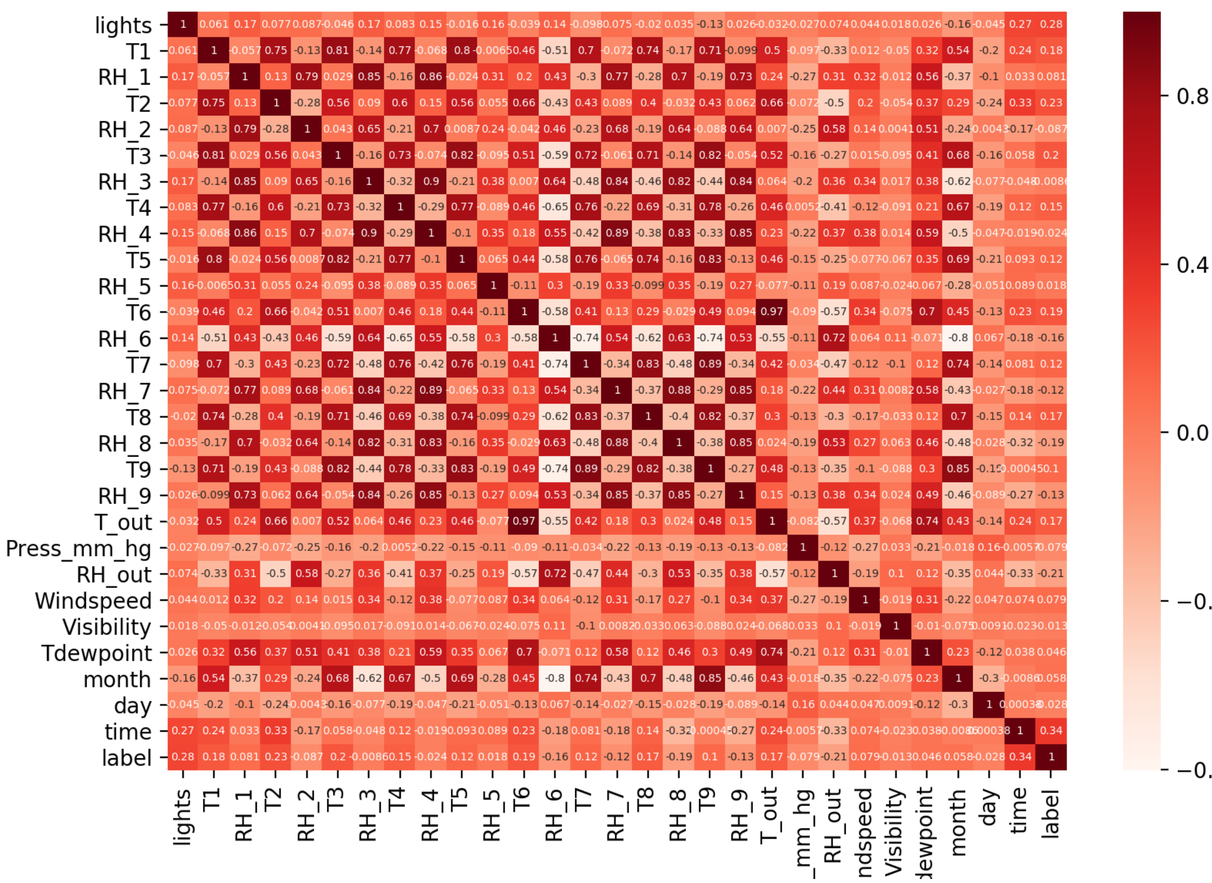
CS 334: Homework #3
Ruohan Zhang

1. Feature Extraction + Feature Selection
(a) The first column of the energy data has a date and timestamp (e.g., 3/20/16 5:30). What are features that you can extract from this data that might be useful for predicting energy usage?

Answer: I can split this column and potentially get 4 features: day, month, year and the time of the day. For this dataset, year is always 16, so I directly drop 'year'.
Add feature 'day' (int), 'month' (int) and 'time' (float, e.g 17:20 = 17.333333).
Drop 'date'.

