Predicting Dartmouth's Daily Energy Usage

By Henry I. Stewart and Tev'n J. Powers

Spring 2012

Most Dartmouth college students typically only pay one bill per quarter. They are particularly insulated from the costs of their daily or monthly kW usage. At Dartmouth, the GreenLite project attempts to make students more aware of their hourly electricity consumption. Given the rising cost of energy, we would like to be able to predict how much energy the entire Dartmouth campus will use on a minutely basis. Knowing this information could be useful to the Dartmouth Power Plant (Facilities Operations and Management) in offsetting fuel costs or finding ways to reduce utility costs.

Our data comes from the GreenLite developr page at:
http://dev.greenlite.cs.dartmouth.edu
Professor Lorie Loeb was kind enough to give us an account to gather information from the GreenLite's database.The website is a front end interface that retrieves data from a MySQL server that stores the feature data (temperature, paper usage, power, energy, etc).

We wrote a python script that parsed the sample data exported from the GreenLite developer interface. Our script removed all spaces, and non-numerical characters from the exported file. After the script parsed power, energy, temperature, and annualized cost, our script wrote files that supported .csv file format.

At the beginning of the project, we sought to predict how much energy the entire Dartmouth campus would produce on a daily and monthly basis. For the first milestone, we trained our algorithm using daily energy. As we sought to expand the sample set (use SMO algorithm), we began to run into issues with data collection. We found that the GreenLite project interface does have not enough daily samples from the "Dartmouth " key to support the use of the SMO algorithm. We decided to change the sample data from daily data to minutely data. The advantage of using minutely data is we may gather more samples on a shorter calendar time frame. For instance, just two weeks of minutely data provides roughly 10,080 samples. Hourly data was not available before January 1, 2011, therefore we determined that there was not enough hourly data available to give a large enough sample set

for our training algorithm (specifically SMO).

We limited the scope of our training to May 7, 2012 to May 17, 2012. Therefore, we sought to predict energy usage on May 18, 2012. We also revised and added features from the milestone submission. In our final report, our features included: power (kW) in 2012 and 2011, the annualized cost of energy (in dollars) in 2012 and 2011, energy in (kWh) in 2011, binary vectors that determined whether that day was a weekend or weekday in 2011 and 2012. Our output vector represented minutely energy usage on May 18, 2012. We staggered the data so that there was a 24 hour delay in our sample.

There were a few missing pieces to our dataset that we did not know at the start of the project. In the GreenLite developer interface, there is a key labelled "Dartmouth" that we thought allowed us to retrieve data from the entire campus. We later found out from Professor Loeb that this data set did not include minutely data for the entire campus. Instead, the "Dartmouth" node provided only an aggregation of the most recently renovated dormitories and a handful of administrative buildings. This data set did not include any of the largest energy consumers on campus (e.g. the Alumni Gym and Burke Chemistry building). In other words, our dataset only included a few dormitories and administrative buildings. We also ran into missing data on several random minutes throughout 2012. This may have been due to server error or instrument error.

Methodology Approaches

Energy consumption can create a complex function. Many algorithms have been used to estimate building consumption. Some of these include advanced neural networks, decision trees (classification based on discrete categories), and SVM. In order to perform our prediction of Dartmouths daily energy consumption, our algorithm of choice was Epsilon-Insensitive Support Vector Regression. This algorithm has been noted for its accuracy in predicting complicated time series data such as stock, energy, weather, etc. Usually Support Vector Machines (SVM) are used as a method of classification, but Alex Smola and Bernhard Scholkopfs paper A Tutorial on Support Vector Regression extends the use of SVM to function estimation. Specifically with Epsilon-Insensitive SVR, our goal is to find a function f(x) that has at most epsilon derivation from the obtained targets y. The epsilon parameter is important because it allows for estimations that have some error, but lie

within the +/- epsilon region of the original target value. The dual problem can be written as:

$$L = \frac{1}{2}||w||^2 + C\sum_{i=1}^{l}(\gamma_i + \gamma_i^*) - \sum_{i=1}^{l}(\mu_i\gamma_i + \mu_i\gamma_i^*)$$
$$-\sum_{i=1}^{l}\alpha_i(\epsilon + \gamma_i - y_i + \langle w, x_i\rangle + b)$$
$$-\sum_{i=1}^{l}\alpha_i^*(\epsilon + \gamma_i^* - y_i - \langle w, x_i\rangle + b)$$

