Electricity Load and Price Forecasting with MATLAB

This example demonstrates building a short term electricity load (or price) forecasting system with MATLAB. Two non-linear regression models (Neural Networks and Bagged Regression Trees) are calibrated to forecast hourly day-ahead loads given temperature forecasts, holiday information and historical loads. The models are trained on hourly data from the NEPOOL region (courtesy ISO New England) from 2004 to 2007 and tested on out-of-sample data from 2008. The models are shown to produce highly accurate day-ahead forecasts with average errors around 1-2%. The demo includes publishable reports as well as an Excel front end which enables users to call the trained load forecasting models through a MATLAB-deployed DLL.

1. Background

Accurate load forecasts are critical for short term operations and long term planning for utilities. The load forecast influences a number of decisions including which generators to commit for a given period, and broadly affects the wholesale electricity market prices. Load and price forecasting algorithms typically also feature prominently in reduced-form hybrid models for electricity price, which are some of the most accurate models for simulating markets and modeling energy derivatives. The electricity price forecast is also used widely by market participants in many trading and risk management applications.

The load forecast influences a number of decisions including which generators to commit for a given period, and broadly affects the wholesale electricity market prices. Load forecasting algorithms typically also feature prominently in hybrid models for electricity prices, some of the most accurate class of approaches for modeling electricity markets. The electricity price forecast is used widely by market participants in many trading and risk management applications.

Traditionally, utilities and marketers have used commercial software packages for performing load forecasts. The main disadvantage of these is that they are a black box, offering no transparency into how the load forecast is calculated. They also only typically offer 80-90% of the functionality needed by a utility. In many cases it is just not possible to meet all of the requirements through an off-the-shelf product, for instance taking into account regional loads, different weather patterns and so on.

MathWorks tools provide the flexibility of building a completely customized load forecasting system that meets 100% of the requirements. And because of the built-in models, high-level language and ease of connecting to data, the time taken to develop such a system is also dramatically lower than building an equivalent system in a lower level programming language, as is demonstrated in this example.

2. Data

The data used for this example are historical hourly temperatures, system loads and day-ahead electricity prices from the New England Pool region.

IMPORTANT: **The Access database shown in the webinar is not provided with this archive due to size restrictions. The equivalent data sets are provided in the folders *Load\Data* and *Price\Data* for the load and price forecasting studies respectively.**

The original data were obtained from the New England ISO at http://www.iso-ne.com/. At the time of writing this was the direct link to Zonal hourly prices and load data. The script "*importData.m*" in the main folder can be used to read these spreadsheets into MATLAB. The Natural Gas spot price data (required for the price forecasting piece) can be obtained from the Wall Street Journal at http://www.wsj.com.

3. Setup

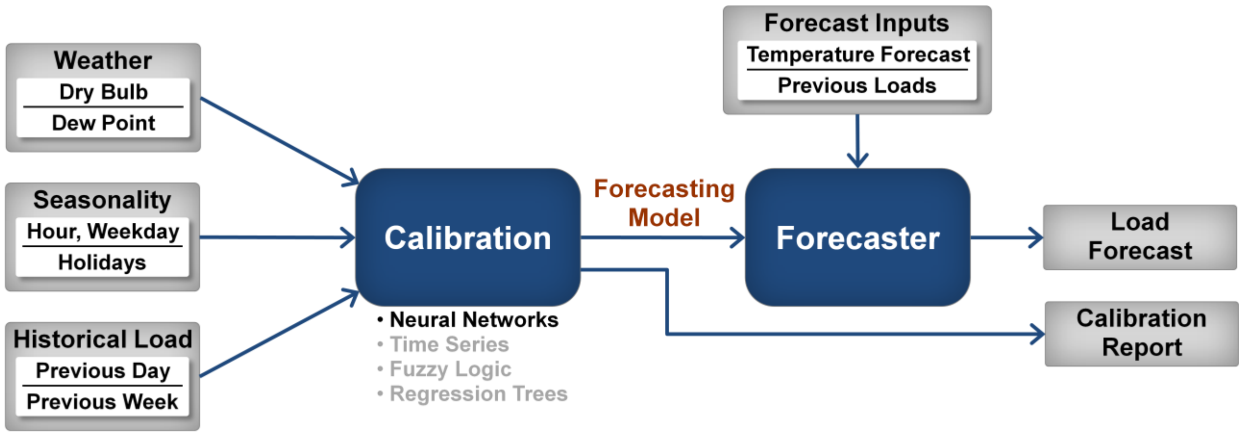
The only setup that is required is adding the *Util* folder to the MATLAB path. This can be done through MATLAB preferences. The scripts *LoadScriptNN, LoadScriptTrees* and *PriceScriptNN* do this in the first section of the code.

