

Electrical Energy Modeling In Y2E2 Building Based On Distributed Sensors Information

Mahmoud Saadat

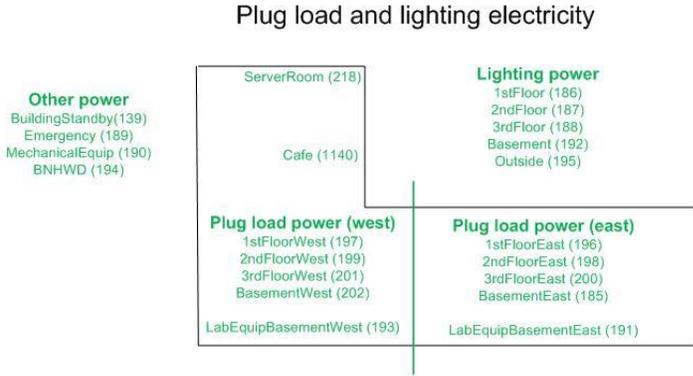
Saman Ghili

1 Introduction

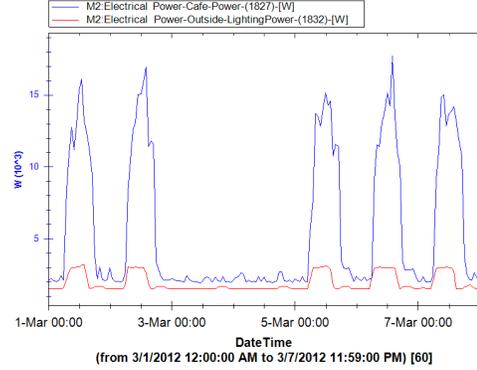
Close to 40% of the primary energy consumption in the U.S. comes from commercial and residential buildings [1]. Therefore, reducing this energy consumption is very important, both economically, and environmentally (due to the amount of CO₂ emitted in the process of generating the electricity). Predicting the energy consumption of buildings (energy modeling) is a key component in reducing the building energy consumption. Two traditional types of energy modeling are forward modeling (using purely physical simulations) and inverse modeling (using statistical methods along with expert knowledge, to relate the energy consumption to a set of general inputs such as the outdoor temperature, etc.) [2]. If we have access to sensor data (both inside and outside the building) however, a better alternative is sensor based energy modeling using statistical machine learning techniques. [2] uses seven different machine learning algorithms to predict the total hourly energy consumption of three residential buildings (called Campbell Creek Homes) designed to evaluate the effectiveness of residential construction and efficiency technologies in a controlled environment. In this project, we apply three of these algorithms (Simple Linear Regression, Support Vector Regression or SVR, and Least Squares Support Vector Machines or LS-SVM) to the energy consumption and sensor data from the Y2E2 building on Stanford campus. This rich data-set allows us to model different parts of the total consumption (i.g. lighting, AC, plug loads, etc.) separately, which could lead to more accurate results compared to the case where we are modeling the total consumption as a whole.

2 The Y2E2 data

The Yang and Yamazaki Environment & Energy Building (Y2E2), is a good case study for energy performance modeling. A number of electrical power sensors are installed throughout the building. The power sensors measure lighting, plug load, and overall power at different floors and portions of the building. As an example, Figure 1a below shows that light is measured on every floor, plug loads are measured on west and east portions of each floor separately and a number of other power sensors such as the server room, cafe, etc. are also available. The sensors data is collected every minute in SEE-IT database and can be accessed, plotted, and exported using SEE-IT application. An example of a plot generated by SEE-IT is shown in Figure 1b. In this plot, the electrical power of the caf on the first floor and outside lighting are illustrated in a one-week period from 3/1/2013 to 3/8/2012.



(a) The location of sensors.



(b) Sample power consumption time series plotted by SEE-IT.

3 Algorithms, attributes, and features

3.1 Algorithms

Let \mathbf{x} be a vector of attributes, $\phi(\mathbf{x})$ be a feature mapping, and $\{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^m$ be our training set. We are looking for a hypothesis in the form of $h_{\mathbf{w},b}(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) + b$. In linear regression, \mathbf{w} and b are minimizers of $\sum_{i=1}^m (h_{\mathbf{w},b}(\mathbf{x}^{(i)}) - y^{(i)})^2$. In SVR, \mathbf{w} and b are obtained from the following quadratic program [3]:

$$\underset{\mathbf{w},b}{\text{minimize}} \quad \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^m (\xi_i + \xi_i^*) \quad (3.1)$$

$$y^{(i)} - \mathbf{w}^T \phi(\mathbf{x}^{(i)}) - b \leq \epsilon + \xi_i, \quad 1 \leq i \leq m$$

$$\mathbf{w}^T \phi(\mathbf{x}^{(i)}) + b - y^{(i)} \leq \epsilon + \xi_i^*, \quad 1 \leq i \leq m$$

and in LS-SVM we have to solve the following optimization problem [4]:

$$\underset{\mathbf{w},b}{\text{minimize}} \quad \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^m \xi_i^2 \quad (3.2)$$

$$\mathbf{w}^T \phi(\mathbf{x}^{(i)}) + b + \xi_i = y^{(i)}, \quad 1 \leq i \leq m$$

where C and ϵ are tuning parameters, and ξ_i and ξ_i^* are slack variables.

