is the number of the observations in the in-sample data, $\hat{y}_m$ is the point forecast for method m, and $\hat{y}$ is the combined point forecast. The upper and lower prediction intervals are produced similarly.

2. Reproduction information

We have provided two R scripts that allow users to reproduce the point forecasts and prediction intervals submitted to the M4-competition. These can be found in

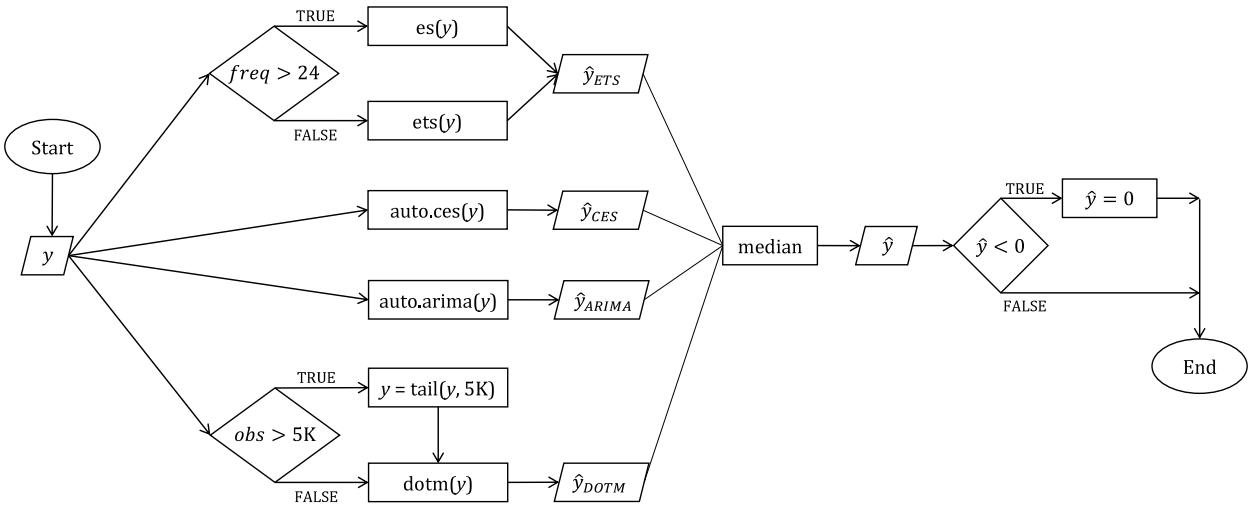

Fig. 1. The flowchart of the SCUM approach.

the GitHub folder of the M4-competition.[1] We used R version 3.4.3 (2017-11-30), *forecast* version 8.2, *smooth* version 2.3.1, *forecTheta* version 2.2. Moreover, a parallel implementation was adopted in our code, where the following packages were used: *doSNOW* (1.0.16), *foreach* (1.4.4), *snow* (0.4–2), *iterators* (1.0.9).

All forecasts and prediction intervals were produced using the Bath Balena High Performance Computing (HPC) Service at the University of Bath. More specifically, we used Intel Skylake nodes that are dual socket nodes clocked at 2.6 GHz. Each Skylake node has 2× Intel Xeon Gold 6126 CPUs providing 24 cores and 192GB memory (8GB per core) via 6 memory channels per CPU socket.

3. Computational time

Our scientific research focuses on forecasting in retail, which motivated a lot of the steps in the development of the SCUM algorithm. Modern retail settings consist of hundreds of thousands of combinations of stock keeping units and locations. Moreover, as multiple replenishment cycles occur each day, the number of time series forecasts that need to be produced each day by large multinational retailers approaches the seven-digit threshold. This exponential increase means that the time that a method requires to produce forecasts becomes of major importance. In fact, it is argued that any added benefits in accuracy when comparing approach *A* with approach *B* (the benchmark) should be balanced against the additional computational time that is required to compute the forecasts for *A* compared to *B* (Nikolopoulos & Petropoulos, 2018). In essence, the excess computational time can be converted to a monetary cost, which, ideally, can be compared against the cost of the error that is saved by using *A* over *B*.

