Short Term Electric Load Forecasting

ROMAIN JUBAN

March 12, 2014

1 Exploring and cleansing the data

The goal is to forecast hourly load for a US utility given temperature. We need to predict the load for 20 zones plus the aggregate load, which represents the total demand for the utility. Load and temperature data history ranges from 2004/01/01 01:00 to 2008/06/30 06:00. Given actual temperature history, 8 weeks in the load history are missing and are required to be backcasted[1]. In addition, we need to forecast hourly loads from 2008/7/1 to 2008/7/7. No temperature data is given for this week[2].

1.1 Load data

The load data includes hourly loads (in kW) for a US utility divided into 20 zones. Fig. 1 shows the average demand for all zones during the period covered by the data. We can see that the demand varies greatly across zones.

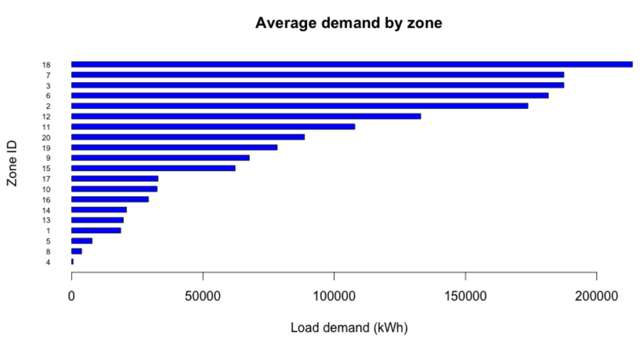

Figure 1: Average demand for each zone

Demand for Zone 9 is very erratic and does not seem to be related to the temperature values. Zone 4 has outliers in demand as we can see on Fig. 2. We replaced them by the mean of the data for Zone 4. Zone 10 has a big jump in demand in year 2008 (Fig. 3). We can take the mean before and after the jump and take the difference to normalize the load profile before the prediction. The jump will be added to the final prediction.

[1] 2005/3/6-2005/3/12, 2005/6/20-2005/6/26, 2005/9/10-2005/9/16, 2005/12/25-2005/12/31, 2006/2/13-2006/2/19, 2006/5/25-2006/5/31, 2006/8/2-2006/8/8, 2006/11/22-2006/11/28

[2] Temperature forecasting will not be studied in this paper

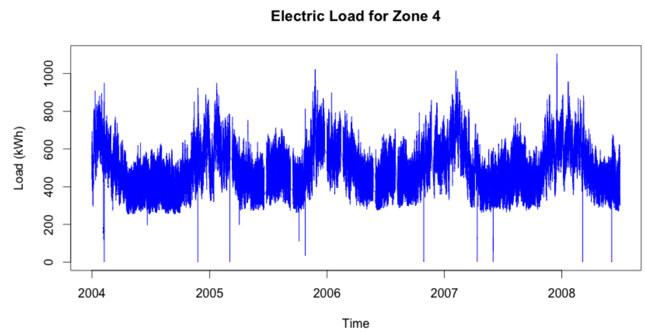

Figure 2: Load profile for Zone 4

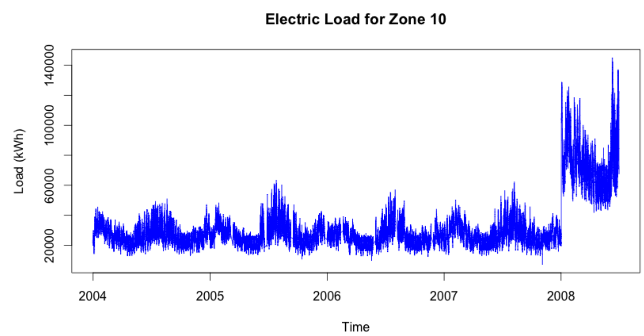

Figure 3: Load profile for Zone 1

1.2 Temperature data

The temperature data is taken from 11 different weather stations and is measured in ($^\circ F$). We identified outliers in the temperature data for site 8 and replaced them with the mean data for the same day. We do not know *a priori* the relative location of those weather stations with respect to the 20 zones of the US utility.

2 Data processing and relevant features

Fig. 5 shows an overall increasing trend year by year. This trend may be due to climate change and/or human activities[3]. We define a quantitative variable *Trend* to capture the increasing trend by assigning a natural number to each hour in the

[3] energy demand and consumption increase over the years

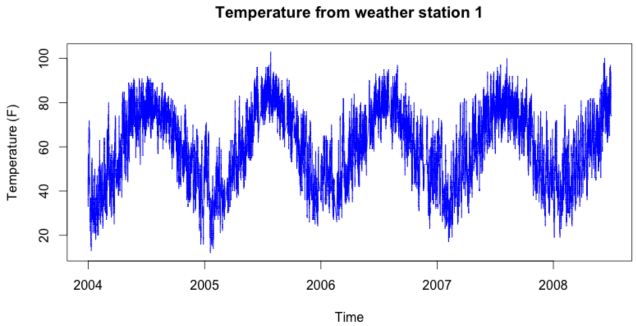

Figure 4: Time series of temperature for station 1

natural order (1,2,3,..).

