Reconciled boosted models for GEFCom2017 hierarchical probabilistic load forecasting

Cameron Roach

Abstract

When forecasting time series in a hierarchical configuration it is necessary to ensure that forecasts reconcile at all levels. The 2017 Global Energy Forecasting Competition (GEFCom2017) focused on addressing this topic. Quantile forecasts for eight zones and two aggregated zones in New England were required for every hour of a future month. This paper presents a new methodology for forecasting quantiles in a hierarchy which outperforms a commonly used benchmark model. A simulation-based approach was used to generate demand forecasts. Adjustments were made to each of the demand simulations to ensure all zonal forecasts reconciled appropriately. To ensure bottom level zonal forecasts correctly sum to aggregated zonal forecasts a weighted reconciliation approach was implemented. We show that reconciling in this manner improves forecast accuracy. A discussion of results and modelling performance is presented. Brief reviews of hierarchical time series forecasting and gradient boosting are also included.

1 Introduction

Hierarchical time series forecasting occurs in situations where a dependent variable of interest can be disaggregated across nodes of a hierarchy. Examples include forecasting sales of a product within towns and also by state; forecasting economic indicators within states and for an entire country; and in this case, forecasting demand in bottom level and aggregated zones of an electricity network. When forecasting hierarchical time series, the base forecasts typically do not reconcile as one would expect. Forecasts of electricity demand in each of the bottom level zones may not sum up to the forecasts of aggregated zones. Hence, it is often necessary to carry out a reconciliation step to adjust these base forecasts.

In this paper I propose a methodology for hierarchical forecasting of electricity demand across eight zones in New England. In addition to the eight bottom level zones, electricity demand for two aggregated zones are also forecast. This methodology was used in GEFCom2017 in the defined data track. Electricity and weather data were supplied by ISO New England. As this was the defined data track, only electricity demand, dew point temperature, dry bulb temperature and calendar data were allowed as model inputs. We were presented with an ex-ante forecasting problem requiring forecasts of the 10th, 20th, … and 90th quantiles of the demand distribution for every hour of a future month for all zones.

To produce quantile forecasts for demand, weather scenarios are simulated for every zone in the forecast month. A demand model is then used to predict demand for every zone and hour over the forecast horizon. Residuals are also simulated and added which produces simulations for actual demand rather than just the conditional mean. Zonal forecasts are then adjusted to ensure they reconcile appropriately within each simulation. Quantiles are calculated for each hour using the reconciled demand simulations.

The boosted demand model is fit using the XGBoost algorithm (Chen and Guestrin 2016). Regularization with L1 and L2 penalties is applied to avoid over-fitting.

Recent work on hierarchical reconciliation has focused on adjusting base forecasts to obtain reconciled forecasts with improved accuracy (Hyndman, Ahmed, et al. 2011; Wickramasuriya, Athanasopoulos, and Hyndman 2015; Hyndman, Lee, and Wang 2016). These methods only focus on adjusting forecasts of the conditional mean. Despite a shift towards probabilistic forecasting in the energy industry (Hong, Pinson, Fan, et al. 2016) the literature on reconciling probabilistic forecasts in a hierarchical setting remains limited. To the author's knowledge the only relevant paper is Ben Taieb, Taylor, and Hyndman (2017) which proposes a methodology for producing coherent hierarchical probabilistic forecasts of smart meter demand. A contribution of this paper is to enrich the literature on quantile forecasting for hierarchical electricity demand.

The paper has the following structure: Section 2 provides concise reviews of relevant literature on hierarchical energy forecasting and gradient boosting. Competition data and methodology are described in Sections 3 and 4. Section 5 discusses the modelling results and concluding remarks are provided in Section 6.

2 Background theory

2.1 Hierarchical forecasting

When several time series exist in a hierarchy it is necessary to ensure forecasts at each level of the hierarchy reconcile in a sensible manner. When time series exist in a hierarchy they can be expressed in terms of a summing matrix $\mathbf{S}$ (Hyndman, Ahmed, et al. 2011). This summing matrix allows for all nodes to be expressed in terms of the bottom level nodes. For an observation occurring at time t,

$$\mathbf{y}_t = \mathbf{S}\mathbf{y}_{bt}, \tag{1}$$

where $\mathbf{y}_t$ gives the observed values for all aggregated and bottom level nodes and $\mathbf{y}_{bt}$ gives the observed values for only the bottom level nodes.

Hyndman, Ahmed, et al. (2011) showed base forecasts can be reconciled using only the summing matrix $\mathbf{S}$. At forecast horizon h,

$$\tilde{\mathbf{y}}_h = \mathbf{S}(\mathbf{S}'\mathbf{S})^{-1}\mathbf{S}'\hat{\mathbf{y}}_h, \tag{2}$$

where $\tilde{\mathbf{y}}_h$ are the reconciled forecasts and $\hat{\mathbf{y}}_h$ are the base forecasts. This is referred to as ordinary least squares (OLS) reconciliation and was shown to outperform bottom-up and top-down reconciliation approaches for both simulated and real-world data.

Subsequent studies by Wickramasuriya, Athanasopoulos, and Hyndman (2015) and Hyndman, Lee, and Wang (2016) showed that reconciliation could be improved by incorporating a matrix of weights. A generalized least squares (GLS) approach is given by,

$$\tilde{\mathbf{y}}_h = \mathbf{S}(\mathbf{S}'\mathbf{\Sigma}_h^{\dagger}\mathbf{S})^{-1}\mathbf{S}'\mathbf{\Sigma}_{\mathbf{h}}^{\dagger}\hat{\mathbf{y}}_h, \tag{3}$$

where $\mathbf{\Sigma}_h$ is the covariance matrix of the residuals for forecast horizon h and $\mathbf{\Sigma}_h^{\dagger}$ is the Moore-Penrose generalized inverse. It is often difficult to calculate $\mathbf{\Sigma}_h$ and so an alternative weighted least squares (WLS) method can be used instead. If we let $\mathbf{W}$ be a diagonal matrix with elements serving as the weights, then the WLS approach is,

$$\tilde{\mathbf{y}}_h = \mathbf{S}(\mathbf{S}'\mathbf{W}\mathbf{S})^{-1}\mathbf{S}'\mathbf{W}\hat{\mathbf{y}}_h. \tag{4}$$

In Hyndman, Lee, and Wang (2016) it was suggested that the diagonal elements of $\mathbf{W}$ could be equal to the inverse of the h-step forecast variances. When using an ARIMA time series model the h-step ahead forecast variances can be approximated as proportional to the one-step ahead forecast variances. Furthermore, since each fitted value is effectively a one-step ahead forecast, residuals can be used to calculate these variances making (4) a practical means for reconciliation. However, this approach is not necessarily feasible when dealing with other model types that do not produce one-step ahead forecasts when fitted to historical values. Given this, in Section 4.4.1 I propose two alternative weight matrices that can be easily constructed for any model.

2.2 Gradient boosting

Gradient boosting has been used in many machine learning challenges with good results (see for example Ben Taieb and Hyndman (2014) and Koren (2009)). Gradient boosting was first proposed by Schapire (1990). A rigorous statistical overview of boosting is carried out by Friedman, Hastie, and Tibshirani (2000) and Friedman (2001). Chen and Guestrin (2016) proposed the extreme gradient boosting (XGBoost) algorithm which allowed for easy scaling while using less computational resources.

Essentially, boosting works by training an ensemble of weak learners that are then able to provide better predictions than a single model otherwise would. Suppose we are given a data set with n observations and p predictors, $\{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$ where $\mathbf{x}_i \in \mathbb{R}^p$ and $y_i \in \mathbb{R}$. Then predictions are given by,

$$\hat{y}_i = \phi\left(\mathbf{x}_i\right) = \sum_{k=1}^{K} \nu f_k\left(\mathbf{x}_i\right), \quad f_k \in \mathcal{F},$$

where K is the number of weak learners used, ν is a shrinkage parameter to control the learning rate and $\mathcal{F}$ is the model space of the weak learners. Each $f_k(\mathbf{x}_i)$ is fit in a stage-wise manner to the residuals r_i of the previous fit. Residuals are initially set equal to the observed response, $r_i = y_i$ for all i. Then, for each step k a weak learner f_k is fit to the data set $\{(\mathbf{x}_i, r_i)\}_{i=1}^{n}$. Residuals are then updated according to $r_i = r_i - \nu f_k\left(\mathbf{x}_i\right)$.

