Predicting Future Energy Consumption

CS229 Project Report

Adrien Boiron, Stephane Lo, Antoine Marot

Abstract

Load forecasting for electric utilities is a crucial step in planning and operations, especially with the increasingly stressed utilization of equipment. The objective of this project was to accurately backcast are forecast hourly energy loads (in kW) for a US utility for 20 zones using more than 4 years of historical loads and temperature data from 11 weather stations. Several multivariate linear regression models were developed and reliably predicted energy loads with down to 5% mean training and testing errors using a Leave One Week Out Cross-Validation (LOWOCV) algorithm over the 4.5 years of data available. Along the way, the data was notably treated to remove problematic events for training such as outages, and to fit sudden jumps in load.

Indicator Model

As part of the kaggle competition GEFCom2012 [1], historical hourly energy loads data were provided for 20 zones, without specific information, along with temperature data for 11 weather stations. Our first approach consisted in training a multiple linear regression model on the entire dataset available, with different combination of the following features: temperature, month, day of the week, hour, and trend. This model – called the indicator model – could be written as:

$$Y_t = \theta_0 + \theta_1 Trend + \theta_2 T_t + \theta_3 T_t^2 + \theta_4 T_t^3 + \theta_5 Month + \theta_6 Day + \theta_7 Hour$$

Note that in the cases of the Month, Day, and Hour features, the predictor variables are qualitative instead of quantitative. For example, the θ_5 could be seen as a vector having a different value for each month, and $Month$ as an indicator retrieving the right constant as follows:

$$X_1 = 1, if\ the\ month\ is\ January \mid X_1 = 0\ if\ not$$

Regarding the temperature law, a 3rd degree polynomial was found to provide the best fit, as it is also in the literature [2]. Relationship between weather stations and zones was not available. It was therefore necessary to correlate weather stations and zones in order to use the appropriate features. The selection of the most correlated stations for each zone was done using data during early morning hours in winter, since we found out through analysis that was the time of highest correlation between temperature and load (logical since no human activity, except heating relied to temperature).

Let us note that zone 9 will mostly be left out in our discussion. It is because the load history in zone 9 is very different in nature to what is found in all other zones, such that a specific and different model would be needed for zone 9. Since time was lacking to investigate this zone properly, it is left aside in this analysis.

Additional features such as the interaction between Month/Temperature, Hour/Temperature, and Day/Hour were progressively and similarly added to improve this model. A Leave One Week Out Cross-Validation (LOWOCV) algorithm was written to test the indicator model, and the mean training (and testing) error of the most complex one was found to be of more than 6.5% (see Results) - which was not satisfying. That is why we went on to a more elaborated model that also optimizes the temperature polynomial and trend coefficient for each season, day and hour. That is the full model we will now describe.

Full Model

Another approach to capture human activity within a model is actually to divide the training set in independent smaller data sets for which this activity should be identical, and to fit a model to each of those reduced training sets. In practice, the full data set was divided in 4*7*24 = 672 data sets, one for each different season, day and hour. Those small data sets had around 55 training examples each, and would comprise all the available data points corresponding for example to a 8am on a Monday in Winter.

$$Y_t = \theta_0 + \theta_1 Trend + \theta_2 T_t + \theta_3 T_t^2 + \theta_4 T_t^3 + \theta_5 T_{t-1} + \theta_6 T_{t-1}^2 + \theta_7 T_{t-1}^3$$

The model, used to fit each of the reduced sets and described above, used as features the trend of the overall data, and a degree 3 polynomial of the temperature at time t as well as at time t-1, taking the temperature from the most correlated weather station for each zone. This correlation can be realized during winter mornings and kept as argument as before, or the weather station with the highest correlation can be determined for each season, day and hour. Both give quite similar results. Multivariate linear regression, through the normal equations, is used to fit the θ_i's. Also, holidays were manually modified in the data to be treated as weekend days.

To test this model, a LOWOCV algorithm was also written. In practice, it would for each hour of the test week, extract the season, day and hour, in order to pick and use the appropriate model among all those we trained on the other weeks. Satisfying training and testing error of the order of 5% were obtained with that model - some results are shown on Figure 1.