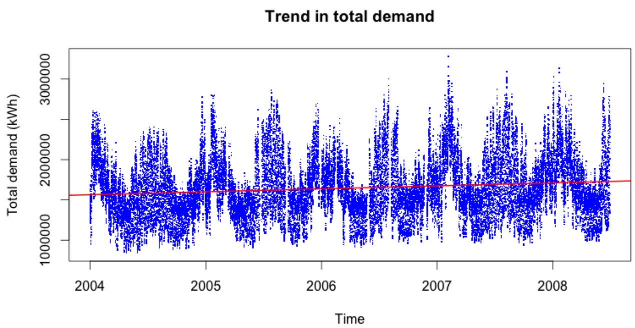

Figure 5: Trend of the total demand

Electricity demand follows a seasonal cycle as we can see on Fig. 6. There is a clear time-of-year effect in the demand data, with two peaks in Winter (February) and Summer (July). This shows that the electricity demand is highly correlated to the temperature. The winter peak load occurs during the valley temperature period in winter when electricity is used for heating ($T < 15°C$)and the summer peak load occurs during the peak temperature period in summer when electricity is used for cooling ($T > 20°C$). Fall and Spring seem to have a lower demand, when the temperature is average (no need for heating nor cooling).

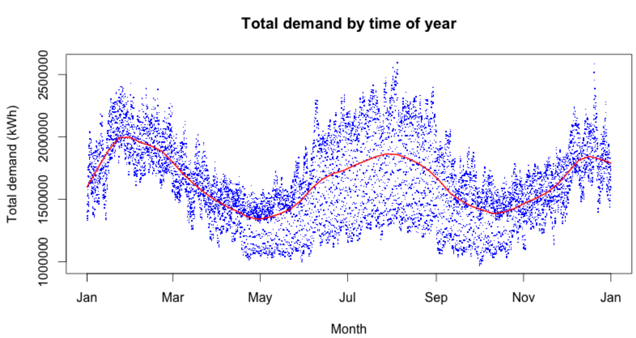

Figure 6: Total demand averaged over one year

Electricity demand is subject to the day of the week. Load consumption behavior is different during weekends and weekdays. Boxplots of the demand by day of the week are shown in Fig. 7. We can also include in our model the month of the year that can model large-scale effects such as seasons and special periods of the year[4]. Boxplots of the demand by month of the week show this relationship in Fig. 8. We define variables *Weekday* and *Month* to capture those effects.

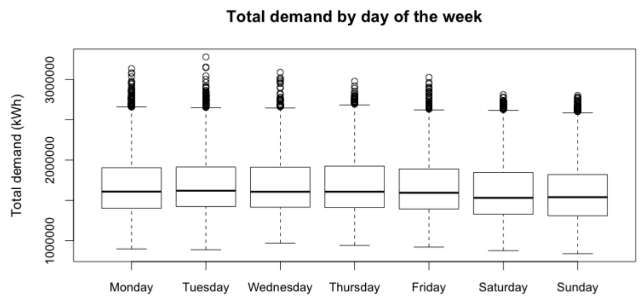

Figure 7: Total demand by day of the week

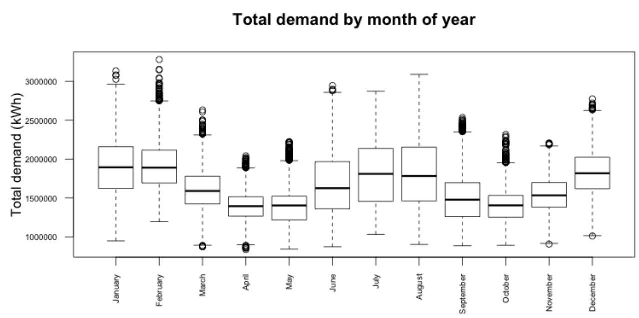

Figure 8: Total demand by month of the year

We now see how demand changes with the time of day for weekdays and for weekends in Fig. 9 and 10. Patterns are different during the working hours, as expected, with peaks in demand around 8am and 5pm. The night-time patterns are quite similar. This suggests to add an interaction term between time of day and day of week in the model. We thus add the variables *Hour* and *Hour x Weekday*

3 Temperature data

The scatterplot on Fig. 11 shows that the relationship between the load and the temperature is non-linear. As expected, load tends to be higher for extreme temperatures, while it is quite low for medium temperature. The relationship relies on met-

[4] summer break, Christmas break,...