After optimization and solving for the Lagrange multipliers α and α^* the function becomes:

$$f(x) = \sum_{i=1}^{l}(\alpha_i - \alpha_i^*)K\langle x_i, x\rangle + b$$

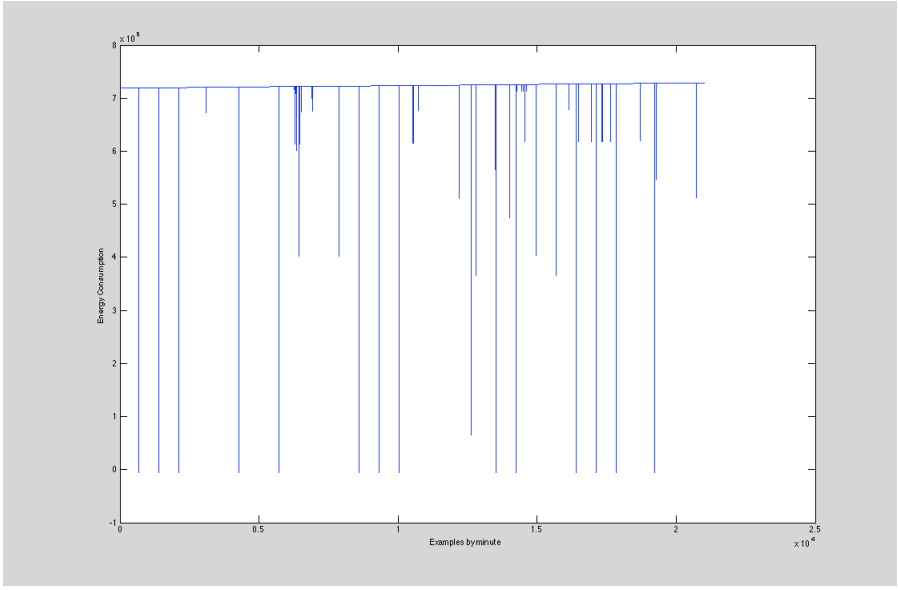
Optimization is the minimization of the Lagrangian equation with respect to the primal variables at the same time as maximizing the Lagrange multipliers (the dual variables). Since our data forms a nonlinear function, we employ the use of a Kernel function to make the SV algorithm nonlinear. We use the Gaussian Radial Basis Function Kernel that has been used in many previous studies for building energy analysis (Brown, Leigh, Brown ,et. al). There are two ways to go about solving for the Lagrange multipliers in the function. The simplest implementation is to use a commercial Quadratic Programming function (i.e. Matlabs quadprog) to solve for the multipliers. However the limitation with this implementation is that the Quadratic Program can become very large (read: too large) as the number of examples in the training data increases. When the number of examples exceeds a few hundred, then the program will not have enough memory to perform the computation. Initially we thought that a QP would be sufficient if we could use a few hundred examples of daily (or even hourly) data. We were told that the Greenlite Project database had daily, hourly, and minutely data for each feature dating back ten years. However we soon realized that the project was inconsistent, and only had sporadic daily and hourly data. Therefore we were forced to use the minutely data instead. In this case, we would need thousands of more examples to create a large time window, than we would have given the case of daily or hourly data.

Facing this new challenge, we decided to implement Sequential Minimization Optimization (SMO). The SMO algorithm is the second way of

computing the Lagrange multipliers. This algorithm is designed to speed up convergence for large datasets and to make sure it converges even in degenerate conditions. Instead of solving a large quadratic program, SMO iterates through all of the Lagrange multipliers, optimizing the objective value jointly for both parameters. This is a better method of computing the final Lagrange multipliers because when optimizing two parameters, the step can be done analytically, not with a quadratic program.

Final Thoughts

Below is a plot of the true energy consumption from 5/7/2012 5/17/2012:
Figure 1:



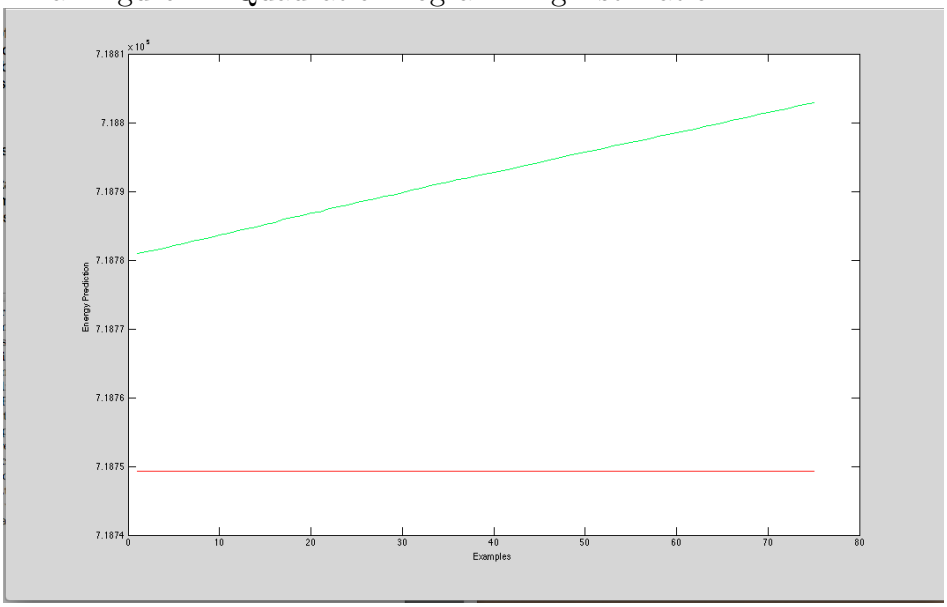
Although the data does behave irregularly, we would expect to see more of a wave-like pattern. This would demonstrate that energy consumption rates at a given hour would be similar across multiple days (i.e. energy consumption at 5 p.m. or 2 a.m. today would be similar to energy consumption to 5 p.m. or 2 a.m. tomorrow). However this is not the behavior demonstrated by our data.