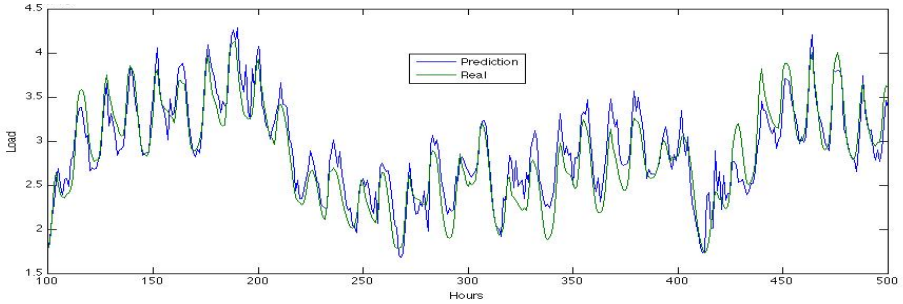

Figure 1. Example: Zone 10 Forecast in Late 2007

Bias vs. Variance

The next step was to construct learning curves to diagnose if we had rather bias or variance issues. Despite the small size of each of the training sets, the learning curve plotted here for this model firstly showed that enough data is available to make the testing error converge to the training error, and thus that we did not need to train the model upon every week: a limited number was necessary, which lead a substantial acceleration of the algorithm. It also taught us that in order to further improve our model the bias should be reduced by adding more features. Potential other features were then envisioned, such as the temperature 2 hours and 3 hours before, or also the temperature given by the second and third best correlated weather stations.

Those features were added to the model, still keeping the division for each season, hour and day. In doing so, the training error substantially dropped from 5.22% to as low as 2.72%. But concerning the testing error, there were no major improvements and it appeared that the best results were often obtained with fewer features. For some zones, it was better to use the two best correlated stations than only one, meaning that the zone could be in between two weather stations

for instance, and for others only one was more effective with this data set. Notably, adding the previous temperature was beneficial for quite a few of them.

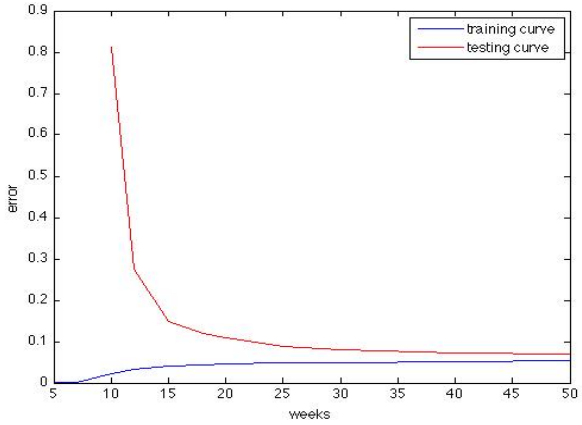

Figure 2. Learning Curve

This significant increase in variance was due to overfitting. There was not enough data for each season, day and hour to fit appropriately more features and to lower effectively the testing error (if we use them all, we have more than 40 parameters for 55 points!). One idea was then to group weekdays and weekends, increasing the training sets size but risking to loose the intrinsic difference in human activity between each of those days. When doing so, continuous improvement was noticed when adding more correlated stations. For some zones, grouping weekdays and weekends to then fit more correlated stations performed better than treating each day separately with only one station.

We do believe those supplementary features are promising and carry information, but in order to include them ideally into the model, without overfitting and without having to deal with complicated trade-offs, more data would be necessary.

On forecasting

Another possibility to improve the accuracy of this model could be also to use the load history. For example, for very short term forecasting, adding as feature the load of the previous hour drops the testing error to around 0.5%. We could also imagine, depending on how far in the future the forecast is needed, using the load at the same hour the day or the week before, etc.

An attempt to forecast iteratively a whole week showed interesting results. Using the load of the previous hour proved to be more accurate on the first days of the week but more and more inaccurate after several iterations, hour after hour. To then reach a point where it could be less accurate than a method not using this feature. Those results also varied depending on the zone and the week, being the sign of underlying complexity and instability.