The weak learner is fit by minimising the objective function,

$$\mathcal{L}\left(\phi\right) = \sum_i l\left(\hat{r}_i, r_i\right) + \sum_k \Omega\left(f_k\right),$$

where l is a loss function and Ω is a penalty function to avoid over-fitting. Terms for L1 and L2 regularization are included within Ω and so the penalty function can effectively carry out lasso, ridge and elastic net type penalisation.

3 Data

Here we present a brief overview of the data and forecasting problem. A detailed discussion of the GEFCom2017 data is provided by Hong, Xie, and Black (2019). Hourly electricity data for eight zones spanning New England was made available by ISO New England. Hourly weather data comprising dry bulb temperatures and dew point temperatures was also provided. This analysis uses data from January 2005 to April 2017. When training a model to forecast for a particular month I used data from January 2005 up to to two months prior to the start of the forecast period. For example, when forecasting April 2017, only data from January 2005 to January 2017 is used

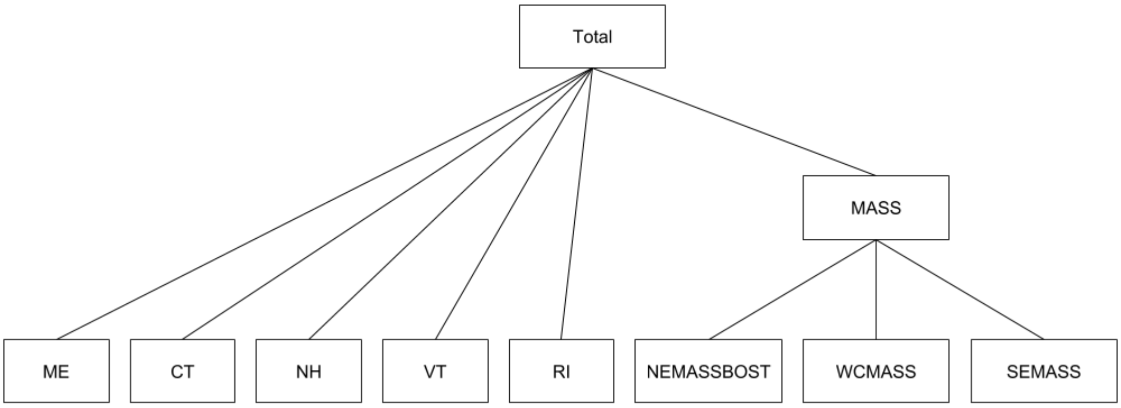

Figure 1: Load forecasting hierarchy for GEFCom2017. There are two aggregated zones and eight bottom level zones.

for training[1]. Public holiday data was also allowed in the competition. Massachusetts (MASS) is composed of three bottom level zones: Southeast Massachusetts (SEMASS), Western/Central Massachusetts (WCMASS) and Northeast Massachusetts (NEMASSBOST). The remaining bottom level zones are Maine (ME), Connecticut (CT), New Hampshire (NH), Rhode Island (RI) and Vermont (VT). The sum of all eight bottom level zones is designated as "TOTAL". Figure 1 shows the structure of the hierarchy.

3.1 Electricity demand

Figure 2 shows the time series data for one top level zone (Total) and one bottom level zone (Vermont). Daylight saving time (DST) hours have been omitted as they contain either a reading of 0 MW or are the sum of two periods.

3.2 Weather variables

The defined data track of GEFCom2017 only allows dry bulb temperature and dew point temperature to be used as model predictors. Scatter plots of demand and these two temperature variables for Maine are shown in Figure 3. Note that a similar relationship is present in all other zones. Figure 4 shows that both variables are strongly correlated except at higher temperatures. It seems reasonable to expect improvement in predictive power by including both temperature variables within the model.

Each bottom level zone has data from one weather station for each of these two temperature variables. Naturally, aggregated zones have several stations available. Weather stations that belonged to an aggregated zone were averaged to obtain temperature variables. Using all weather variables separately was tested against this approach, but it was found that averaged temperature values performed similarly when validating on a test data set.

[1] This two month gap is in general consistent with how data arrived during the competition. To be clear, I do not expect that a two month gap between the end of the training period and start of the forecast horizon improves forecasts. A two month gap is only used to ensure reasonably consistency with competition proceedings.

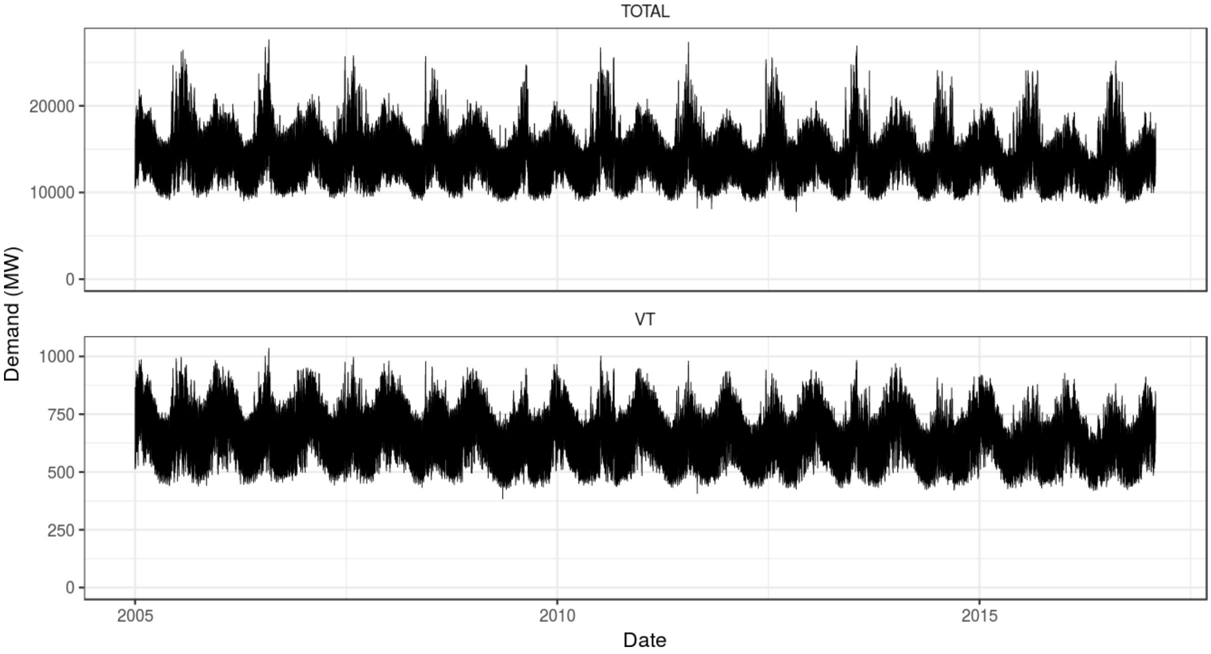

Figure 2: Electricity demand for the total of all zones and the bottom level zone Vermont. Strong seasonality and volatility is observed for both the total and the bottom level zone.

