Commercial Building Load Prediction using Modified Learning from Experience & Recursive Least Squares Methods

Improving Baseline Calculation for Building Energy Management

8/1/2011
University of California, Berkeley
Tyler C. Jones
David M. Auslander

DRAFT

Modified Learning from Experience Algorithm with Recursive Least Squares algorithm

The central aim of this paper is to develop and compare two methods for calculating a central load baseline for a commercial building, with the intention of using these methods for Automated building management during a Demand Response event. Two methods of calculating a central load baseline were developed. The first method, LBNL's Best uses a simple average of 3 of the hottest days of the previous 10 days with a correction factor to account for the variation in the morning-of building behavior. The second method implements a Modified Learning from Experience (MLFE) algorithm with a Recursive Least Squares (RLS) best fit algorithm for increased accuracy of load prediction. The method was compared to the previous method developed for the project and achieved better results for a given test set of days available in the sMap database. The purpose for this improvement was the minimization of plant uncertainty for the building control system. It is crucial that there is not a significant amount of plant mismatch in proper control of the building power, thus increasing stability of the control system for the given demand response time period.

Theory

The algorithm uses inputs for a given day and produces a single output value for a baseline prediction at each time step, using what is known as Fuzzy Set Theory. Fuzzy set theory is based on the fact that uncertainty is nearly always present in real life systems. Since its introduction by Lotfi Zadeh in 1965, it has brought about a shift from the logic of probability theory, which is based on classical binary (two-valued) logic to continuous-valued logic. Fuzzy logic involves a mapping between elements of two or more domains. Just as an algebraic function maps an input variable to an output variable, a fuzzy system maps an input group to an output group.

Uncertainty has been viewed over the centuries as incompleteness, imprecision, and complex. Uncertainty can be manifested in many forms: it can be fuzzy (not sharp, unclear, imprecise, approximate), it can be vague (not specific, amorphous), it can be ambiguous (too many choices, contradictory), it can be of the form of ignorance (dissonant, not knowing something), or it can be a form due to natural variability (conflicting, random, chaotic, unpredictable).

Fuzzy logic can be used extensively in many applications related to commercial building management systems and the grid. Fuzzy logic has the ability to predict system behavior not requiring complete accuracy but a relatively high degree of accuracy, usually set by the user with the advantage of significantly shorter calculation time. It also is very useful in predicting, within a reasonable tolerance, the behavior of systems with highly nonlinear characteristics. Systems using fuzzy logic also have a high potential to understand complex systems, devoid of analytic formulations or systems in which the causes and effects are not generally understood but can be observed.

Calculations

The new method requires a set of training data to generate membership functions that require a set of input that have been "fuzzified" and return a single delta output, in our case real power for the entire building. Data for the hottest ten days over the past year was gathered and used to train the algorithm. A set of input test data for three days were plugged in and tested for accuracy. The algorithm is run for each individual time step, in the preliminary testing phase, every hour, but eventually can be implemented for every 15 minute interval.

The given set of inputs used for the training MLFE algorithm includes previous day power data for the building, morning-of power data for the building, and current real-time weather data. The inputs used for the prediction algorithm are the following:

Previous days power data:

X_1: 3 hottest days of previous 10 days average power data for each time step (**PL**(d,h))

Morning-of data:
X_2: Actual load at 10 AM day-of, **AL**(d,10)
X_3: Actual load at 11 AM day-of, **AL**(d,11)

Real-time weather data:
X_4: Relative humidity at each time step , **RH**(d,h)
X_5: Outside air temperature at each time step, **OAT**(d,h)

Output:
Y: Power at each time step, **AL**(d,h)

The data above is first normalized using the maximum values for each category seen in the entire data set. Once normalized, Gaussian membership functions have been used for input output functions, although any membership function could have been used. The output membership function is a delta function, an impulse function of no width, that occurs at a value b_i with full membership.
Rules are developed by the algorithm in the following manner for our system of multiple inputs and a single output:

IF X_1 and X_2 and X_3 and X_4 and X_5 THEN Y

Gaussian Membership functions take the form of Equation(1).

