Electricity Load and Price Forecasting with MATLAB

This example demonstrates building a short term load forecast system with MATLAB for Sydney, Australia. It also touched on how to easily convert this into a Price forecasting model and the challenges faced in the Australia market. Accurate load forecasts are critical for distribution 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 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.

# Data

The data used for this example are historical hourly temperatures, system loads and day-ahead electricity prices from the AEMO on NSW and Temperature data of Sydney Observatory from the BOM. We also will be using data from New England Pool region in the USA. These are provided in both Excel spreadsheets, MAT files and an access database in the *Data* folder

# Models

The historical data is used to calibrate both a Neural Network and a Bagged Regression Tree model for day-ahead load forecasting. 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
* Wet bulb temperature
* Humidity
* Dew point temperature
* Hour of day
* Day of the week
* Holiday/weekend indicator
* Previous 24-hr average load
* 24-hr lagged load
* 168-hr (previous week) lagged load

The price predictors also include (spot and lagged) natural gas prices and lagged electricity prices. The model is trained on data from 2004 to 2007 and tested on completely out-of-sample data from 2008.

# Outputs

The load forecasting example includes a deployed Excel application that uses a MATLAB-generated DLL function and the trained models to perform day-ahead load forecasting using both Neural Network and Regression Tree models. Both the load and price forecast examples also generate a report that shows the training procedure and validation on out-of-sample data with weekly accuracy plots for the 2008 test period.

# Setting up the Demo

* If using the database (recommended), set it up as an ODBC source in Windows. The access file is available in the *Data* folder.
* Add the *Util* folder to the MATLAB path. This contains common functionality for both forecasting examples
* If using the deployed Excel application (*forecastWorksheet.xlsm*), recompile the application with deployment project *Forecaster.prj* or register the DLL in *Load\Forecaster\distrib* on your Windows machine. This can be done with *$MATLABROOT\runtime\win32\mwregsvr.exe*

# Running the Load Forecasting Example

The main script L*oadScriptNN.m* will walk you through the analysis*.* However, a lot of the analysis can be done interactively at the command line. You can,

* Import data from the database using query builder. The *Date, Hour, DryBulb, DewPnt* and *SYSLoad* fields from the *NEData* table are required for the load forecasting system. These should be imported as a structure.
* Generate the historical predictors matrix with function *genPredictors*
* Build a Neural Network using the Neural Network Fitting Tool (*NFTOOL*). The inputs to the network are the predictors and the output is the historical system load in field *SYSLoad*. All other parameters can be set to default values. Generate MATLAB code from *NFTOOL* and export the network to workspace
* Run the Neural Network on the historical predictor matrix to compute the forecast load and compare it to the actual load.

This is a good point to switch to the script and step through the more formal analysis. The script divides the historical data into a training set and a test set and computes performance metrics on the out-of-sample test set. The script can be published to create an HTML report.

# Running the Excel Load Forecasting Tool

Make sure the DLL in *Load\Forecaster\distrib* has been registered on the machine and then open *forecastWorksheet.xls.* This worksheet uses historical data up to 3/31/2010 and the temperature forecast for 4/1/2010 to compute the day-ahead load forecast for 4/1/2010. To run the forecaster, click the “Compute Forecast” button. This generates the day ahead forecast using both the Neural Network and Bagged Regression Trees models and creates a plot of the forecast load profile. The function can also be called from a cell. Cell H1 has a prototype call for this function. To use it, select cells E3:E36, paste in the formula (without the leading ‘) and press Ctrl+Shift+Enter.

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

# 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.