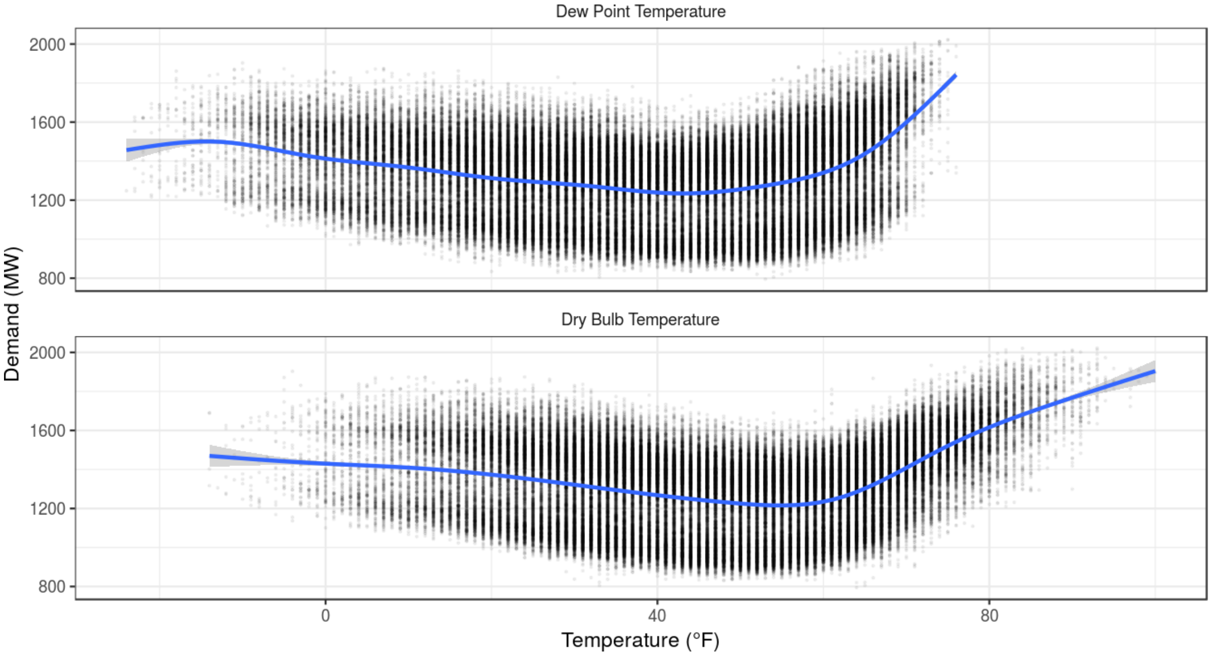

Figure 3: Scatter plots of demand and temperature variables in Maine.

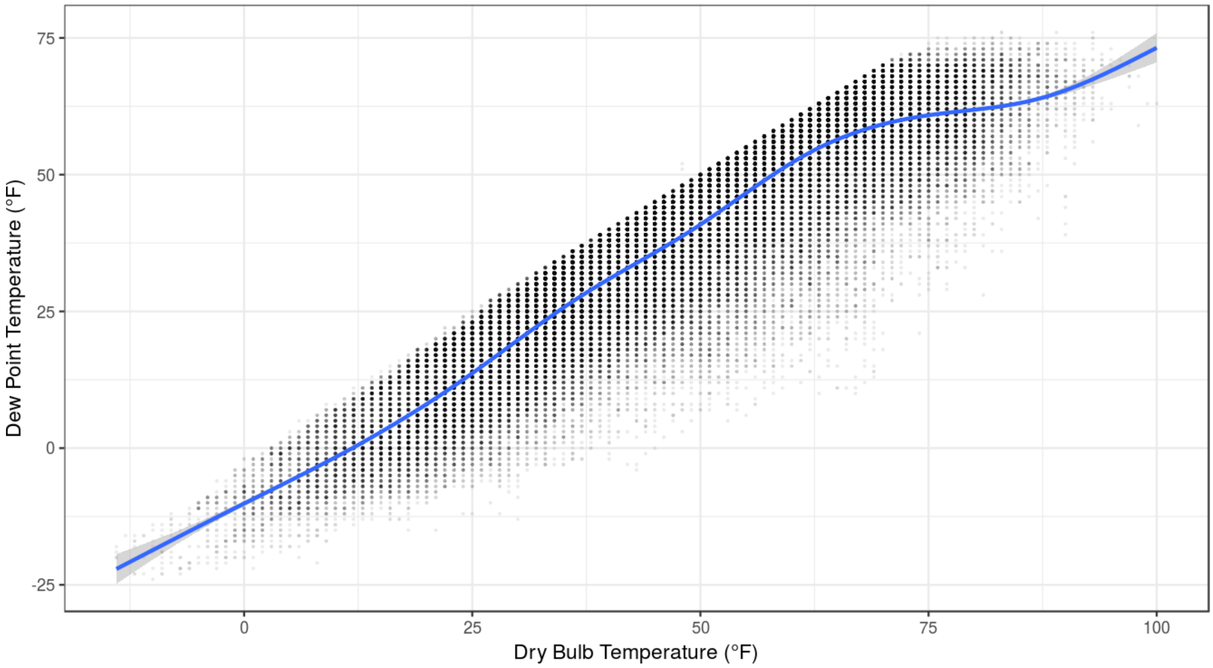

Figure 4: Correlation between dry bulb temperature and dew point temperature in Maine. A non-linear relationship is evident in this scatter plot.

3.3 Hierarchy structure

The hierarchy consists of 8 bottom level nodes and 2 aggregated nodes. It is an unbalanced hierarchy with 3 of the bottom level nodes combining to form Massachusetts and the remaining bottom level nodes and Massachusetts aggregating to form the total. A visualisation of of this structure is provided in Figure 1.

Figure 1 can be represented in matrix notation using the summing matrix **S** from (1). Expressing the GEFCom2017 hierarchy in the form of (1) gives,

$$\begin{bmatrix} y_{TOTAL,t} \\ y_{ME,t} \\ y_{NH,t} \\ y_{VT,t} \\ y_{CT,t} \\ y_{RI,t} \\ y_{MASS,t} \\ y_{SEMASS,t} \\ y_{WCMASS,t} \\ y_{NEMASSBOST,t} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} y_{ME,t} \\ y_{NH,t} \\ y_{VT,t} \\ y_{CT,t} \\ y_{RI,t} \\ y_{SEMASS,t} \\ y_{WCMASS,t} \\ y_{NEMASSBOST,t} \end{bmatrix},$$

where $y_{k,t}$ is the demand for zone k at time t.

4 Methodology

The following sections give a detailed description of the forecasting methodology.[2] For a given month I fit a separate model for each zone using a gradient boosting algorithm. I assessed the performance of L1 and L2 regularization using cross-validation. After selecting the regularization parameters that performed best I forecast demand for each zone over the forecast horizon using weather and residual simulations. This created demand simulations for each zone. Each demand simulation was reconciled to ensure that the sum of child nodes were equal to their parent nodes. The final step involved calculating quantiles of the demand simulations for each hour of the forecast horizon.

4.1 Training and test data sets

Each month between June 2016 and April 2017 (the final month of the competition) was used as a test data set. While only four test sets (January 2017 to April 2017) were assessed in the competition, this paper expands on this in order to compare the baseline (Vanilla) and boosted models across each month of an entire year.

Models were trained using data from January 2005 to two months prior to the start of the forecast period. This gap is in general consistent with how data arrived on the ISO New England website during the competition, where there was usually a two month processing time for new data. As discussed in Section 3.1 daylight saving time (DST) hours were omitted. The training data set was used when carrying out parameter tuning via 5-fold cross-validation. Residuals were calculated for the training set and were later used during the residual simulation step (see Section 4.3.2).

4.2 Model specification

4.2.1 Boosted model

I used a linearly boosted model from the XGBoost library (Chen, He, et al. 2017). Models were fit in R (R Core Team 2017) using the caret package (Kuhn 2017) to carry out cross validation and parameter tuning. A linear booster was chosen over a tree booster as both gave similar results but the linear booster typically ran faster. 5-fold cross-validation was used when tuning as this offered an acceptable compromise between computational burden and variation in the folds.