Below is the result of the Quadratic Program and an open source implementation of Online Support Vector Regression. We have only plotted data

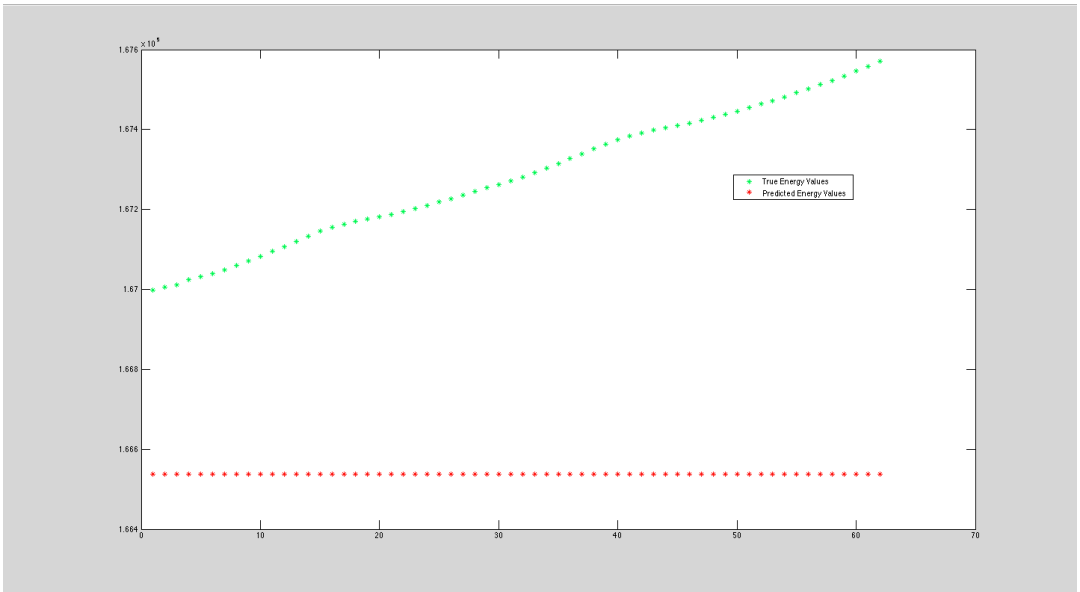
for a small subset of examples from our total data set because the Quadratic Program cannot handle thousands of examples.

As you can see, these results are not beneficial in terms of predicting campus energy consumption over a significant period of time. When the data collected only stretches over a few hundred minutes there will not be that much of a change in consumption. With that being known, our problem now looks like a linear problem. Since the models were trained on a few hundred data samples that varied very slightly from one another, the predictions will similarly increase very slowly.

a. Figure 2 - Quadratic Programming Estimation



b. Figure 3 - Open Source Online Support Vector Regression Results



The key to solving this problem is a successful implementation of SMO. With a correct implementation, the algorithm can then be trained on thousands of examples and have a very good model for prediction. Unfortunately, we did not have a successful implementation of this algorithm by the time of our submission.

References

[1] Support Vector Machines for Classification and Regression by Steve R. Gum
[2] A Tutorial on Support Vector Regression by Smola and Scholkopf
[3] A Fast Algorithm for Training Support Vector Machines by Platt
[4] The Simplified SMO Algorithm, Stanford CS 229, Andrew Ng
[5] http://onlinesvr.altervista.org/Download.html
[6] Kernel Regression for Real-Time Building Energy Analysis by Brown et. al.
[7] Efcient SVM Regression Training with SMO by Flake and Lawrence
[8] Comparing Support Vector Machines with Gaussian Kernels to Radial Basis Functi
Classifiers by Scholkopf et al.
[9] http://www.codeforge.com/read/131255/svm_SMO.m__html
[10] A Fast SMO Training Algorithm for Support Vector Regression by Zhang et al.