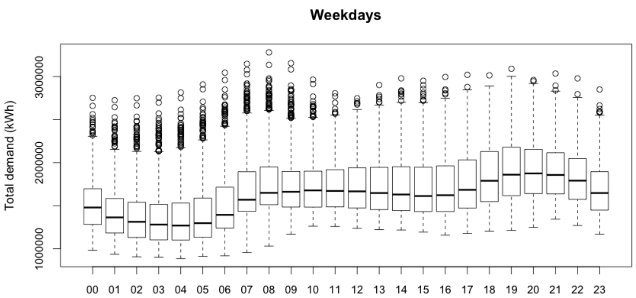

Figure 9: Total demand by hour of the day (Weekdays)

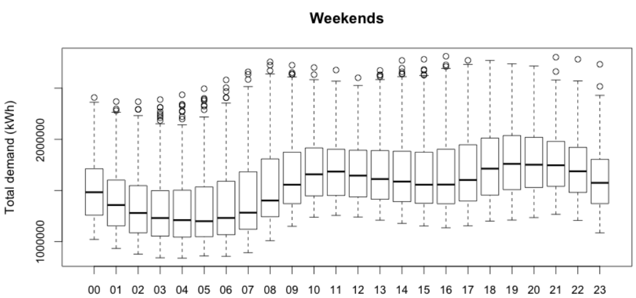

Figure 10: Total demand by hour of the day (Weekend)

rics such as HDD and CDD[5] that depend on many criteria such as the location, type of insulation, HVAC system, people's habits,... However a cubic function seems to fairly capture the relationship between load and temperature so 3rd ordered polynomials will be used in the model. Thus we will include variables T, T^2, T^3 as well as interaction terms *Hour x* T, *Hour x* T^2, *Hour x* T^3 and *Month x* T, *Month x* T^2, *Month x* T^3 in our model

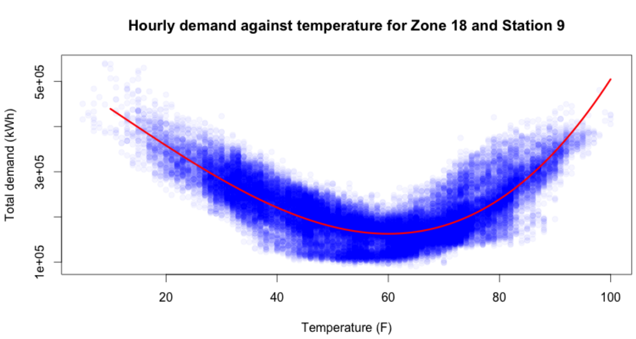

Figure 11: Hourly load as a function of the temperature for Zone 18 and Station 9

[5]Heating Degree Days and Cooling Degree Days are measurements designed to reflect the demand for energy needed to heat and cool a building

4 Forecasting methodology

The formula for the model that will be used to forecast the load contains 10 features (9 variables + intercept term).

$Load = \theta_0 + \theta_1\, Trend + \theta_2 Day\ x\ Hour + \theta_3 Month + \theta_4 Month\ x\ T + \theta_5 Month\ x\ T^2 + \theta_6 Month\ x\ T^3 + \theta_7 Hour\ x\ T + \theta_8 Month\ x\ T^2 + \theta_9 Month\ x\ T^3$

In mathematical notation, this can be written as:

$$F_i = \sum_{j=0}^{9} \theta_j X_j^{(i)} = \theta^T X^{(i)}$$

where F_i denotes the hourly load forecasted at time i and $X^{(i)}$ the vector of features at time i.

We will use a bottom-up approach: We forecast each zone independently and then sum them up to obtain forecasts for the aggregate. Backcasting and forecasting are obtained with a similar approach. We have 20 zones and 11 weather stations. To determine which weather station to use for each zone, we will train 11 models (one for each weather station) on 2/3 of the training data. Then, we will test those models on the remaining 1/3 of the training data (validation set). We can compute the RMSE[6] for those 11 models. The weather station selected is the one that leads to the lowest RMSE. Thus we are able to determine which station to use for each zone. It is interesting to notice that we have no information about the spatial location of zones and weather stations and our approach uses machine learning techniques without any spatial interpolation to find the most relevant weather station.