A similar approach to Ziel and Liu (2016) was used when choosing predictors. For zone k the following model was used,

$$y_{kt} = c_k\left(t\right) + f_k\left(\mathbf{w}_{kt}\right) + \epsilon_{kt}, \tag{5}$$

where at time t,

- y_{kt} is the demand;
- $c_k\left(t\right)$ is a linear function that models the effects of calendar variables;
- $f_k\left(\mathbf{w}_{kt}\right)$ is a linear function that models the effect of weather variables;
- $\mathbf{w}_{kt}$ is a vector containing all weather and lagged weather variables; and
- ϵ_{kt} is the model error.

[2] A tutorial with R code is available from camroach87.github.io.

Equation (5) is of the form discussed in Section 2.2. Calendar variables in $c_k(t)$ include,

- Public holidays;
- Hour of day;
- Day of week;
- Day of year; and
- A trend term which is a natural number ordering the observations.

Weather variables in $\mathbf{w}_{kt}$ include,

- Current dry bulb and dew point temperatures;
- 72 hourly lags for dry bulb temperature; and
- 72 hourly lags for dew point temperature.

The choice of 72 hourly lags was made somewhat arbitrarily. The main goal was to include temperature data from the previous three days to capture any thermal inertia effects in buildings. This is an important factor in energy demand (Ben Taieb and Hyndman 2014). More lags could well be added, though this would increase the computation time which I wished to avoid. In total there are 156 predictors. Note that predictors were not scaled prior to fitting models.

4.2.2 Vanilla model

For zone k, Tao's Vanilla model (Hong 2010) is,

$$\begin{aligned} y_{kt} =& \alpha_{0k} + \sum_{m=1}^{11} \alpha_{1km} M_{mt} + \sum_{d=1}^{6} \alpha_{2kd} D_{dt} + \sum_{h=1}^{23} \alpha_{3kh} H_{ht} + \sum_{d=1}^{6}\sum_{h=1}^{23} \alpha_{4kdh} D_{dt} H_{ht} \\ &+ \alpha_{5k}\text{Trend}_k + f_k(T_{kt}) + \epsilon_{kt} \end{aligned}$$

where at time t,

- T_{kt} is the dry bulb temperature;
- $f_k(T_{kt})$ models temperature effects;
- M_{mt} is a dummy variable for month $m \in \{1, 2, \ldots, 11\}$;
- D_{dt} is a dummy variable for day of week $d \in \{1, 2, \ldots, 6\}$;
- H_{ht} is a dummy variable for hour $h \in \{1, 2, \ldots, 23\}$; and
- Trend_k is a natural number that orders the observations.

The temperature effects are modelled by,

$$\begin{aligned} f_k(T_{kt}) =& \beta_{1k} T_{kt} + \beta_{2k} T_{kt}^2 + \beta_{3k} T_{kt}^3 \\ &+ \sum_{m=1}^{11} (\beta_{4km} T_{kt} + \beta_{5km} T_{kt}^2 + \beta_{6km} T_{kt}^3) M_{mt} \\ &+ \sum_{h=1}^{23} (\beta_{7kh} T_{kt} + \beta_{8kh} T_{kt}^2 + \beta_{9kh} T_{kt}^3) H_{ht}. \end{aligned}$$

Weather simulations for the models are constructed by shuffling historical weather data back and forwards a maximum of 4 days. Each historical year and shuffled time series within serves as a simulation. As I was attempting to simulate actual demand values I also simulated residuals using variable-length block bootstrapping. Residuals were not simulated in the Vanilla model which was consistent with the benchmark method of GEFCom2017.

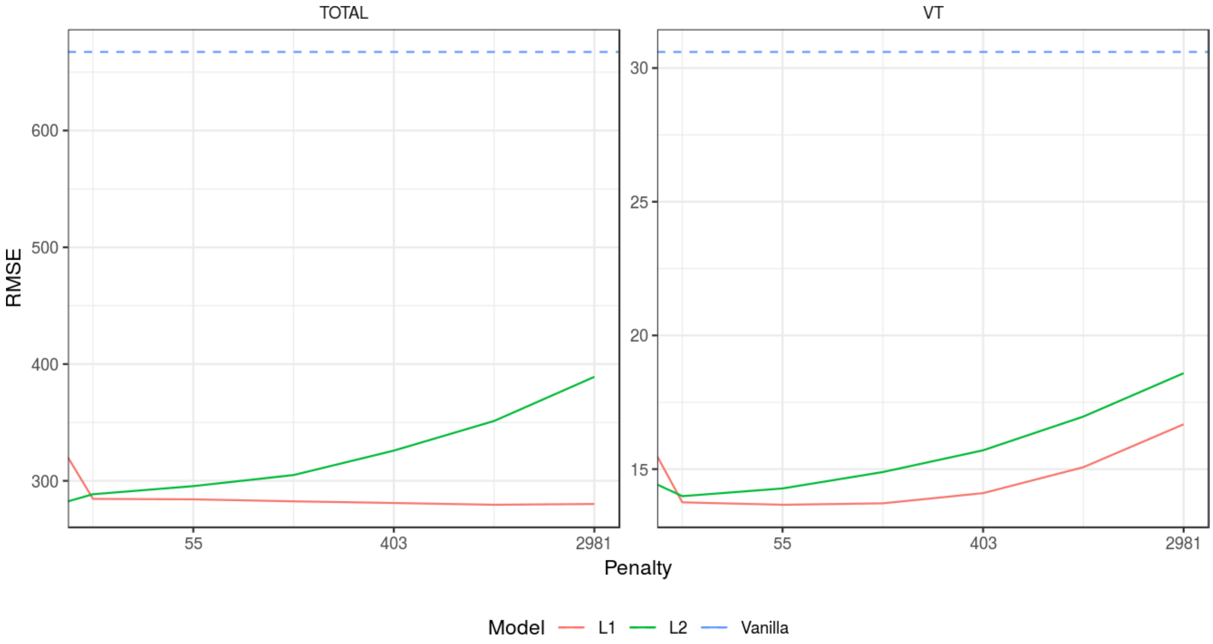

Figure 5: 5-fold cross-validation RMSE scores. The x-axis gives the magnitude of the penalty size for both the L1 and L2 regularization. Results for one aggregated zone and one bottom level zone are shown. Similar results are observed for all other zones.

4.2.3 Regularization

Due to the high dimensionality of our model there was a risk of over-fitting to the training data. To manage this risk, I fit several models with L1 and L2 regularization and different penalty values. 5-fold cross validation was then performed on the training data to pick the best model. These regularized models were also tested against a baseline model. The baseline model chosen was Tao's Vanilla model (Hong 2010) which has previously been used as a benchmarking model (Hong, Pinson, and Fan 2014) and was also used in GEFCom2017.

The change in root-mean-square error (RMSE) during 5-fold cross-validation is shown in Figure 5. Both the L1 and L2 regularized models outperformed the Vanilla model. With a sufficiently large penalty the L1 model gave the best RMSE results.

4.3 Simulating in a hierarchy

The challenge requires competitors to forecast 9 quantiles (10th, 20th, . . . and 90th) for every hour in a future month. This is an ex-ante forecasting problem as we do not have any data for predictors in this situation. To forecast a demand distribution I first simulated weather scenarios. Residuals were simulated by sampling from days with similar calendar characteristics.

As the New England zones form a hierarchy it is necessary to preserve correlation between them. For example, weather in one zone will be highly correlated with an adjacent zone. Hence, simulations need to reflect this. Correlation between zones is also present for residuals (Figure 7) and so care was taken when simulating residuals as well.

4.3.1 Weather simulations

Weather simulations were produced using the shifted-date method (Xie and Hong 2016). Historical weather time series were shifted back and forward by a maximum of 4 days each way. This resulted in 9 weather scenarios for each year. 11 years of historical weather data were used which gave a total of 99 weather scenarios. This approach has the advantage of ensuring realistic weather simulations are produced as well as preserving weather correlation between zones.

A double seasonal block bootstrap approach similar to Hyndman and Fan (2010) was tested against this shifting approach but was found to perform worse. This is most likely due to the unrealistic discontinuities that are introduced at block boundaries during the bootstrapping process. This was not an issue for their paper's goal of predicting maximum demand, but is in this instance.

