12-752: Data-Driven Energy Management of Buildings Assignment#2

Mario Bergés
Assistant Professor
Department of Civil and Environmental Engineering
Carnegie Mellon University
Pittsburgh, PA 15213
`marioberges@cmu.edu`

November 25, 2015

Some notes before you begin:

- Make sure you document everything you do, and not just write down the answer to the question. This will both help during grading as well as improving your learning process.
- Do not write down any solution or process that you do not understand. If you feel that you do not understand how to do something, seek some help: e-mail the TAs or the instructor
- To submit your assignment, please do so using Blackboard. Two files should be uploaded via Blackboard: (a) the IPython Notebook (i.e. a .ipynb file) documenting all the tasks found in the assignment and all of your answers (including the output of your code); and (b) a PDF copy of this notebook
- Please upload a single compressed ZIP file containing the above, and name it as follows: *andrewID_assignment_#.zip* (where *andrewID* is your AndrewID and # is the assignment number

1 Preface

This assignment builds on the 2014 version of the same. In other words, you should first make sure you understand the steps taken in the solution to Assignment #2 from 2014 before continuing with this assignment. The solutions are provided in Blackboard under the Lectures forlder (Lecture #5, to be precise). You should also make sure to have read Paper #2, which discusses the piece-wise linear regression model that we will be implementing in this assignment.

1.1 Loading the Dataset

In this assignment we will be utilizing a dataset of historical electric power measurements collected for a number of buildings on campus during the last 12 months, as well as a dataset of temperature measurements collected from a weather station located on campus, during the same period of time. The datasets can be found on Blackboard under the Datasets folder.

Just as with the last assignment, the first step will be to laod the CSV files into memory. There are, however, some important caveats that you should consider when doing this:

- We now have two files, with different number of rows and columns.
- The difference in the number of rows arises due to two facts:
  - The temperature and power measruements are not collected at the same sampling rate, nor for the sampe sampling period.
  - The `campusDemand.csv` file has measurements for many different meters on campus (i.e., even if the temperature and power measurements were done at the same sampling rate and for the same period of time, the power measruement file would be larger).
- You will need to perform some form of interpolation in order to harmonize the two time-series (i.e., to make them have the same time-stamps).
- We will only be focusing on one single meter from the dataset, so pick one and forget about the rest.

> **Task #1 [20%]:** Load the CSV files into memory. Specifically, store the contents of the files into a single variable named `data`, which should be an `ndarray`, or a similar data structure, containing 3 columns (time stamp, power measurement, temperature) and as many rows as required after harmonizing the datasets.

2 Preparing the Data

> **Task #2 [10%]:** Create an array to store the indeces for the rows in `data` that correspond to measurements collected during the weekends (i.e., between 12:00am on Saturday until 11:59pm on Sunday).

> **Task #3 [10%]:** Create another similar array to store the indices for the rows in `data` that correspond to "occupied" times (i.e., times during which the building is likely to be occupied)

Now let's see how load changes as a function of temperature, during occupied and unoccupied times.

> **Task #4 [5%]:** Create a figure with two subplots (one row, two coluns). The left subplot will show electrical power as a function of temperature during occupied times. The right subplot will show electrical power as a function of temperature during unoccupied times. Comment on what you find out.

3 Preparing the model

As you may remember, the model we are trying to fit to this data is as follows:

$$L_o(t_i, T(t_i)) = \alpha_i + \sum_{j=1}^{k} \beta_j T_{c,j}(t_i) \tag{1}$$

This is the model for the occupied case and though there is a separate (similar) model for the unoccupied case, it is really easy to see that it is an instance of this same model when $k = 1$.

Task #5 [15%]: Create a function called `Tc`, which takes as input an ndarray of temperature values, and a tuple containing the temperature boundaries, and computes the values for $T_{c,j}$, for every value of j (i.e., from 1 to k, where k is a number you decide is appropriate for your particular dataset. The starter file has a definition of the function and a stub for the array that should be returned.
The returned array should be similar to the table that is shown in the paper, though with many more rows (as many as the number of timestamps in your dataset).

