Hierarchical Load Forecasting: 2017 Global Energy Competition - Project milestone

Inés Dormoy, Louis Gautier
ICME & MS&E, Stanford University
idormoy@stanford.edu, lgautier@stanford.edu
The code for our project is available on this repository

1 Introduction

Load forecasting, which consists of predicting the demand for energy consumption, is a crucial task in the Energy industry. Historical load data often integrates a hierarchical aspect. Indeed, utilities commonly have access to individual consumption data at the meter level and want to forecast aggregate energy consumption at various levels of groupings of individual meters (which could be neighborhoods, regions, etc.). Therefore, in this context, hierarchy refers to high-level grouped time series being the sum of bottom base time series. The 2017 GEFCom competition [1] by Hong et. al was precisely a hierarchical load forecasting competition. It consisted of providing probabilistic load forecasts at several levels of aggregation based on a dataset containing historical load data at the level of individual meters, hierarchy information, as well as weather data. Successful teams used models as varied as Quantile Regression [2], Generalized Additive Models [3], Gradient Boosting [4], Quantile Random Forest [5] and SARIMA [6]. Few teams took into account the hierarchical aspect of the data. According to the organizers of the competition, this is probably because hierarchical reconciliation methods weren't widely used in 2017 [1]. Since 2017, hierarchical methods have been further developed, including bottom-up and top-down approaches[7], trace minimization [8], and deep learning approaches [9]. In this project, our goal is to compare baseline methods not leveraging the hierarchical information with hierarchical methods in terms of performance, efficiency, and interpretability.

We first developed base models that allow us to forecast load at each level of the hierarchy. On the one hand, we trained classical Ensemble models with basic feature engineering, which don't directly model the time dependency and auto-correlation patterns in the data. On the other hand, we fitted widely used time series models, like SARIMA models, and Prophet, an additive model that includes components for trend, seasonal, and cyclical effects. Then, we compared keeping the individual base models without reconciliation with using hierarchical reconciliation methods, from top-down and bottom-up approaches to two optimal reconciliation approaches [10][8].

2 GEFCom 2017 Competition

2.1 Dataset and Task Description

This dataset contains seven years (from 2005 to 2011) of hourly load data measured at the level of 183 individual meters. Based on the historical data from 2005 to 2010, the task is to forecast the load during the last year of data (2011), at different levels of hierarchy, from the individual meters themselves to intermediary stations aggregating the consumption of several meters. The hierarchical structure is presented in Appendix A. This dataset also comes with temperature and relative humidity data from 28 weather stations for which the location is not provided. One of the greatest challenges for working with this dataset is its size. It gives access to a total of 10 million load readings and 3.4 million weather data points.

2.2 Feature Engineering

We performed basic feature engineering to be able to better capture seasonality effects. We added cosine and sine transformations of the month and the hour, as we assume that close months and close times of the day will have the same load patterns, and we added an indicator variable of whether the day is a holiday in Massachusetts, the state the data was collected in.

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

3 Methods

3.1 Base learners

Historical Averages We first implemented a very simple baseline that consists of estimating the load at each level of the hierarchy by averaging historical loads with the same (month, day of the week, time of the day) tuple for each one of the considered grouping of meters. This baseline was only fitted and tested on the top time series.

LightGBM We then used an ensemble model called *LightGBM*, which is scalable enough to be trained on our large dataset with reasonable memory requirements, and allows us to model the complex non-linear dependency between load at each level, seasonality effects and weather. We concatenated the measured temperature and humidity values with the historical load data and used the feature engineering described in 2.2 to capture seasonality effects. We performed hyperparameter selection using k-fold cross-validation and a search procedure with a fixed time budget. We trained one model for each grouping level (to predict the aggregate load on each mid-level and aggregate level and on the total consumption). In order to limit the number of models to train to predict the bottom level (individual meters), we made the assumption that meters that belong to the same mid-level grouping have similar consumption patterns. We thus used one model for each mid-level aggregation level to predict the bottom time series, and each one of these models had the one-hot encoded version of the meter ID of the observed measurements as features.

SARIMA Before fitting a SARIMA model on our top time series, we scale the data to avoid convergence issues when fitting. We first apply a first-order differentiating to remove the trend, which is sufficient to observe a zero mean in the process. The PACF and ACF plots indicate a 24-hour seasonality, as well as decays for seasonal as well as non-seasonal PACF and ACF values. Therefore, we fit a SARIMA(p=1,d=1,q=1)x(P=1,D=24,Q=1) model on the scaled top time series. We additionally used AIC and BIC to test other values of p, q, P, and Q.