$$\mu = exp\left[-\frac{1}{2}\left(\frac{x_i - c_i}{\sigma_i}\right)^2\right] \qquad (1)$$

x_i : ith input variable

c_i : ith center of the membership function (where membership achieves maximum value)

σ_i : spread of ith membership function (constant)

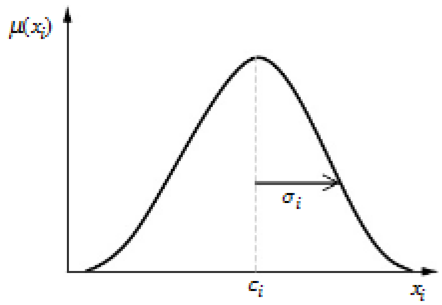


Figure 1: Typical Gaussian membership function

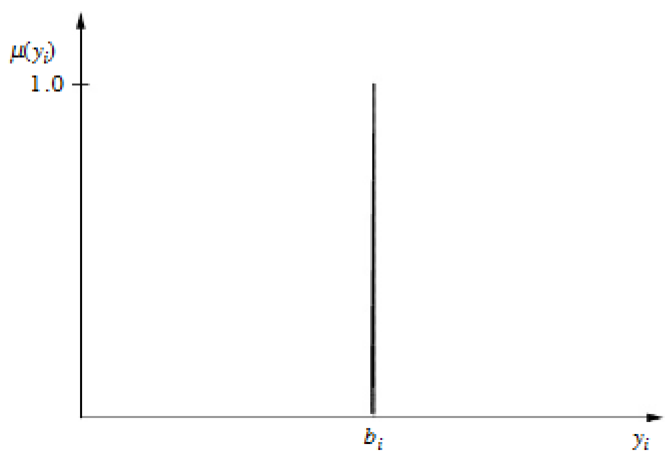


Figure 2: Delta membership function

First, a Modified Learning from Experience (MLFE) algorithm must be used to generate a rule-base, since we have no knowledge of the characteristic behavior from the data sets. The rule-base consists of the number of rules and the rule parameters

The process is initiated by setting the number of rules, R = 1, and for b_1 , $c_1^1, c_2^1, c_3^1, c_4^1, c_5^1$ we use the first day training data-tuple in Z. X_1 is set to be c_1^1 , X_2 is set to be c_2^1 and so on, and b_1 is set to y^1. In the algorithm, b_i is the point in the output space at which the output membership function for the ith rule is a delta function, and c_j^i is the point in the jth input universe of discourse where the membership function for the ith rule achieves a maximum. The relative width, σ_j^i , of the jth input membership function for the ith rule is always greater than zero. It is important to note that the spreads can never be set to zero, to avoid a division by zero error later in the algorithm. The spread will be assumed to be equal to 0.06.

For this example we would like the fuzzy system to approximate the output to within a tolerance of 0.05, thus we set ε_f= 0.04. We also introduce a weighting factor, ω, which is used to calculate the spreads for the membership functions, as given later in Equation (3). The weighting factor is used to determine the amount of overlap between the membership function of the new rule and that of its nearest neighbor. For this project, the value of the initial weighting factor, ω is set to 0.08.

The output of the training data set is then compared to the real power data from the training sets to see how well the fuzzy system is mapping the information. The required stopping condition for the algorithm is shown below. The difference must be smaller than the user set tolerance in order for no additional rules to be added.

$$\left|f\left(x^i\middle|\theta\right) - y^i\right| < \varepsilon_f \quad (2)$$

If the tolerance is exceeded, a rule is added to the rule-base to represent (x2,y2) by modifying the current parameters θ. R is set to 2, b_2= y^2, and $c_j^2 = x_j^2$.