4. Importing Data

The MAT files *Load\Data\ DBLoadData.mat* **and** *Price\Data\ DBPriceData.mat* contain data equivalent to that imported from the Access database in the webinar recording. This dataset is sufficient for all of the analyses in the example.

5. Building the Forecaster

The three steps to building the forecaster include creating a matrix of predictors from the historical data, selecting and calibrating the chosen model and then running the model live in the Excel interface. The Load Forecasting example demonstrates using both Neural Networks as well as Bagged Regression Trees to forecast load.



5.1 Generating the Predictor Matrix

The predictors for non-linear regression are generated by the function *genPredictors.* This function creates a matrix of inputs to the model. For the load forecast, the predictors include

- Dry bulb temperature
- Dew point temperature
- Hour of day
- Day of the week
- Holiday/weekend indicator (0 or 1)
- Previous 24-hr average load
- 24-hr lagged load
- 168-hr (previous week) lagged load

The predictors for the price forecast also include (spot and lagged) natural gas prices and lagged electricity prices. Though not shown in this example, you could also extend this list to include outage information.

5.2 Calibrating the Model

The scripts *LoadScriptNN.m* and *LoadScriptTrees.m* demonstrate how the Neural Network and Bagged Regression Trees models, respectively, can be trained and validated. The models are trained on data from 2004 to 2007 and tested on completely out-of-sample data from 2008. The model accuracy on out-of-sample periods is computed with the Mean Absolute Error (MAE) and Mean Absolute Percent Error (MAPE) metrics. Various charts of the error distribution as a function of hour of the day, day of the week and month of the year are also generated.

The Neural Network model can also be trained interactively using the Neural Fitting Tool (*nftool*).

All of the load and price forecasting scripts can be published to generate HTML reports of the results. The following table contains links to these published reports.

Script	Description	Required Toolboxes	Optional Toolboxes	Launch Reports
LoadScriptNN.m	Load Forecasting with Neural Networks	Neural Networks	Statistics	Report
LoadScriptTrees.m	Load Forecasting with Regression Trees	Statistics		Report
PriceScriptNN.m	Price Forecasting with Neural Networks	Neural Networks	Statistics	Report
comparePredictors.m	Comparison of different predictors models for the price forecast	Neural Networks		Report

5.3 Building and Running the Excel Load Forecasting Application

Products Required: MATLAB Compiler, MATLAB Builder EX

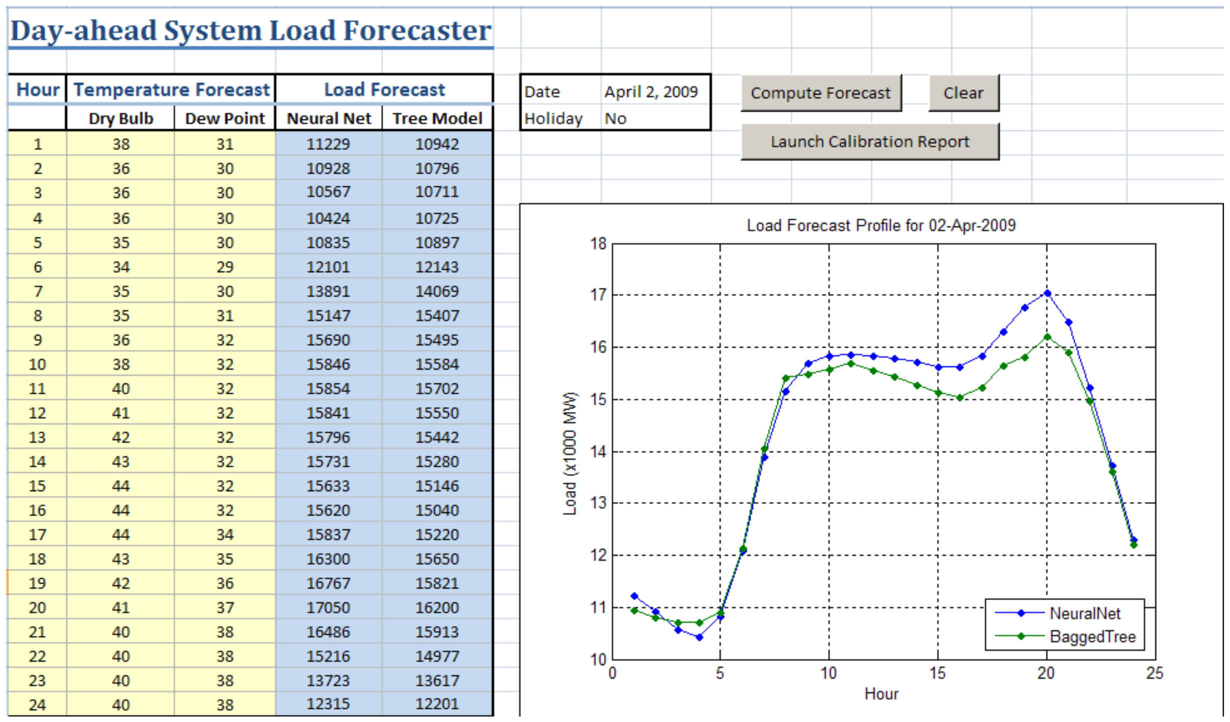
The application includes a spreadsheet ***Forecaster.xlsx*** which contains the Excel front end for the load forecasting application. The VBA code in *vbamodule.bas* as well as the VBA code generated by MATLAB Builder EX can be imported into this file to make it a fully functional deployed application as shown in the webinar. The deployment project *Forecaster.prj* includes the relevant files needed for building the deployed application. This can be opened in MATLAB and built to create the DLL needed by the Excel application.