4.3.2 Residual resampling

When predicting demand for simulated weather data the fitted model is only returning a conditional mean. The error term in (5) also needs to be accounted for. To do this, I sampled from the historical residuals and added this sample to the predicted demand. This combination of conditional mean and residual produced a realistic demand simulation. The historical residuals were calculated by predicting demand on the training data set and taking the difference between the predicted and actual demand.

When simulating residuals I sampled a sequence of historical residuals to preserve correlation between adjacent observations in the time series. A variable-length block bootstrapping approach similar to Hyndman and Fan (2010) was used. A block of variable length was sampled from historical years at close to the same point of the year. The day of year the the block started from was allowed to vary by as much as seven days from the day of year for which I required residuals. The length of the block was uniformly distributed between 14 and 21 days. These numbers were somewhat arbitrary and can be varied, but produced reasonably realistic auto-correlation functions (ACFs) when compared to the actual (see Figure 8 for an example). Correlations in the simulated residuals tend to be lower than the actuals due to discontinuities introduced at the borders of the blocks. To try and reduce the magnitude of the discontinuities, block boundaries occurred at midnight when the variance of the residuals was usually lowest (Figure 6).

It was also important to make sure that whatever dates were chosen when resampling were consistent between zones. Sampling different historical dates for each zone would lead to a break down in the inter-zone residual correlation resulting in less realistic simulations. Residual correlations between zones are shown in Figure 7.

To check the sampled residuals form a realistic time series their ACF is compared against the ACF of the historical data (Figure 8). The simulated residuals appear to have a similar ACF as the actuals. The ACF of the simulated residuals are lower than the actuals' ACF as expected, but to an acceptable degree.

4.4 Hierarchical reconciliation

Once a demand simulation has been created it is necessary to reconcile all of the time series in the hierarchy. Here I test several methods of accomplishing this.

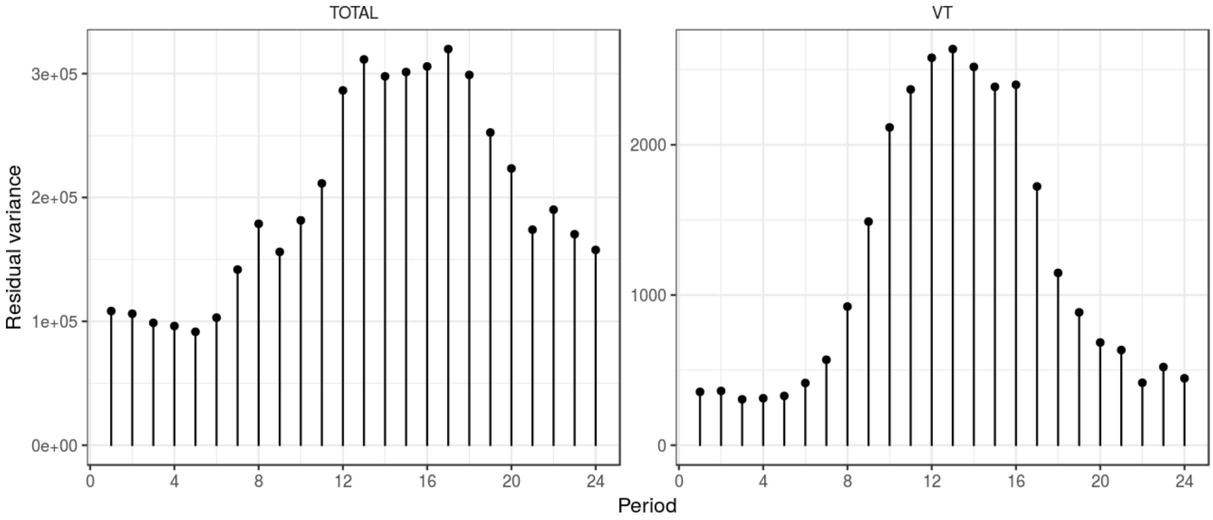

Figure 6: Residual variance for each hourly period of the day. Variances have been calculated using residuals from all 12 training data sets. Results for one aggregated zone and one bottom level zone are shown though similar residual variance behaviour is observed in all other zones. Residual variance is highest during the middle of the day and lowest close to midnight.

4.4.1 Choosing weights for reconciliation

Several methods for reconciling the hierarchy were tested. The first involved using only the summing matrix $\mathbf{S}$ as per (2). The other two methods were based on specifying different weight matrices in (4).

As mentioned, the entities of $\mathbf{W}$ can be calculated based on the variances of ϵ_h. This works well with time series models such as ARIMA and exponential smoothing where each fitted value is already a one-step ahead forecast, but for our model this is not the case. Computing one-step ahead forecasts for the historical data would require refitting the model at each step which is computationally prohibitive. As an alternative, I propose two different weight matrices. The first is based on the mean values of each zone's demand and the second is calculated from variance of the residuals. The inverse matrices are specified below,

$$\mathbf{W}_{\text{mean}}^{-1} = \frac{1}{\sum_k \bar{y}_k} \cdot \operatorname{diag}\left(\{\bar{y}_k\}_{k=1}^{K}\right),$$

$$\mathbf{W}_{\text{var}}^{-1} = \operatorname{diag}\left(\left\{\sigma_k^2\right\}_{k=1}^{K}\right),$$

where $\operatorname{diag}\left(\{x_k\}_{k=1}^{K}\right)$ represents a diagonal matrix with elements $x_1, x_2, \ldots, x_K$, K is the total number of zones and σ_k^2 is the variance of the residuals for zone k.

The intuition behind these weights follows from our goal of shifting the more accurate forecasts less than the inaccurate forecasts when reconciling. In the absence of one-step ahead forecasts the variance of residuals should serve as a useful proxy. Since residuals and demand are correlated the mean weight matrix may also prove useful.