If an additional rule is needed, the centers for the new rule are set to the next training data inputs, x_j^i , and the relative widths are determined by the MLFE algorithm based on achieving an appropriate overlap between membership functions. This overlap is set by a user defined weighting factor (ω).

$$\sigma_j^i = \frac{1}{\omega}\left|c_j^{i'} - c_j^{min}\right| \quad (3)$$

where:

$c_j^{i'}$: the current input training data set, x_j^i

c_j^{min}: the nearest membership function centers to the new membership function centers $c_j^{i'}$
ω: the user defined Gaussian membership function width weighting factor

This process of adding additional rules using the next training data set repeats until Equation (2) is satisfied.

Recursive least squares

After the rule-base has been generated with the MLFE algorithm, the RLS algorithm calculates $\hat{\theta}(k)$ at each time step k from the past estimate $\hat{\theta}(k-1)$ and the latest data pair that is received, x^k & y^k.

Recall that b_i is the point in the output space at which the output membership function for the ith rule is a delta function, and c_j^i is the point in the jth input universe of discourse where the membership function for the ith rule achieves a maximum. The relative width, σ_j^i , of the jth input membership function for the ith rule is always greater than zero.

Now we calculate the regression vector, ξ based on the training data using Eq. (4)

$$\xi_i(x) = \frac{\mu_i(x)\ \prod_{j=1}^{n} exp\left[-\frac{1}{2}\left(\frac{x_j - c_j^i}{\sigma_j^i}\right)^2\right]}{\sum_{i=1}^{R}\prod_{j=1}^{n} exp\left[-\frac{1}{2}\left(\frac{x_j - c_j^i}{\sigma_j^i}\right)^2\right]} \qquad (4)$$

Recall that in the least squares algorithm the training data x_i are mapped into $\xi(x_i)$ which is then used to develop an output $f(x_i)$ for the model.

A covariance matrix is used to determine the least squares estimate vector of the training set, $\hat{\theta}$, which is calculated using the regression vector and a previous covariant using Equation (6). To do this, an initial covariance matrix, P_0 must first be calculated using a parameter, α and the identity matrix, I. P_0 is used to update the covariance matrix, P, in the next time step. A recursive relation is established to calculate values of the P matrix for each time step using Equation (5). The value of the parameter α should be greater than 0. Here a value of 100 is used for α . I is an R x R identity matrix.

$$P(0) = \alpha I$$

$$P(k) = \frac{1}{\lambda}\left\{I - P(k-1)\xi(x^k)\left[\lambda I + \left(\xi(x^k)\right)^T P(k-1)\xi(x^k)\right]^{-1}\left(\xi(x^k)\right)^T\right\}P(k-1) \qquad (5)$$

$$\hat{\theta}(k) = \hat{\theta}(k-1) + P(k)\xi(x^k)\left[y^k - \left(\xi(x^k)\right)^T\hat{\theta}(k-1)\right] \qquad (6)$$