In our study, forecasting was realized within the data, using the real temperature as input for the forecasting algorithm. This was used to decouple load prediction and temperature prediction, and build the most accurate possible load model. A next step would be to create also a model for the temperature. In that case, data sets going back 100 years are available, that should permit to create a satisfying temperature model with machine learning techniques. Then, both load and temperature model could be used to realize complete forecasting.

Data Adaptation 1: Energy outages

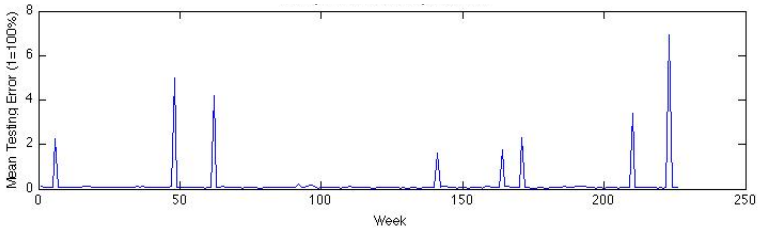

Figure 3. Outage Detection using LOWOCV

Some data adaptations were needed to improve the performance. For example, load data may have sudden drops corresponding to energy outages in the zone. These outages were detected through the LOWOCV algorithm and systematically isolated to not affect the model training. Removing outages from data using a criterion on the maximum LOWOCV testing error reduced training error from 14.72% to 5.84% in zone 4 while using the same training algorithm.

Data Adaptation 2: Load jump

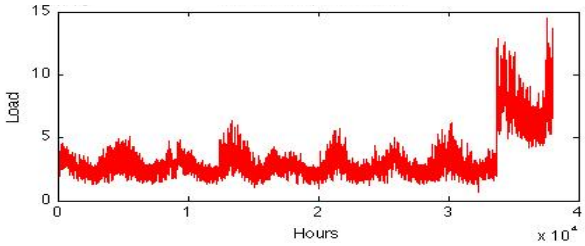

Figure 4. Load History in Zone 10

In certain zones, load jumps can occur, created by sudden and permanent load transfers or the use of a supplementary power plant. A linear shift prediction method was applied to capture such a jump occurring in 2008 in zone 10. In practice, it meant that more features were added to the design vector so that the normal equations also fit the jump. This lead to such a model:

$$Y_t = \theta_0 + \theta_1 Trend + \theta_2 T_t + \theta_3 T_t^2 + \theta_4 T_t^3 + \theta_5 T_{t-1} + \theta_6 T_{t-1}^2 + \theta_7 T_{t-1}^3 + \alpha(\theta_8 + \theta_9 T_t)$$

Here, α was for example 1 for 2008 and 0.1 for the years before, while the jump fitters θ_8 and θ_9 would only be non-null and calculated through the normal equations for zone 10 where the jump occurs. The results for the jump fitting are shown below on Figure 5. Without the alpha shift correction, the mean training error in zone 10 was 22.63%, with it, only 4.74%.

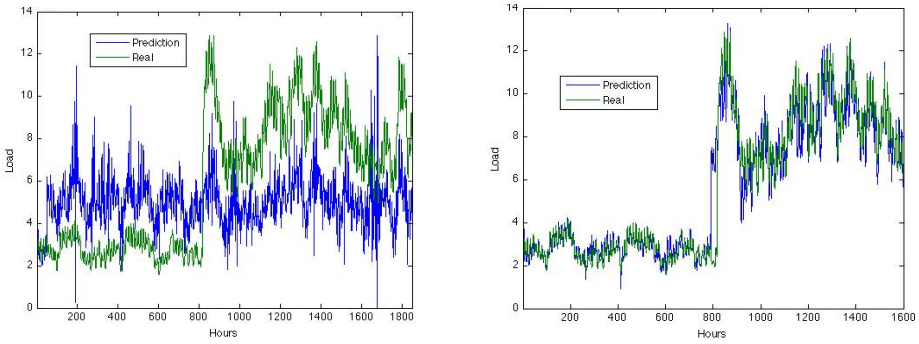

Figure 5. Zone 10 Forecast with Alpha Shift off (left) and on (right)

Results

The performance of the early indicator model and full model are listed below along with different simpler models for which one feature or adaptation was removed independently from the full model, or using only temperature, no hour dependency, etc. The full model, apart from zone 9, provides a very good prediction of energy consumption. The errors listed below are training errors, which are equal to the testing error found in doing LOWOCV since learning convergence is achieved in all those configurations.