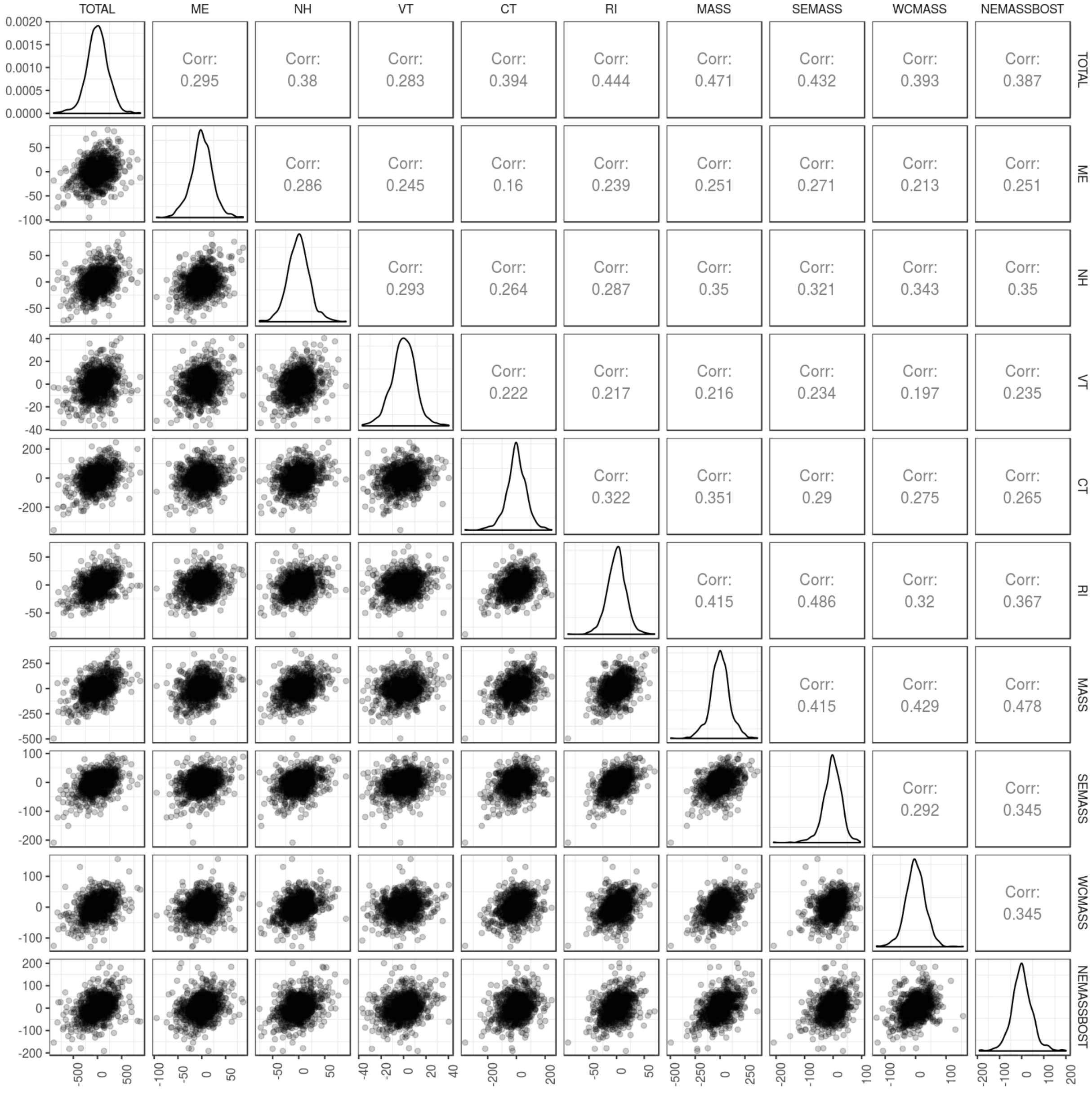

Figure 7: Correlation of zone residuals based on 1,000 points sampled from the hierarchy. Positive correlation is observed between all zones.

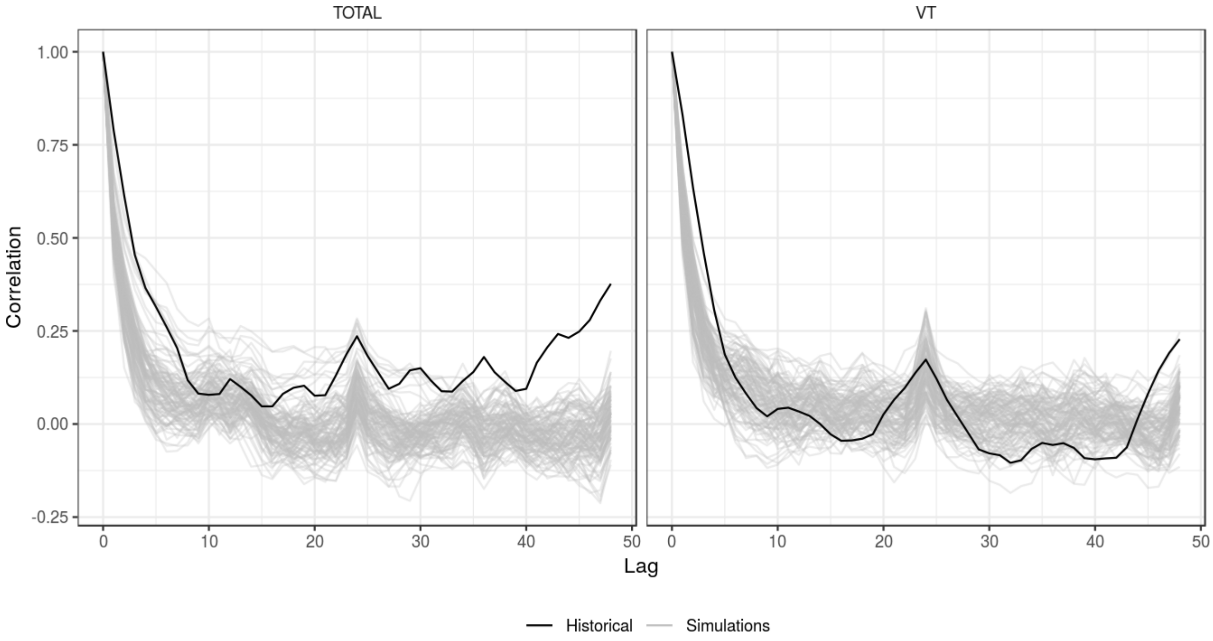

Figure 8: Auto-correlation functions for historical residuals and simulated residuals in February 2017 for one aggregated zone and one bottom level zone.

5 Discussion

5.1 Reconciliation results

Monthly RMSE scores for each hierarchical reconciliation method are shown in Figure 9 and Table 1. All 12 test data sets have been used to produce these results. Overall, the WLS approach using $\mathbf{W}_{\text{var}}$ gives the best performance. The WLS approach using $\mathbf{W}_{\text{mean}}$ performed slightly worse. Using the OLS approach is the worst performed of the three reconciliation methods.

To see why the OLS approach performs worse than the WLS approaches we can compare the forecasts. Figure 10 shows one of the simulations' base demand forecasts and the reconciled forecasts for one day in the forecast period. While the aggregated zone's reconciled forecasts look reasonable the bottom level forecast has severe variance introduced when using the OLS methodology. This variance appears in bottom level zones that only have one parent zone (Total). It so happens that OLS adjustments made to base forecasts for these bottom level zones are of a comparable magnitude to the adjustments made to the Massachusetts aggregated zone, whereas the bottom level zones making up Massachusetts receive significantly smaller adjustments. This discrepancy in adjustments appears to be caused by the unbalanced structure of the hierarchy.

Given these results the WLS reconciliation method using $\mathbf{W}_{\text{var}}$ as the weight matrix was chosen to reconcile base forecasts.

5.2 Quantile forecast results

Quantile forecasts were produced with the WLS reconciliation approach using $\mathbf{W}_{\text{var}}$ weights and L1-regularization as this model appeared to perform best. An example of the quantile forecasts for one aggregated zone and one bottom level zone is shown in Figure 11. By inspection it appears as though the quantile forecasts are capturing the variance in the actuals well. Benchmarking is carried out against the Vanilla model to better understand how well the model is performing.

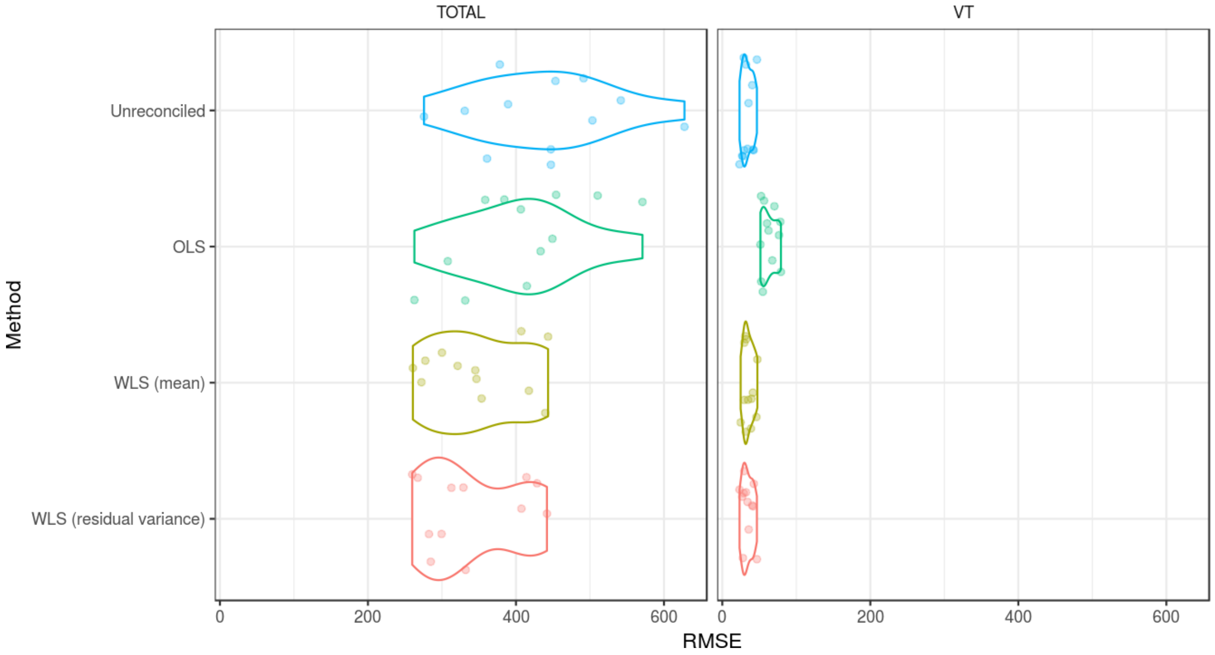

Figure 9: Hierarchical reconciliation results. The RMSE for each forecast month from May 2016 to April 2017 is plotted. The contoured lines are violin plots and represent the density. Similar results are observed in all other zones.