Using this function, you should be able to obtain all the values for T_c for any subset of the data (i.e., the weekends, weekdays, occupied or unoccupied times).

Task #6 [15%]: Create another function called `DesignMatrix`, which takes in three arguments: (1) an ndarray of temperature values, (2) a corresponding ndarray of timestamps and (3) a tuple of temperature boundaries. The function then returns the design matrix for the regression problem (i.e., the matrix we have been discussing in class). Internally, this function should call the function you created previously, `Tc`, in order to obtain k of the columns of the matrix. Ideally your funciton would return something with the `numpy matrix` data type.

If you are confused by the term "Design Matrix", don't be. It's just a fancy way of calling the matrix $\mathbf{X}$ in the generic formulation of the regression problem below:

$$y = \mathbf{X}\beta + \epsilon \tag{2}$$

where β is the vector of coefficients, y is the dependent variable (in our case the power measurements), and ϵ is the vector of errors or the differences between our model predictions (i.e., $\mathbf{X}\beta$ and the actual values of the power consumption. Wikipedia has more details: Design Matrix

You may remember that in addition to the columns provided by `Tc`, the other columns can be easily generated by repeating an identity matrix ($\mathbf{I}$) of size, say, p, where p is the number of 15-minute intervals in the period of time you are analyzing (i.e., $p = 480$ in the case of weekdays).

You may also want to consider adding a column of ones to the matrix, so that you can have a coefficient that will absorb the average of the whole dataset. If you don't do this, it is the same as doing a simple linear regression in 2 dimensions where your model $y = ax + b$ has $b = 0$, or no intercept with the y-axis.

4 Fitting the model

In this section, we will assume that you have an time-series of power measurements stored in an ndarray called, say, `power_values` and a design matrix called, say, `X`. What we want to do now is to find a vector β that can minimize the difference between our model predictions $\mathbf{X}\beta$ and the power measurements (which, mathematically we call y). We want this difference to be calculated using a square loss function, so essentially this amounts to finding the value of β that minimizes this:

$$\|\mathbf{X}\beta - y\|^2 \tag{3}$$

You may remember from our first lecture on regression that the solution to this is:

$$\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T y \tag{4}$$

So we can use this formula to calculate our least-square estimates of the coefficients in β.

Task #7 [15%]: Create a function `beta_hat`, which takes in a design matrix `X` and an ndarray of power values `power_values` and outputs the least square estimators for β, namely $\hat{\beta}$, using the formula above.

5 Putting it all together

So far, you have only created functions but not used them for anything other than testing. This is the time to put it all together and figure out if it really works. As you may have seen, the focus with this assignment is not so much on whether we are getting the right result out of our model (so far we have assigned 90% of the grade and have not really done much with the dataset), but rather to focus on understanding the concepts behind the implementation of linear regression analysis techniques.

The last 10% of the grade, however, will be used to check how well your implementation worked. Here's what I want you to do:

Task #8 [2%]: Select a specific subset of the data you want to train a model on. This could be, for instance, unoccupied weekends. Separate the dataset into two portions of the same number of samples. One will be used for training (you can call that one, say, `trainData`) and another one for testing (e.g., `testData`).

Task #9 [2%]: Create a tuple `T_bound` that defines the boundaries for the temperature intervals that define the different temperature components $T_{c,j}$, $\forall j \in 1, \ldots k$ in your subset dataset (you can look at both the training and testing for this).

Task #10 [2%]: Use those boundaries to call `DesignMatrix` and generate the design matrix for your problem (using only the training set)

Task #11 [2%]: Using the design matrix from above, find the $\hat{\beta}$ values using the `beta_hat` function.

Task #12 [2%]: Finally, in a single figure with a graph of power versus time, plot the actual power consumption in your test set, and the predictions of your model (i.e., $\mathbf{X}\hat{\beta}$).