Prophet No preprocessing is applied before fitting the Prophet model. We apply cross-validation to compute the coverage of the model. This baseline was only fitted and tested on the top time series.

3.2 Hierarchical reconciliation

Once we have developed the base learners that allow us to predict the load at each level of the hierarchy, the predictions most likely won't be consistent with each other: the sum of the loads at the lower levels don't exactly add up to the higher levels of hierarchy. To obtain more consistent, explainable, and hopefully more accurate forecasts, we need to apply a reconciliation method to enforce the sum constraints encoded by the hierarchy. We used three of these methods.

Bottom-up The most simple one, the Bottom-up approach, consists of using only predictions at the bottom (meter) level and summing the results up to obtain forecasts for each grouping at higher hierarchy levels.

Top-down The top-down approach consists of fitting and predicting only the total time series (sum of all the meter-level time series), and distributing it among lower levels of hierarchy using the contributions $p_{j,l}$ with $y_{j,l} = p_j * y_{l+1}$ for the j-th sub-element of the grouping having time series (y_{l+1}). Several approaches are possible, but based on our experiments, the most effective is to use the proportion of historical averages:

$$p_{l,j} = \sum_{t=1}^{T} \frac{y_{j,l,t}}{T} / \sum_{t=1}^{T} \frac{y_{l+1,t}}{T} \tag{1}$$

Optimal reconciliation with MinTrace More generally, as outlined in [10], any reconciliation method can be written as finding the optimal mapping matrix G such that for all $h \in [|1, T|]$, the reconciled forecast $\tilde{y}_h$ verifies $\tilde{y}_h = SG\hat{y}_h$, where S is the aggregation matrix encoding the (known) hierarchical structure and $\hat{y}_h$ refers to the base forecast for every level of hierarchy. With this structure, the error of our forecast can be written as $V_h = Var(y_{T+h} - \tilde{y}_h) = SGVar(y_{T+h} - \hat{y}_h)G^T S^T$. Finding the optimal reconciliation method consists of minimizing this error over matrices G subject to the constraint that the reconciliation is perfect: $SGS = S$. As proven in [8], given an approximation for $W_h = Var(y_{T+h} - \hat{y}_h)$, this minimization problem has a unique known solution. To approximate W_h, the authors made the simplifying assumption that $W_h = k_h I$ for all h, with k_h a constant per timestep [8]. In this context, G is independent of the data and the solution to the optimization problem is the least squares estimator of $\hat{y}$ against S. In practice, we used HierarchicalForecast, a Python Open Source package that implements bottom-up, top-down and MinTrace approaches.

4 Results

4.1 Experimental setting and metrics

In all our experiments, we excluded all meters that were not active in either the train or the test dataset. We fitted our models on the first six years of data and evaluated them in one pass in the last year (2011). We report the RMSE and MAE as error metrics.

4.2 Experimental results and discussion

Base learners comparison on the sum time series We evaluate the performance of our base learners in Table 1, which outlines the performance of each base learner to predict the top time series (sum of all meter loads) with a time horizon of one year ($T = 8760$).

Table 1: Base learners performance comparison on the sum time series for the whole 2011 year

Model	RMSE	MAE
Historical Averages	119,038	62,794
LightGBM	**45,998**	**33,867**
SARIMA	577,745	500,788
Prophet	258,566	219, 949

As we can observe, the time series models we tried are performing worse on this long-term forecasting task than our traditional Machine Learning baselines. This is most likely due to the fact that with such a large horizon, seasonality and trends are more important than auto-correlation patterns, and our sequence models don't leverage weather information. Our Historical Averages baseline likely suffers from the curse of dimensionality, which explains its poor performance compared to *LightGBM* which is our most accurate base learner on the top time series.

Impact of hierarchical reconciliation Table 2 outlines the impact of different methods to reconcile the forecasts obtained by our *LightGBM* estimator. When looking at Appendix A, the Sum column corresponds to the orange top-level, Top column to the I00 blue level, Middle to E0 level green, and Bottom to the yellow level. The RMSE and MAE are computed as the mean on each level of the hierarchy. The no-reconciliation approach consists of fitting and predicting each one of the bottom, middle, top, and sum time series, while the bottom-up approach is only fitted on the bottom time series.