Table 1: RMSE scores for each reconciliation method averaged across all zones.

Month	Unreconciled	OLS	WLS (mean)	WLS (residual variance)
May 2016	178.1	170.6	138.8	124.5
June 2016	175.4	165.7	135.5	127.9
July 2016	238.0	226.4	188.8	178.7
August 2016	204.0	191.6	160.4	154.7
September 2016	171.6	164.7	144.5	139.8
October 2016	117.5	111.0	107.2	115.0
November 2016	133.4	125.2	113.2	114.4
December 2016	204.4	199.2	178.5	170.7
January 2017	140.6	132.0	114.7	112.0
February 2017	150.5	144.8	131.3	128.0
March 2017	178.1	172.3	165.6	169.4
April 2017	166.5	158.8	160.3	171.1

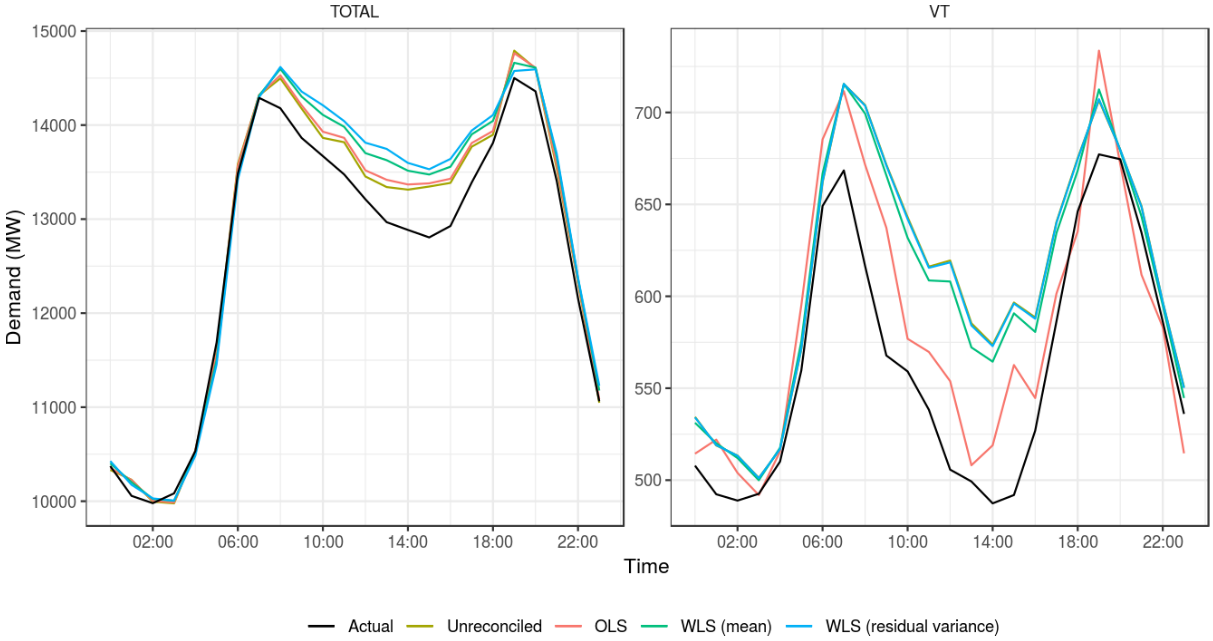

Figure 10: Original and reconciled forecasts using different weights. Note that I have deliberately chosen a day where over-forecasting occurs to better show how the OLS reconciliation method introduces variance.

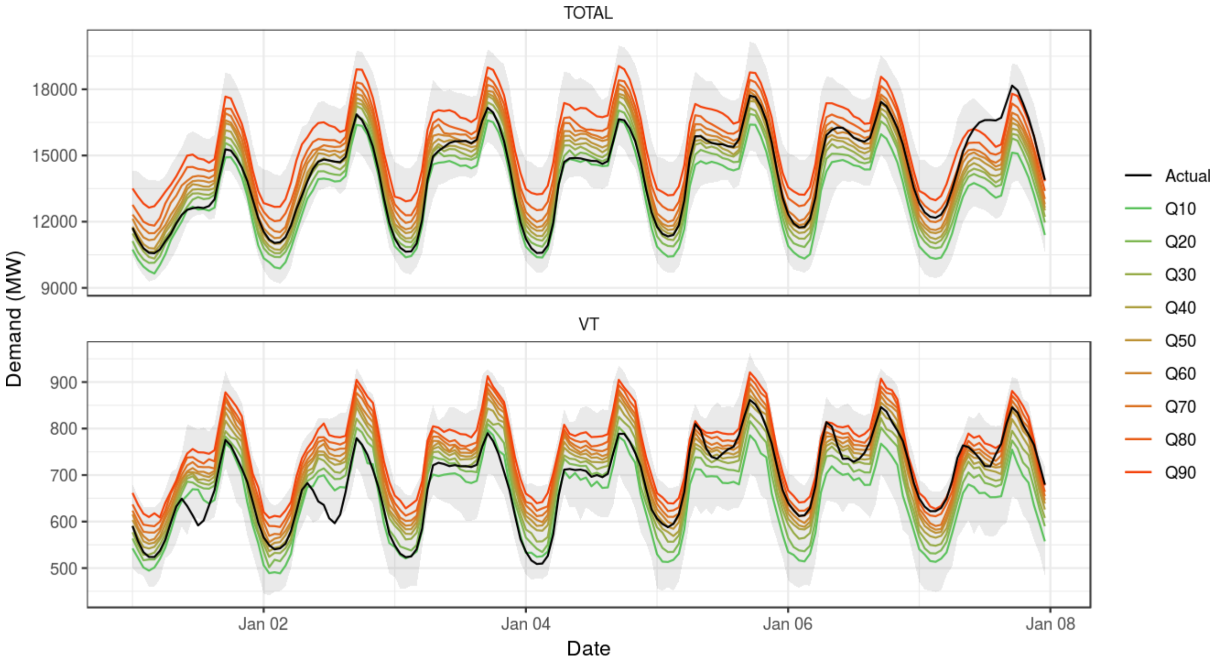

Figure 11: Actuals and quantile forecasts in first week of January 2017. The shaded area shows the maximum and minimum simulated demand values.

Table 2: Expected pinball loss scores for each zone averaged across all 12 test sets. Lower values indicate better performance.