5 Results analysis

5.1 Multiple Linear Regression

To determine the coefficients θ_k we have trained our model by minimizing

$$RSS(\theta) = \sum_{i \in training} (F_i - O_i)^2$$

where F_i is the forecasted load obtained by using the formula above and O_i is the actual hourly load observed. A prediction (backcasting or forecasting) is calculated using a new vector of features at a future time i: $F_i = \theta^T X^{(i)}$

5.2 Exponentially Weighted Least Squares

Human's energy consumption is a time-varying variable so higher weights should be assigned to the recent observations than the older ones. Therefore, we can perform a multiple linear regression assigning different weights for each observation. The first hour in the history has an unit weight and the

[6]Root Mean Squared Error defined as RMSE $= \sqrt{\frac{1}{N}\sum_{i=1}^{N}(O_i - F_i)^2}$ where O_i is the observed load at time i and F_i is the forecasted load at time i

n^{th} hour in the history has the weight λ^{n-1}, where $\lambda > 1$ is the weighting factor. The best lambda is chosen by cross-validation[7]. As we can see on Fig. 12, its corresponds to a value of 1.00015

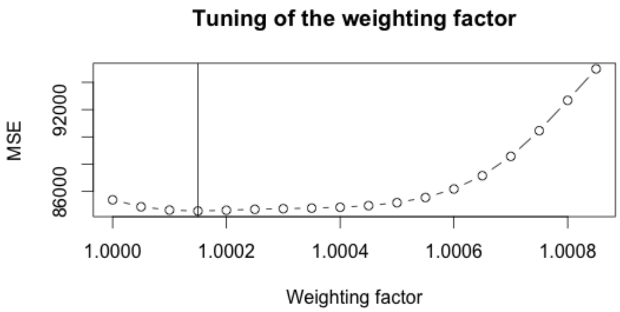

Figure 12: Determination of the optimal weighting factor by cross-validation

5.3 Shrinkage methods: Lasso and Ridge

As an alternative to a Simple Linear Regression, we can fit a model using techniques that shrink the coefficient estimates towards zero, reducing their variance and making more stable predictions. *Ridge* regression minimizes

$$RSS(\theta) + \lambda||\theta||_2^2$$[8]

The *Lasso* in addition to constraining the coefficient estimates, performs feature selection by setting certain coefficients exactly to 0. It minimizes

$$RSS(\theta) + \lambda||\theta||_1$$[9]

λ is a tuning parameter that controls the shrinkage of the coefficients. Those methods did not reduce significantly the variance of the model but increased the bias, resulting in bad predictions.

5.4 Random Forests

Random Forests builds a large number of decision trees by generating different bootstrapped training data sets and averages all the predictions. But when building these trees, each time a split in a tree is considered, a random sample of m predictors is chosen from the full set of p predictors. The classifiers may be weak predictors when used separately, but much stronger when combined with other predictors. Randomness allows weak predictors to be taken into account and decorrelates the trees. Two tuning parameters are needed to build a Random Forests algorithm: the total number of trees generated and the number of features randomly selected at each split when building the trees. Typically, a good value for m is around $\sqrt{p}$ where p is the number of predictors (9 in our case), that is 2 or 3 predictors. The number of trees can be chosen by cross-validation on the training dataset but a large number of trees will not overfit the data, so we can pick a forest quite large (1000 trees).

5.5 Results

The forecasting accuracy is evaluated by weighted root mean square

- Multiple Linear Regression: 86848
- Weighted Least Squares: 84568
- Ridge/Lasso Regression: 86848
- Random Forests $m = 2, ntree = 100$: 79146
- Winner of the competition: 60777

6 Possible improvements

It is also important to consider lagged temperature as well as current temperature due to buildings' thermal inertia. In addition to the current temperature $T(t)$ we can incorporate temperature data from the 3 previous timestamps ($T(t), T(t-1), T(t-2)$ and $T(t-3)$) in a weighted moving average $T_{weighted}(t) = \frac{1}{\sum_{k=1}^{3} \alpha^{k-1}} \sum_{k=1}^{3} \alpha^{k-1} T(t-k)$. A 3-hour temperature window will most likely to be "remembered" by the load in the next hour.

Holiday effect on load profile is similar to weekend effect in many aspects due to close of business and office buildings. Holiday surrounding days can be affected due to extended holiday activities (Black Friday). Effects are difficult to predict because some holidays may vary year by year whereas some others do not and impacts of holidays vary from one to another.

For the eight in-sample weeks, where we were required to backcast the load, we can use both data before and after the week to be forecasted to have a more accurate prediction. In order to do that, we can fit two forecasting models. The first model is estimated in the usual way, using data available up to the start of the in-sample week. The second model reverses the time ordering of the data and is equivalent to backcast using data after the end of the in-sample week. We expect the forecasts to do best at the beginning of the week, and the backcasts to do best at the end of the week, because they involve data which are closer to the days being predicted. Therefore, after estimating the two sets of models, we can take a weighted combination of the two sets of forecasts in order to produce the final forecasts.

[7]Cross-validation is a model validation technique where a fraction of the training set is used to train the model and the remaining part is used to test the model (validation set) in order to assess the model and limit problems like overfitting

[8]$||.||_2$ is the L_2 norm

[9]$||.||_1$ is the L_1 norm