Zone	1	2	3	4	5	6	7	8	9	10	Mean	Mean without 9
Full Model	5.81%	3.91%	3.91%	5.84%	6.35%	3.93%	3.91%	5.22%	42.60%	4.74%	7.09%	5.22%
Remove Outages	-	-	-	14.72%	-	-	-	-	-	-	7.53%	5.69%
Remove Alpha Shift	-	-	-	-	-	-	-	-	-	22.63%	7.98%	6.16%
Remove T(t-1)	6.33%	4.19%	4.19%	6.18%	6.88%	4.22%	4.19%	5.59%	49.61%	5.31%	7.86%	5.66%
Remove Trend	6.43%	4.36%	4.36%	6.10%	6.51%	4.35%	4.36%	6.17%	44.14%	4.88%	7.77%	5.85%
Remove Holidays	5.87%	4.07%	4.07%	5.93%	6.44%	4.07%	4.07%	5.27%	42.28%	4.78%	7.14%	5.29%
Indicator Model	6.70%	4.40%	4.40%	15.34%	7.10%	4.40%	4.40%	5.50%	34.90%	11.30%	7.99%	6.57%
Only T(t)	6.97%	4.65%	4.65%	16.11%	7.04%	4.65%	4.65%	6.56%	51.23%	23.54%	9.94%	7.76%
Only T(t), no hour/day dependance, no holiday...											Around 17%	

Zone	11	12	13	14	15	16	17	18	19	20	Mean	Mean without 9
Full Model	4.81%	5.22%	5.73%	7.46%	5.74%	6.74%	4.36%	5.09%	6.35%	4.07%	7.09%	5.22%
Remove Outages	-	-	-	-	-	-	-	-	-	-	7.53%	5.69%
Remove Alpha Shift	-	-	-	-	-	-	-	-	-	-	7.98%	6.16%
Remove T(t-1)	5.25%	5.74%	6.12%	8.16%	6.18%	7.32%	4.77%	5.59%	6.93%	4.41%	7.86%	5.66%
Remove Trend	6.11%	6.30%	6.13%	8.09%	6.09%	7.22%	5.41%	6.14%	7.14%	5.03%	7.77%	5.85%
Remove Holidays	4.83%	5.27%	5.76%	7.53%	5.77%	6.82%	4.39%	5.13%	6.41%	4.09%	7.14%	5.29%
Indicator Model	5.40%	5.90%	5.90%	8.10%	5.90%	7.40%	5.30%	6.00%	6.90%	4.50%	7.99%	6.57%
Only T(t)	6.57%	6.82%	6.54%	8.78%	6.54%	7.81%	5.85%	6.64%	7.74%	5.38%	9.94%	7.76%
Only T(t), no hour/day dependance, no holiday...											Around 17%	

Conclusion

An efficient learning algorithm was developed to accurately predict load consumption in 19 zones using a division of the data, a limited amount of features and some data adaptations. This model is able, when knowing the temperature, to predict the load with 5% mean error over more than 4.5 years of hourly data. Two of those data adaptation techniques, that substantially improved the training in specific zones, were described. Also, additional features were proposed, that could provide a higher accuracy over bigger sets of data. At last, further work could be to create a model to predict temperature in order to realize and test complete forecasting.

References

[1] www.kaggle.com. Global Energy Forecasting Competition 2012 – Load Forecasting
[2] Tao Hong, "Short Term Electric Load Forecasting", PhD dissertation, North Carolina State University, Sept 10th 2010