Table 3 shows the average computational time (in seconds) that is needed to produce forecasts for seasonal monthly simulated series of varying lengths (5, 10, 15 and 20 years of in-sample data; 100 series were simulated at each length) using the SCUM approach. We also present the time that automatic ETS and ARIMA approaches require for producing forecasts on the same data (using the *forecast* package). The computations were performed on a machine that features Intel Core i7 7500U with 16GB memory and Windows 10 Pro (no parallelisation was applied). We observe that the good performance of SCUM involves only a moderate increase in computational time relative to approaches that are considered widely as benchmarks in forecasting research (Petropoulos, Wang, & Disney, 2019). In contrast, approaches that involve machine learning elements or the selection of models using cross-validation techniques (rolling origin evaluation) will result in a multifold increase in computational time when the models require retraining. An in-depth computational analysis of the SCUM approach compared to other submissions in the M4-competition is provided by Makridakis, Spiliotis, and Assimakopoulos (2019) in this special issue.

Table 3
Average computational time for producing forecasts (in seconds).

Approach	Observations (in months)			
	60	120	180	240
ETS	0.542	0.632	0.732	0.812
ARIMA	0.580	2.819	1.278	1.911
SCUM	1.884	4.489	3.313	4.285

4. Why did this approach perform well and how can it be improved?

In our opinion, the main reasons for the good performance of SCUM are as follows.

- Combination of the models.
  The previous competitions have showed, as indeed did this one, that combining forecasting methods

[1] https://github.com/M4Competition/M4-methods.

Table 4
Percentage differences between the performance of the combined forecast using the median operator (SCUM) and those of the four individual models and a combination based on the arithmetic mean.

	Data set	ETS	CES	ARIMA	DOTM	Mean combination
sMAPE	Yearly	12.3%	10.8%	11.0%	0.1%	0.9%
	Quarterly	5.1%	8.0%	6.5%	3.0%	0.7%
	Monthly	5.2%	6.7%	4.6%	3.6%	−0.3%
	Weekly	5.5%	21.4%	0.7%	14.6%	1.5%
	Daily	4.5%	2.1%	7.1%	2.2%	0.8%
	Hourly	10.5%	124.2%	4.3%	6.2%	16.0%
	Overall	7.1%	8.5%	6.7%	2.6%	0.3%
MASE	Yearly	11.4%	13.3%	10.2%	−0.2%	−0.1%
	Quarterly	3.8%	7.2%	4.2%	6.1%	0.2%
	Monthly	3.8%	5.7%	2.0%	7.2%	−0.3%
	Weekly	7.6%	13.9%	5.8%	15.5%	1.2%
	Daily	8.4%	0.4%	5.8%	1.3%	1.8%
	Hourly	15.4%	72.8%	−3.9%	22.8%	5.7%
	Overall	7.7%	9.6%	6.5%	3.3%	0.0%

Table 5
Relative frequency with which each model was used to the calculation of the final point forecasts.

Data set	ETS	CES	ARIMA	DOTM
Yearly	59.1%	27.5%	58.0%	55.5%
Quarterly	56.3%	37.7%	52.5%	53.7%
Monthly	57.1%	41.9%	44.8%	56.3%
Weekly	61.5%	49.9%	42.2%	47.0%
Daily	63.2%	38.8%	50.0%	48.2%
Hourly	64.7%	35.2%	46.2%	54.1%

produces forecasts that are more accurate than those of the individual methods. This is due to the increased robustness of the final forecasts and the decrease in the risk of having a completely incorrect forecast. While individual models might fail from time to time, their combination tends to be closer to the true value. Table 4 presents the percentage difference between the performance of our approach (SCUM) and that of each of the four models for each frequency and error measure (symmetric mean absolute percentage error, sMAPE, and mean absolute scaled error, MASE) that was used in the M4-competition. A positive value suggests that SCUM performs better than the respective approach. For example, the sMAPE of the ETS for the yearly data is 12.3% higher than the respective value of SCUM, and is calculated as $100 \times \frac{sMAPE_{ETS} - sMAPE_{SCUM}}{sMAPE_{SCUM}}\%$, where $sMAPE_{ETS}$ and $sMAPE_{SCUM}$ are the sMAPE values for the yearly data for ETS and SCUM respectively. In some cases (notably for yearly data and ETS, CES and ARIMA), the percentage differences are higher than 10%. It is worth mentioning that CES does not perform well for hourly data, with the respective percentage differences of SCUM from this approach being 124.2% and 72.8% for the sMAPE and the MASE respectively. Only in two cases do we observe a deterioration in performance compared to a single model, namely for DOTM applied to yearly data and ARIMA applied to hourly data when the performance is measured using the MASE.