Building the deployed component:

Ensure you have a Visual Studios or other compiler compatible with MATLAB Builder EX. Make sure the compiler is setup with the function *mbuild*. Open and build the project file *Forecaster.prj* in MATLAB. This will create an Excel COM DLL as well as a VBA .bas file. Open ***Forecaster.xlsx*** and navigate to the Visual Basic editor in the developer tab. Import the BAS files *Forecaster\distrib\Forecaster.bas* as well as *vbamodule.bas*. Save the spreadsheet with an .XLSM extension. You should then be able to run the Excel interface.

Using the interface:

Clicking the **Compute Forecast** button calls the *loadForecast* function (in the MATLAB-generated DLL). This function takes the forecast date, temperature and holiday information, pulls in historical electricity loads prior to that date, and computes a day ahead forecast using both the Neural Network and Bagged Regression Tree models. The forecast from both models is displayed both as raw data as well as a MATLAB-generated graphic.


Day-ahead System Load Forecaster
Compute Forecast
Clear
Launch Calibration Report
Load Forecast Profile for 02-Apr-2009
Load (x1000 MW)
Hour
NeuralNet
BaggedTree

Date	April 2, 2009
Holiday	No

Hour	Temperature Forecast		Load Forecast	
	Dry Bulb	Dew Point	Neural Net	Tree Model
1	38	31	11229	10942
2	36	30	10928	10796
3	36	30	10567	10711
4	36	30	10424	10725
5	35	30	10835	10897
6	34	29	12101	12143
7	35	30	13891	14069
8	35	31	15147	15407
9	36	32	15690	15495
10	38	32	15846	15584
11	40	32	15854	15702
12	41	32	15841	15550
13	42	32	15796	15442
14	43	32	15731	15280
15	44	32	15633	15146
16	44	32	15620	15040
17	44	34	15837	15220
18	43	35	16300	15650
19	42	36	16767	15821
20	41	37	17050	16200
21	40	38	16486	15913
22	40	38	15216	14977
23	40	38	13723	13617
24	40	38	12315	12201

Click “Launch Calibration Report” to launch the HTML published report for the Neural Network training and testing script.

6. Running the Price Forecasting Example

Run the script *PriceScriptNN.m* in the *Price* folder. The script is very similar to the previously mentioned script for load forecasting.

Appendix A: Table of Files in Archive

File	Description
Main Analysis Scripts	
LoadScriptNN.m	Calibrate and Test Neural Network Load Forecaster
LoadScriptTrees.m	Calibrate and Test Bagged Regression Tree Load Forecaster
PriceScriptNN.m	Calibrate and Test Neural Network Price Forecaster
TreesInDetail.m	Refine Bagged Regression Tree Model
comparePredictors.m	Compare Price Forecasting Accuracy For Different Sets of Predictors
Helper Functions	
fitPlot.m	Create plot of series, model-predicted series and residuals
dynamicDateTicks.m	Apply date ticks to plots that update with zooming and panning.
importData.m	Optional script to read in zonal hourly load and price Excel spreadsheets from ISO New England.
genPredictors.m	Generate a matrix of predictors (temperatures, fuel prices and day/week/holiday) for each electricity price or load observation
fetchDBPriceData.m	Import temperatures and electricity loads and prices from Energy database
fetchDBLoadData.m	Import temperatures and electricity loads from Energy database
filterDates.m	Helper function for createHolidayDates
createHolidayDates.m	Generate a list of holiday dates between a start and end date.
Forecaster Excel Application Functions	
loadForecast.m	Generate live load forecast using historical loads, holiday and temperature forecast information.
Other Files	
Forecaster.prj	Deployment project file that can be used with MATLAB Builder EX to build the DLL required for the Excel application
vbamodule.bas	VBA code included in the archive to complete the Excel application