3.2 Attributes

First we note that time (measured in hours) should be included in the model for any part of the total consumption. If everything else was fixed, one would expect ϕ to be a periodic function of time (period=24h). Therefore we include $\sin(t \frac{2\pi}{24})$ and $\cos(t \frac{2\pi}{24})$ in our attributes. Besides time, we could include different attributes when modeling each data-set. In modeling the lighting power, we add the illumination L (measured in Lux). In modeling the power consumed by the AC system however, we add two other attributes, the outside dry bulb temperature T , and an binary variable s indicating whether the main window is open or closed on each floor. Also note that some of these attributes are different on different floors, which means that some of the modelings should be done separately for each floor. When modeling the total power

consumption of the building, we include all of these attributes. In order to distinguish between weekends or holidays and weekdays, we can add a binary attribute d that is 1 on the weekdays and 0 otherwise, or we can model them separately.

3.3 Features

If we use the kernel trick for training in LS-SVM and SVR, we will not need to explicitly write down the features. However, in LS, or when we are using a linear kernel in LS-SVM and SVR, we need to state the features explicitly. Here we briefly explain what features we are going to use for those cases.

For features that are functions of time, because of the periodicity, we use $2n_t$ features $\{\sin(kt\frac{2\pi}{24}), \cos(kt\frac{2\pi}{24})\}_{k=1}^{n_t}$, where n_t will be determined using a feature selection scheme. For the outside temperature and illumination, we use polynomials of degrees n_T and n_L respectively. Although the degree of each of these polynomials can also be determined using a feature selection scheme, they are easier to tune intuitively. For example, it makes sense to use a polynomial of degree 3 in the outside temperature. The reason is that a cubic polynomial can capture the facts that i) the needed power is positive if the outside temperature is either below or above the set-point temperature, and ii) the power needed for cooling is different from that of heating.

Finally, the interaction of these features can also be used. Specifically, we should include the interaction of the holiday and weekend indicator d with all the other parameters. This is equivalent to modeling these days and the weekdays separately.

4 Implementation and results

We train the LS and LS-SVM for the power consumption data gathered every over a 9 months (from February 1st to October 1st, 2013) once every 15 minutes ¹, and assess the accuracy of the trained model by looking at the Coefficient of Variance (CV) for a test data-set of power consumption collected over an (almost) three-week period (from October 4th to October 23rd, 2013, one data point per hour). The Coefficient of Variance is defined as follows:

$$CV = \frac{\frac{1}{N-1} \sum_{i=1}^N (y^{(i)} - \hat{y}^{(i)})^2}{\bar{y}} \times 100 \quad (4.1)$$

where N is the size of the test set, $y^{(i)}$ s are the actual consumption values, and $\hat{y}^{(i)}$ s are the ones predicted by the model.

Before training, a simple preprocessing was performed by scaling all the attributes, so that they all are of the same order. We then trained different models to the following data-sets: AC power for each of the three floors, the lighting power of the building, the plug load for the whole building, and the total power consumption. We trained Linear Least Squares, and LS-SVM with three different Kernels: linear, polynomial, and Gaussian. The tuning parameters were found by a 4-fold cross validation and grid-search as suggested in [5]. We implemented both methods in MATLAB. We also tried implementing SVR. However the slow running time of the optimization problem for SVR due to inequality constraints (even when using LibSVM), together with the facts that there is one more regularization parameter (ϵ) and we use a brute-force grid search method for picking the cross-validation parameters made the feasible size of

¹The Y2E2 sensor data is gathered every minute, however using all the data would hugely increase the training size without adding much information due to the auto-correlation in the time series data.

the data set on which we could train the model very small, and we decided to include the results only for LS and LS-SVM.

We used forward feature selection in choosing n_t , n_L , and n_T . In all cases, n_t was either 2 or 3, n_L was 1 or 2, and n_T was 2 or 3. When using a Gaussian kernel $K_G(x, z) = \exp\left(-\frac{\|x-z\|_2^2}{2\sigma^2}\right)$, the parameter σ was included in the cross-validation with grid-search. For a polynomial kernel $K_P(x, z) = (x^T z + c_P)^{D_P}$, c_P was found in the grid search, while D_P was incremented until the validation error started to increase (similar to forward selection).

As a final note before we present the results, we note that in building energy modeling, the CV errors are usually larger compared to modeling the energy consumption of a city, a state, or a country. The reason is that, roughly speaking, the power consumption time series in a building is noisier than that of the whole city or state. For example, the CV errors obtained in [2] are of the order of %10, and sometimes as high as %40 for almost all the methods tried.

