The Benchmarks

Like done in the M4 competition, there will be fourteen (14) benchmark methods, ten (10) statistical and four (4) Machine Learning (ML) ones. As these methods are well known, readily available, and straightforward to apply, the accuracy of the new ones proposed in the M4 Competition must provide superior accuracy in order to be adopted and used in practice (taking also into account the computational time it would be required to utilize a more accurate method versus the benchmarks whose computational requirements are minimal).

Statistical Benchmarks

1. Naive: A random walk model, defined as

$$\hat{Y}_{n+i} = Y_n \,,\; i = 1,\, 2, \ldots h \,.$$

2. Seasonal Naive (sNaive): Like Naive, but this time the forecasts of the model are equal to the last known observation of the same period in order for it to capture possible weekly seasonal variations.

3. Simple Exponential Smoothing[1] (SES): The simplest exponential smoothing model, aimed at predicting series without a trend, defined as

$$\hat{Y}_t = aY_t + (1-a)\,\hat{Y}_{t-1}.$$

The smoothing parameter *a* is selected from the range [0.1, 0.3] by minimizing the insample mean squared error (MSE) of the model, while the first observation of the series is used for initialization.

4. Moving Averages (MA): Forecasts are computed by averaging the last *k* observations of the series, as follows

$$\hat{Y}_t = \frac{\sum_{i=1}^{k} Y_{t-i}}{k},$$

where *k* is selected from the range [2, 5] by minimizing the insample MSE.

5. Croston's method[2] (CRO): The method proposed by Croston to forecast series that display intermittent demand. Th ε method decomposes the original series into the non-zero demand size z_t and the inter-demand intervals p_t, deriving forecasts as follows:

$$\hat{Y}_t = \frac{\hat{z}_t}{\hat{p}_t},$$

where both z_t and p_t are predicted using SES. The smoothing parameter of both components is set equal to 0.1. The first observation of the components are used for initialization.

[1] Gardner Jr., E. S. (1985). Exponential smoothing: The state of the art. Journal of Forecasting, 4, 1–28.
[2] Croston, J. D. (1972). Forecasting and stock control for intermittent demands. Journal of the Operational Research Society, 23, 289–303.

6. Optimized Croston's method (optCro): Like CRO, but this time the smoothing parameter is selected from the range [0.1, 0.3], like done with SES, in order to allow for more flexibility. The non-zero demand size and the inter-demand intervals are smoothed separately using (potentially) different *a* parameters.

7. Syntetos-Boylan Approximation[3] (SBA): A variant of the Croston's method that utilizes a debiasing factor as follows:

$$\hat{Y}_t = 0.95\frac{\hat{z}_t}{\hat{p}_t},$$

8. Teunter-Syntetos-Babai method[4] (TSB): A modification to Croston's method that replaces the inter-demand intervals component with the demand probability, d_t, being 1 if demand occurs at time *t* and 0 otherwise. Similarly to Croston's method, d_t is forecasted using SES. The smoothing parameters of d_t and z_t may differ, exactly as optCRO. The forecast is given as follows:

$$\hat{Y}_t = \hat{d}_t\hat{z}_t,$$

9. Aggregate-Disaggregate Intermittent Demand Approach[5] (ADIDA): Temporal aggregation is used for reducing the presence of zero observations, thus mitigating the undesirable effect of the variance observed in the intervals. ADIDA uses equally sized time buckets to perform non-overlapping temporal aggregation and predict the demand over a pre-specified lead time. The time bucket is set equal to the mean inter-demand interval. SES is used to obtain the forecasts.

10. Intermittent Multiple Aggregation Prediction Algorithm[6] (iMAPA): Another way for implementing temporal aggregation in demand forecasting. However, in contrast to ADIDA which considers a single aggregation level, iMAPA considers multiple ones, aiming at capturing different dynamics of the data. Thus, iMAPA proceeds by averaging the derived point forecasts at each temporal level, generated using SES. The maximum aggregation level is set equal to the maximum inter-demand interval.

Machine Learning Benchmarks

11. Multi-Layer Perceptron (MLP): A single hidden layer NN of 14 input nodes (last two weeks of available data), 28 hidden nodes, and one output node. The Scaled Conjugate Gradient method is used for estimating the weights which are initialized randomly, while the maximum iterations are set equal to 500. The activation functions of the hidden and output layers are the logistic and linear one, respectively. In total, 10 MLPs are trained to forecast each series and then the median operator is used to average the individual forecasts in order to mitigate possible variations due to poor weight initializations.

---

[3] Syntetos, A. A., & Boylan, J. E. (2005). The accuracy of intermittent demand estimates. International Journal of Forecasting, 21, 303–314.

[4] Teunter, R. H., Syntetos, A. A., & Babai, M. Z. (2011). Intermittent demand: Linking forecasting to inventory obsolescence. European Journal of Operational Research, 214, 606–615.

[5] Nikolopoulos, K., Syntetos, A. A., Boylan, J. E., Petropoulos, F., & Assimakopoulos, V. (2011). An aggregate-disaggregate intermittent demand approach (ADIDA) to forecasting: an empirical proposition and analysis. Journal of the Operational Research Society, 62, 544–554.

[6] Petropoulos, F., & Kourentzes, N. (2015). Forecast combinations for intermittent demand. Journal of the Operational Research Society, 66, 914–924

12. Random Forest (RF): This is a combination of multiple regression trees, each one depending on the values of a random vector sampled independently and with the same distribution. Given that RF averages the predictions of multiple trees, it is more robust to noise and less likely to over-fit on the training data. We consider a total of 500 non-pruned trees and 4 randomly sampled variables at each split. Bootstrap sampling is done with replacement. Like done in MLP, the last 14 observations of the series are considered for training the model.

13. Global Multi-Layer Perceptron (GMLP): Like MLP, but this time, instead of training multiple models, one for each series, a single model that learns across all series is constructed. This is done given that M4 indicated the beneficial effect of cross-learning. The last 14 observations of each series are used as inputs, along with information about the coefficient of variation of non-zero demands (CV^2) and the average number of time periods between two successive non-zero demands (ADI). This additional information is used in order to facilitate learning across series of different characteristics.

14. Global Random Forest (GRF): Like GMLP, but instead of using an MLP for obtaining the forecasts, a RF is exploited instead.

The code for generating the forecasts of the abovementioned benchmarks will be available on the **GitHub** repository of the competition.

Note that the benchmark methods are applied at the product-store level of the hierarchically structured dataset. The bottom-up method is then used for obtaining reconciled forecasts at the rest of the hierarchical levels.

Also note that benchmarks are not eligible for a prize, meaning that the total amount will be distributed among the competitors even if the benchmarks perform better than the forecasts submitted by the participants. Similarly, any participating method associated with the organizers and the data provider, will not be eligible for a price.