$$f(x|\theta) = \hat{\theta}^T\xi(x)$$

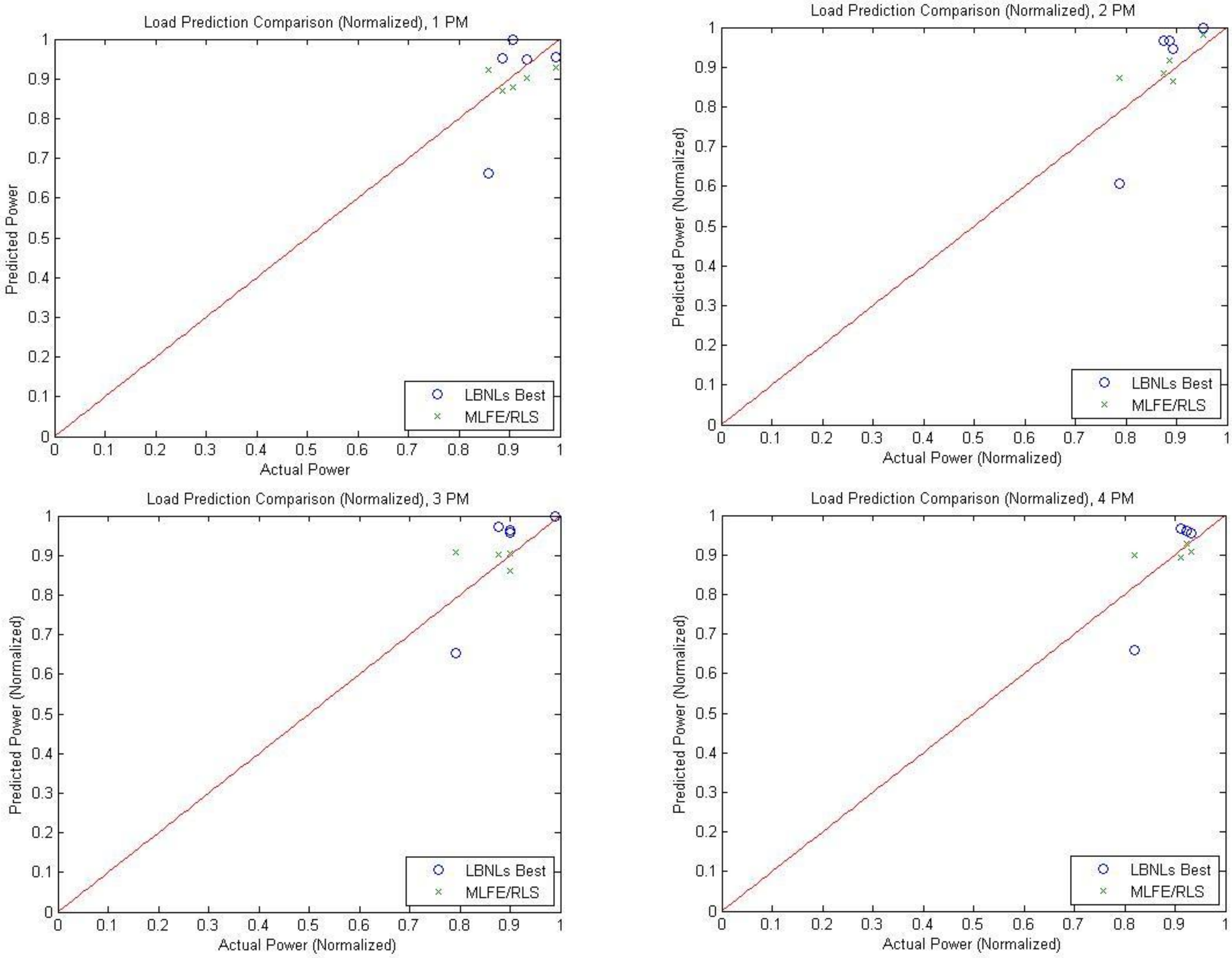
Results

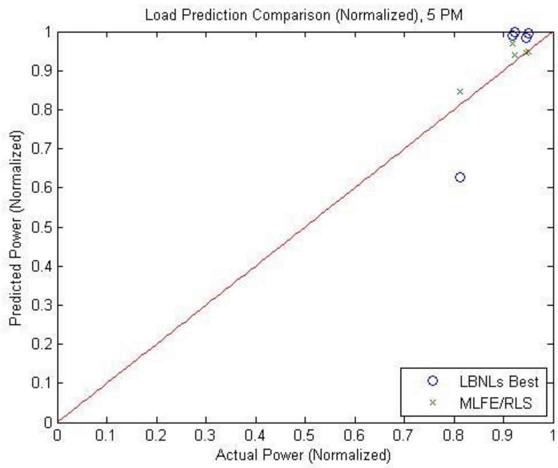
Using the MLFE/RLS algorithm, predictions were made for the afternoon hours from 1 PM to 5 PM in 1 hour intervals. The predicted values were compared against the previous load baseline prediction algorithm. The predicted power data was measured against real building power data for 5 test days and error was calculated as percentage error.