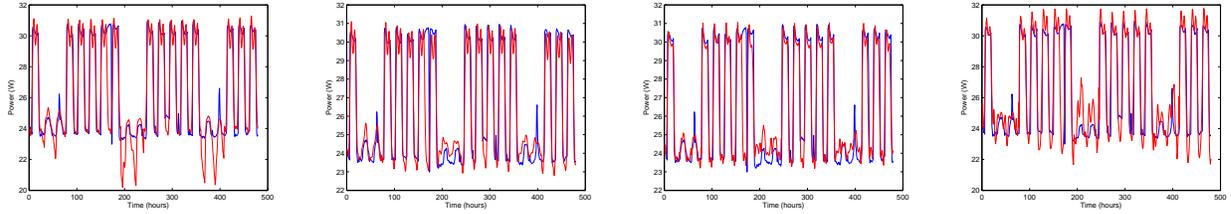
Table 1 summarizes the CV errors (in percent, rounded to the nearest integer) for the application of different methods to different data sets. Although we have modeled the AC power for different floors separately (because the window parameter is different for them), we present the CV error for the sum of these powers here.

	Total Consumption	Plug Loads	Lighting	AC Power
LS	19%	15%	28%	7%
LS-SVM - Linear Kernel	19%	14%	28%	5%
LS-SVM - Polynomial Kernel	24%	20%	33%	7%
LS-SVM - Gaussian Kernel	16%	14%	25%	5%

Table 1: The CV errors in prediction of different hourly power consumptions by different methods.

We can see that LS-SVM with a linear kernel is at least as good as LS, and they are especially very close to each other when the error is large. The first part is expected from the definition of these methods. One can easily show that with the same feature mappings, the solution of LS-SVM converges to that of LS if we send the regularization parameter C to infinity. In the cases where the two models are close, our cross-validation chose a high value for C . LS-SVM with a Gaussian kernel seems to be the most accurate learning method for all of the data-sets, while LS-SVM with a polynomial kernel seems to have the worst accuracy. We suspect that the sub-par performance of the polynomial kernel might be due to effect of the parameter C_P . A large C_P would give higher weights to the interactions between different attributes. Physical intuition (and the results for LS-SVM with a linear kernel) suggest that monomial functions of the attributes T, L , and t are more important predictors of the power consumption than their interactions, which suggests that a small C_P should be used. However, as explained before, the interaction between the holiday indicator attribute d and the other variables helps us capture the difference between the power consumption on the weekends and the weekdays, which we may not be able to capture very well with a small C_P .

Although CV is a good indicator for performance of our methods, there are qualitative differences between the methods that are not reflected through CV. Comparing the predicted time series with the actual one can be useful for a qualitative assessment of the algorithms. Here we look a data set that all four methods approximated relatively well, the AC power consumption. Figure 1 shows the actual and predicted AC power consumption on the second floor for each method over our 20-day test period.



(c) Linear regression (LS). (d) LS-SVM with linear kernel. (e) LS-SVM with Gaussian kernel. (f) LS-SVM with polynomial kernel.

Figure 1: The predicted AC power consumption on the 2nd floor. In all plots the red curve is the predicted time series and the blue curve is the actual power data.

It can be seen that all methods are more accurate on the weekdays, compared to the weekends and holidays. This is likely because for those days i) there are more training examples, and ii) the consumption patterns are more regular. If we wanted to compare the methods qualitatively, again LS-SVM with a Gaussian kernel seems to be the best. Specifically, it is more accurate on the weekends. LS-SVM with linear kernel is a close second. Its performance on the weekends is close to (or maybe even slightly better) than Gaussian kernel, however it shows a slight undershoot on the weekdays. Simple regression has good accuracy on the weekdays, but has a large undershoot on the weekends. And finally, LS-SVM with polynomial is less accurate than all the other methods, the weekdays, and on the weekends, its accuracy is close to that of LS, while worse than the other two.

5 Conclusion

In this project, we implemented the linear Least squares, and the support vector machines least squares algorithms with various kernel functions for predicting the hourly power consumption in the using the sensor measurement data in the Y2E2 building, and compared (qualitatively and quantitatively) the accuracy of the results obtained using these methods.

References

- [1] U.S. Department of Energy, Buildings Energy Data Book, D&R International, Ltd., 2010, available at: <http://buildingsdatabook.eren.doe.gov/>.
- [2] Richard E. Edwards, Joshua New, and Lynne E. Parker. Predicting future hourly residential electrical consumption: A machine learning case study. *energy and Buildings*, 49(0) (2012) 591-603
- [3] A.J. Smola and B. Schlkopf, A tutorial on Support Vector Regression. *Statistics and Computing*, 14 (3) (2004) 199222.
- [4] J. Suykens, T.V. Gestel, J.D. Brabanter, B.D. Moor, J. Vandewalle, Least Squares Support Vector Machines. *World Scientific Publ. Co. Inc.*, 2002.
- [5] Chih-Wei Hsu, Chih-Chung Chang, Chih-Jen Line, A Practical Guide to Support Vector Classification 2010.