Zone	Boosted	Vanilla	Percentage improvement
CT	99.9	108.3	7.8%
MASS	159.6	185.9	14.2%
ME	19.8	22.7	12.6%
NEMASSBOST	69.6	81.6	14.7%
NH	30.8	32.6	5.5%
RI	25.2	27.5	8.4%
SEMASS	47.9	55.4	13.5%
TOTAL	330.5	375.2	11.9%
VT	15.0	18.6	19.4%
WCMASS	48.3	55.0	12.1%

5.3 Benchmarking against Vanilla model

The pinball loss scoring function can be used to assess quantile forecasts (Gneiting 2011). For a probability level, τ, the pinball loss function is defined as,

$$L_\tau\left(y, q_\tau\right) = \begin{cases} \tau(y - q_\tau) & \text{for } y \geq q_\tau, \\ (1-\tau)(q_\tau - y) & \text{for } q_\tau > y. \end{cases}$$

A lower expected pinball loss score indicates better performance. The expected pinball loss for each model and zone can be estimated by taking the mean of all observed $L_\tau\left(y_{kt}, q_{kt\tau}\right)$, where $q_{kt\tau}$ is the quantile forecast at probability level τ for zone k at time t.

A comparison of the Vanilla and boosted[3] models is given in Tables 2 and 3. The boosted model almost always outperforms the Vanilla model. The only exception is for August 2016 when both models appear to have poor performance relative to other months.

5.4 Future research

Here I have explored the performance of the boosted algorithm in one context. However, it is potentially interesting to see how such a model might perform when forecasting over different horizons. Another area that might be of interest is focusing on other methods for dealing with unbalanced hierarchies, for example - adding artificial nodes to balance the hierarchy. Both of these are left as future research topics.

6 Conclusion

In this paper I have presented a methodology for producing probabilistic hierarchical forecasts. A demand model based on linear gradient boosting was shown to outperform a commonly used benchmark model. Additionally, the impact of both L1 and L2 regularization was found to improve the model fit. The best performance was observed using a sufficiently large L1 penalty.

[3] WLS reconciliation approach using $\mathbf{W}_{\text{var}}$ weights and L1-regularization.

Table 3: Expected pinball loss scores for each forecast month averaged across all zones. Lower values indicate better performance.

Month	Vanilla	Boosted	Percentage improvement
May 2016	74.3	53.5	28.1%
June 2016	75.8	72.7	4.2%
July 2016	160.5	149.2	7.1%
August 2016	168.0	175.2	-4.3%
September 2016	128.7	119.3	7.3%
October 2016	47.4	33.5	29.2%
November 2016	62.8	39.1	37.8%
December 2016	77.3	66.4	14.1%
January 2017	102.4	87.3	14.8%
February 2017	101.8	83.5	18.0%
March 2017	94.9	84.8	10.7%
April 2017	60.0	49.6	17.3%

Weather simulations were produced by shifting the weather history back and forth by up to four days. Residual simulations used a variable-length block bootstrapping approach. Forecast reconciliation between nodes of the hierarchy was carried out using several different methods. It was found that using a weight matrix based on the variance of residuals performed best. Advantages of this approach are that bottom level zonal forecasts correctly sum to aggregated zonal forecasts and forecast accuracy improves compared to unreconciled models.

Finally, the quantile forecasts produced by the gradient boosted model outperformed a commonly used baseline model. Quantile forecasts were assessed using the pinball loss function. The gradient boosted model performed better in all zones in the hierarchy over a year of monthly forecasts.

References

Ben Taieb, Souhaib and Rob J Hyndman (2014). “A gradient boosting approach to the Kaggle load forecasting competition”. In: *International Journal of Forecasting* 30.2, pp. 382–394. URL: http://linkinghub.elsevier.com/retrieve/pii/S0169207013000812.

Ben Taieb, Souhaib, James W Taylor, and Rob J Hyndman (2017). “Hierarchical Probabilistic Forecasting of Electricity Demand with Smart Meter Data”. In: URL: https://robjhyndman.com/papers/HPFelectricity.pdf.

Chen, Tianqi and Carlos Guestrin (2016). “XGBoost: A Scalable Tree Boosting System”. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, New York, USA: ACM, pp. 785–794.

Chen, Tianqi, Tong He, et al. (2017). *xgboost: Extreme Gradient Boosting*. R package version 0.6-4. URL: https://CRAN.R-project.org/package=xgboost.

Friedman, Jerome (2001). “Greedy Function Approximation: A Gradient Boosting Machine”. In: *Annals of statistics* 29.5, pp. 1189–1232. URL: http://www.jstor.org/stable/2699986.

Friedman, Jerome, Trevor Hastie, and Robert Tibshirani (2000). “Additive logistic regression: a statistical view of boosting (With discussion and a rejoinder by the authors)”. en. In: *Annals of statistics* 28.2, pp. 337–407. URL: http://projecteuclid.org/euclid.aos/1016218223.

Gneiting, Tilmann (2011). “Quantiles as optimal point forecasts”. In: *International journal of forecasting* 27.2, pp. 197–207. URL: http://dx.doi.org/10.1016/j.ijforecast.2009.12.015.

Hong, Tao (2010). “Short term electric load forecasting”. PhD thesis. North Carolina State University.

Hong, Tao, Pierre Pinson, and Shu Fan (2014). “Global energy forecasting competition 2012”. In: *International Journal of Forecasting* 30.2, pp. 357–363. URL: http://dx.doi.org/10.1016/j.ijforecast.2013.07.001.

Hong, Tao, Pierre Pinson, Shu Fan, et al. (2016). “Probabilistic energy forecasting: Global Energy Forecasting Competition 2014 and beyond”. In: *International Journal of Forecasting* 32.3, pp. 896–913. URL: http://dx.doi.org/10.1016/j.ijforecast.2016.02.001.

Hong, Tao, Jingrui Xie, and Jonathan Black (2019). “Global Energy Forecasting Competition 2017: Hierarchical Probabilistic Load Forecasting”. In: *International journal of forecasting.*

Hyndman, Rob J, Roman A Ahmed, et al. (2011). “Optimal combination forecasts for hierarchical time series”. In: *Computational statistics & data analysis* 55.9, pp. 2579–2589. URL: http://linkinghub.elsevier.com/retrieve/pii/S0167947311000971.

Hyndman, Rob J and Shu Fan (2010). “Density forecasting for long-term peak electricity demand”. In: *IEEE Transactions on Power Systems* 25.2, pp. 1142–1153. URL: http://ieeexplore.ieee.org/document/5345698/.

Hyndman, Rob J, Alan J Lee, and Earo Wang (2016). “Fast computation of reconciled forecasts for hierarchical and grouped time series”. In: *Computational statistics & data analysis* 97, pp. 16–32. URL: http://dx.doi.org/10.1016/j.csda.2015.11.007.

Koren, Yehuda (2009). “The BellKor solution to the Netflix grand prize”. In: *Netflix prize documentation* 81.August, pp. 1–10. URL: http://www.stat.osu.edu/~dmsl/GrandPrize2009_BPC_BellKor.pdf.

Kuhn, Max (2017). *caret: Classification and Regression Training.* R package version 6.0-76. URL: https://CRAN.R-project.org/package=caret.

R Core Team (2017). *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing. Vienna, Austria. URL: https://www.R-project.org/.

Schapire, Robert E (1990). “The Strength of Weak Learnability”. en. In: *Machine learning* 5.2, pp. 197–227. URL: https://link.springer.com/article/10.1023/A:1022648800760.

Wickramasuriya, Shanika L, George Athanasopoulos, Rob J Hyndman, et al. (2015). “Forecasting hierarchical and grouped time series through trace minimization”. In: *Department of Econometrics and Business Statistics, Monash University* Working Paper 15/15. URL: http://business.monash.edu/old/econometrics-and-business-statistics/research/publications/ebs/wp15-15.pdf.

Xie, Jingrui and Tao Hong (2016). “Temperature Scenario Generation for Probabilistic Load Forecasting”. In: *IEEE transactions on smart grid* PP.99, pp. 1–1. URL: http://dx.doi.org/10.1109/TSG.2016.2597178.

Ziel, Florian and Bidong Liu (2016). “Lasso estimation for GEFCom2014 probabilistic electric load forecasting”. In: *International journal of forecasting* 32.3, pp. 1029–1037. URL: http://dx.doi.org/10.1016/j.ijforecast.2016.01.001.