- Pool of forecasting models.
  The forecasting models used in our approach are diverse. ETS is based on time series decomposition, ARIMA captures inter-dependencies in time series, CES focuses on non-linear long-term tendencies in time series, and DOTM employs short-term curvatures and long-term trends. Thus, each model captures something that the others cannot. As can be seen from Table 4, the ranks of the four models are different for different frequencies of data. At the same time, the combination of such different forecasting models leads to an increase in accuracy relative to the individual performances of each of them.
- Median combination.
  Instead of using the mean, we use the median. In essence, the two models that produce the most extreme forecasts are discarded and the final forecast is in between those of the two remaining models. Although it might seem that this should not make a difference for such a small pool of models, it allows us to decrease the influence of models that failed in some time series. For example, if ARIMA produced a forecast with a downward trend while the other models produced forecasts with upward ones, the median makes sure that the final forecast has the latter. The last column of Table 4 shows that the choice of the median operator (as opposed to the mean) did indeed result in small improvements for the majority of the data frequencies. Table 5 shows the frequency with which each of the four models contributed to the calculation of the final point forecasts. Note that different models might be used for different forecasting horizons within the same time series. For example, the required forecast horizon for the yearly frequency was six periods ahead and the number of yearly series was 23,000. This means that we produced 138,000 point forecast combinations for the yearly frequency. Of these 138,000 combinations, 81,616 (59.1%) used ETS, 37,891 (27.5%) used CES, 80,077 (58%) used ARIMA and 76,527 (55.5%) used DOTM. In a small number of cases, two or more models produced the same forecasts, meaning that effectively more than two models contributed to the final combined forecast. As a result, the sums of some rows in Table 5 are slightly higher than 200% (276,111 or 200.08% for the yearly frequency). Overall, ETS models are used most frequently (56% to almost 65% for the various subsets), followed by DOTM; while CES is the model that is used least often, with a frequency that varies from 27.5% to almost 50% of cases.

The same principles were used for the interval forecasts, making them robust in comparison to the prediction intervals produced by the individual models.

Practices that could have been applied to further enhance the accuracy of SCUM include the following.

- Optimal selection of the in-sample window of data.
  SCUM uses the entire in-sample history of data (in the case of DOTM, the most recent 5000 observations are used). However, one could argue that such a long history might not be relevant when the forecast horizon is relatively short. For example, when the required forecast horizon is just 18 months ahead, the longest monthly series consists of 2,794 observations. This argument becomes particularly clear in the case of structural changes in the patterns of the data and/or when outlying values have been recorded. In such cases, fitting the models only on a subset of the most recent data that does not include such irregularities might yield more robust forecasts. However, we must emphasise that this subset should still be long enough to allow for an appropriate estimation of the models' parameters and of the uncertainty around the forecasts. One strategy for defining an optimal in-sample window for each series could include the application of techniques for identifying structural changes in the historical patterns of the data; in the case of structural changes, only the latest data after such changes should be considered when fitting the models. Alternatively, an empirical exercise could shed light on the optimal in-sample window in an aggregate manner (per frequency or category), rather than for each individual time series.
- Parameter optimisation that corresponds to the cost functions explicitly.
  We used the default settings for each model, which implies the minimisation of the mean squared error (MSE) of the one-step-ahead forecast. This can be considered as a limitation, because the estimator is not aligned with the error measures used by the organisers of the competition (an average of sMAPE and MASE). If each of the models was estimated via the minimisation of OWA, then there could be potential accuracy gains. The same also applies to the prediction intervals, which were generated directly from our models: we could have improved their accuracy by generating them from the models with the specific estimators driven by the error measure for intervals (mean scaled interval score).
- Multiple temporal aggregation (MTA).
  Many recent studies have shown the positive impact of multiple temporal aggregation on the forecasting accuracy (for example, see: Athanasopoulos, Hyndman, Kourentzes, & Petropoulos, 2017; Kourentzes, Petropoulos, & Trapero, 2014; Petropoulos & Kourentzes, 2014). Such improvements are associated primarily with reducing the model uncertainty by expressing the input data in alternative time frequencies. We would expect that an integrated SCUM/MTA approach would increase the forecast accuracy further, admittedly with an associated increase in the computational cost.
- Multiple seasonal cycles.
  When forecasting series with high frequencies, the results could possibly have been improved by applying models that explicitly consider multiple seasonal cycles. The multiple seasonal decomposition function (`mslt()`) of the *forecast* package could be used for this purpose. This would be especially important if the desired horizons were greater than those requested by the organisers of the M4-competition.