Table 2: Comparison of different hierarchical reconciliation methods with *LightGBM* as base learner

Level of hierarchy	Sum		Top		Middle		Bottom	
	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
No reconciliation	45,998	33,867	30,728	19,669	**5,753**	**3,080**	**1,679**	**696**
Bottom-up	396,643	309,056	243,194	156,868	41,757	21,863	**1,679**	**696**
Top-down	45,998	33,867	**29,980**	**19,157**	5,905	3,157	4,733	2,251
MinTrace (OLS approx.)	**45,437**	**33,588**	30,038	19,395	6,144	3,663	3,507	2,232

As we can see, using a reconciliation method is beneficial to obtain more accurate forecasts at the sum and top hierarchy levels, but the no-reconciliation method is beneficial at the lower level. This is likely due to the fact that specialized models at lower granularity levels help us improve our accuracy and quantify our uncertainty on higher hierarchy levels.

5 Next steps

Our next objective is to use an LSTM-based sequence model that takes previous loads as well as weather features to model several levels of hierarchy. We then want to use a deep-learning-based approach, inspired by [9], which architecture is directly tailored to hierarchical time series forecasting. Finally, we want to deepen our model's benchmarking and testing to incorporate inference time and explainability, as well as provide insights on different model's tradeoffs, causes of failure, and how they could be used in a real-life setting.

References

[1] Tao Hong, Jingrui Xie, and Jonathan Black. Global energy forecasting competition 2017: Hierarchical probabilistic load forecasting. *International Journal of Forecasting*, 35(4):1389–1399, 2019.

[2] Lingxin Hao and Daniel Q Naiman. *Quantile regression*. Number 149. Sage, 2007.

[3] Trevor J Hastie and Robert J Tibshirani. *Generalized additive models*, volume 43. CRC press, 1990.

[4] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[5] Nicolai Meinshausen and Greg Ridgeway. Quantile regression forests. *Journal of machine learning research*, 7(6), 2006.

[6] M. Braun, T. Bernard, O. Piller, and F. Sedehizade. 24-hours demand forecasting based on sarima and support vector machines. *Procedia Engineering*, 89:926–933, 2014. 16th Water Distribution System Analysis Conference, WDSA2014.

[7] Chul-Yong Lee and Sung-Yoon Huh. Forecasting new and renewable energy supply through a bottom-up approach: The case of south korea. *Renewable and Sustainable Energy Reviews*, 69:207–217, 2017.

[8] George Athanasopoulos Shanika L. Wickramasuriya and Rob J. Hyndman. Optimal forecast reconciliation for hierarchical and grouped time series through trace minimization. *Journal of the American Statistical Association*, 114(526):804–819, 2019.

[9] Syama Sundar Rangapuram, Lucien D Werner, Konstantinos Benidis, Pedro Mercado, Jan Gasthaus, and Tim Januschowski. End-to-end learning of coherent probabilistic forecasts for hierarchical time series. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8832–8843. PMLR, 18–24 Jul 2021.

[10] Rob J Hyndman, Roman A Ahmed, George Athanasopoulos, and Han Lin Shang. Optimal combination forecasts for hierarchical time series. *Computational statistics & data analysis*, 55(9):2579–2589, 2011.

A Hierarchical organization of the data

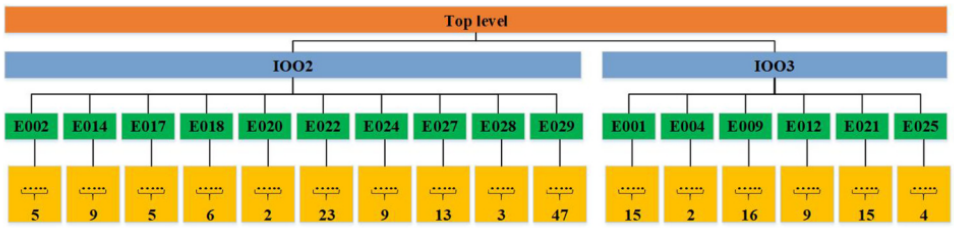

Figure 1: Hierarchical organization of the data, where the numbers at the bottom level indicate the numbers of forecasting locations (total of 183 meters at the base level).