Error Results (in Percentage)

	Test Day 1 May 3, 2011		Test Day 2 Nov 11, 2010		Test Day 3 Oct 28, 2010		Test Day 4 July 6, 2011		Test Day 5 June 30, 2011		Average (RMS error)	
Time	**Old Method**	**MLFE/ RLS**	**Old Method**	**MLFE/ RLS**	**Old Method**	**MLFE/ RLS**	**Old Method**	**MLFE /RLS**	**Old Meth od**	**MLFE/ RLS**	**Old Method**	**MLFE/ RLS**
1:00 PM	**2.52**	**1.03**	**-7.95**	**-5.26**	**-26.45**	**6.00**	**-2.98**	**-3.70**	**5.45**	**-0.87**	**12.71**	**3.98**
2:00 PM	**-0.43**	**3.31**	**-4.00**	**2.99**	**-29.28**	**11.00**	**-3.06**	**-3.43**	**1.30**	**1.04**	**13.30**	**5.55**
3:00 PM	**2.74**	**2.75**	**-6.32**	**2.58**	**-23.46**	**14.85**	**-1.20**	**0.72**	**-0.67**	**-4.27**	**10.95**	**7.12**
4:00 PM	**0.26**	**-0.70**	**4.88**	**-5.83**	**-22.57**	**9.64**	**2.00**	**-2.17**	**-1.54**	**-2.61**	**10.39**	**5.27**
5:00 PM	**-1.31**	**5.45**	**-1.81**	**1.81**	**-27.52**	**3.90**	**-1.51**	**-0.23**	**-2.29**	**-0.07**	**12.41**	**3.11**

Prediction Plots for each Test Day





After an inspection of the results, the MLFE/RLS algorithm seems to offer robustness from unusual phenomenon. Examination of the third test day, October 28, it was found that the building had unusual power behavior on the morning of the test day most likely due to testing in the building or a conference that may have provided unusual morning behavior for the building. Power use at the 10 AM and 11 AM hours was seen as not typical for a non-holiday weekday. It can be seen that the MLFE/RLS algorithm provides a little more robustness to unexpected occurrences in input behavior or disturbances. The errors were much smaller for the particular day. The remaining test days, show a slight advantage for the MLFE/RLS Method over the LBNL's Best Method. In most cases the MLFE/RLS algorithm does better than the LBNL's Best Method. In a few cases the opposite is found to be true. The RMS error in all cases was significantly smaller using the MLFE/RLS Method. A graphical representation comparing the two methods also confirms the accuracy of the two models and clearly shows an advantage using the MLFE/RLS Method. As more test data becomes available, a further examination can be done on accuracy of the two methods.

References

[1] Pardo, A., Meneu, V. and Valor, E. Temperature and seasonality influences on Spanish electricity load, *Energy Economics*, Vol.24, No.1, Jan 2002, pp 55-70.

[2] H.L.Willis, "Distribution load forecasting", IEEE Tutorial course on power distribution planning, EHO 361-6-PWR, 1992.

[3] J.V. Ringwood "Intelligent Forecasting of Electricity Demand".

[4] S. Sachdeva and C. M. Verma, "Load forecasting using fuzzy methods," in *Proceedings of the 2008 IEEE Power System*

Technology and Power India Conference, India, October 2008.

[5] Coughlin, Katie, Piette, Mary Ann, Goldman, Charles, & Kiliccote, Sila. "Estimating Demand Response Load Impacts: Evaluation of Baseline Load Models for Non-Residential Buildings in California." Lawrence Berkeley National Laboratory,LBNL-63728. January 2008.

[6] Federal Energy Regulatory Commission (FERC) Staff Report. "Assessment of Demand Response and Advanced Metering." August 2006.

[7] Federal Energy Regulatory Commission (FERC) Staff Report. "Assessment of Demand Response and Advanced Metering." September 2007.

[8] Coughlin, Katie, Piette, Mary Ann, Goldman, Charles, & Kiliccote, Sila. "Estimating Demand Response Load Impacts: Evaluation of Baseline Load Models for Non-Residential Buildings in California." Lawrence Berkeley National Laboratory, LBNL-63728. January 2008.

[9] Ross, T. J., Fuzzy Logic with Engineering Applications, second edition, John Wiley and Sons, Ltd, West Sussex, England, 2004.

[10] Mendel JM. Fuzzy logic systems for engineering: a tutorial. Proc IEEE 1995;83(3):345–77.