5. Final comment

As can be seen from this note, SCUM is a very simple approach that relies on established forecasting models and uses one of the main forecasting principles: combination. The application of SCUM is very straightforward, as the four components of this approach are available readily in the R statistical software. The preliminary results of the M4-competition (Makridakis, Spiliotis, & Assimakopoulos, 2018) show that this simple approach leads to accurate forecasts, placing SCUM at 6th place on the ladder. A comparison of the SCUM approach with the top five methods in the M4-competition suggests that the SCUM is probably the simplest forecasting approach among the leaders of the competition, providing a balance between computational complexity and forecasting accuracy.

Acknowledgments

This project made use of the Balena High Performance Computing (HPC) Service at the University of Bath.

References

Assimakopoulos, V., & Nikolopoulos, K. (2000). The Theta model: a decomposition approach to forecasting. *International Journal of Forecasting*, *16*(4), 521–530.

Athanasopoulos, G., Hyndman, R. J., Kourentzes, N., & Petropoulos, F. (2017). Forecasting with temporal hierarchies. *European Journal of Operational Research*, *262*(1), 60–74.

Fiorucci, J. A., Louzada, F., & Yiqi, B. (2016). forecTheta: Forecasting time series by theta models. *R package version 2.2*. https://cran.r-project.org/package=forecTheta.

Fiorucci, J. A., Pellegrini, T. R., Louzada, F., Petropoulos, F., & Koehler, A. B. (2016). Models for optimising the theta method and their relationship to state space models. *International Journal of Forecasting*, *32*(4), 1151–1161.

Hyndman, R., Athanasopoulos, G., Bergmeir, C., Caceres, G., Chhay, L., O'Hara-Wild, M., et al. (2017). forecast: Forecasting functions for time series and linear models. *R package version 8.2*. https://cran.r-project.org/package=forecast.

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

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

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

Lichtendahl, K. C., Grushka-Cockayne, Y., & Winkler, R. L. (2013). Is it better to average probabilities or quantiles? *Management Science*, *59*(7), 1594–1611.

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

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2018). The M4 competition: results, findings, conclusion and way forward. *International Journal of Forecasting*, *34*, 802–808.

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2019). The M4 competition: 100,000 time series, 61 forecasting methods. *International Journal of Forecasting*.

Nikolopoulos, K., & Petropoulos, F. (2018). Forecasting for big data: Does suboptimality matter? *Computers & Operations Research*, *98*, 322–329.

Petropoulos, F., & Kourentzes, N. (2014). Improving forecasting via multiple temporal aggregation. *Foresight: Int. J. Appl. Forecast.*, *34*, 12–17.

Petropoulos, F., Wang, X., & Disney, S. M. (2019). The inventory performance of forecasting methods: Evidence from the M3 competition data. *International Journal of Forecasting*, *35*, 251–265.

Svetunkov, I. (2018). smooth: forecasting using smoothing functions. *R package version 2.3.1*. https://github.com/config-i1/smooth.

Svetunkov, I., & Kourentzes, N. (2018). *Complex exponential smoothing for seasonal time series*: *Vol. 1*, (pp. 1–20). Working Paper of Department of Management Science, Lancaster University.