(b) Plot the correlation matrix using a heatmap:



(c) Using (b), identify the features that you think should be used to train a linear regression model and your reasoning for using them

Answer:

1. Feature and Target:
   Drop features which has a correlation coefficient's absolute value less than 0.1 because correlation coefficient close to 0 implies almost no correlation
2. Feature and Feature:
   One of the assumptions of linear regression is that the independent variables need to be uncorrelated with each other. Thus when two features have strong correlation (coefficient > 0.7), drop the one that has a weaker correlation with target. If a feature has multiple highly correlated features, drop half of the correlated features that has smallest correlation coefficients with target.

In total I select 9 features from the train data, and I select the same 9 features directly from test data instead of select after evaluating the correlation between teat data's feature values.

(d) What other preprocessing steps do you need to do to the data?

Answer: Standard Scale all features values

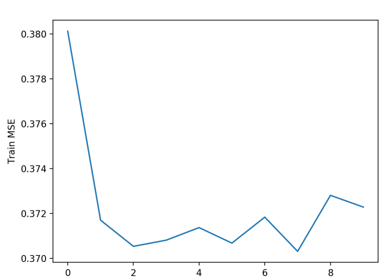
2. (3+7=10 pts) Linear Regression: Single Unique Solution

3. (25+10+5=40 pts) Linear Regression using SGD

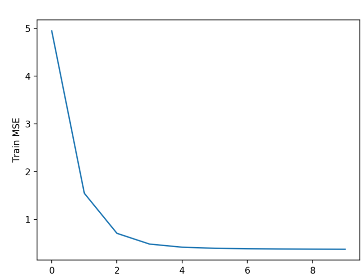
(b) For a batch size of 1 (i.e., b = 1) and a random subset of 40% of the training data, try various learning rates and plot the mean squared error on the training data as a function of the epoch (i.e., one epoch = one pass through the training data). What seems to be an optimal learning rate? Justify your answer based on the plot.

Epoch = 10

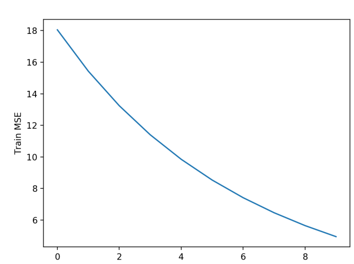
lr=0.001



lr = 0.0001

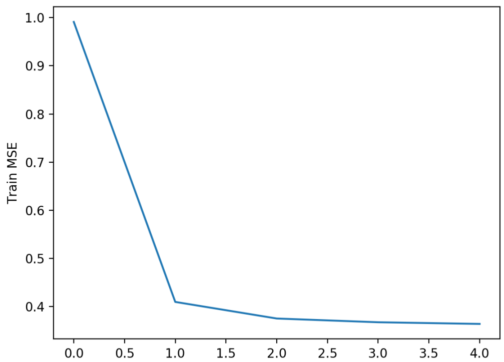


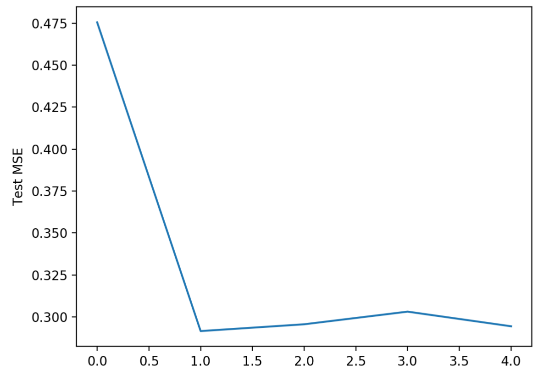
lr = 0.00001



Answer: According to the plot, 0.001 is too fast thus diverge after 10 epoch and 0.00001 is too slow. 0.0001 seems to be an optimal learning rate.

(c) Using the optimal learning rate from (b), train the model on the entire dataset and plot the mean squared error on the training data and the test data as a function of the epoch.





4. (20+5=25 pts) Comparison of Linear Regression Algorithms using SGD and Closed Form solutions

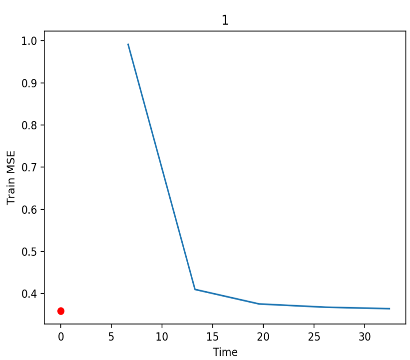
(a) Plot the mean squared error of the training data and the test data as a function of the total time for different batch sizes.

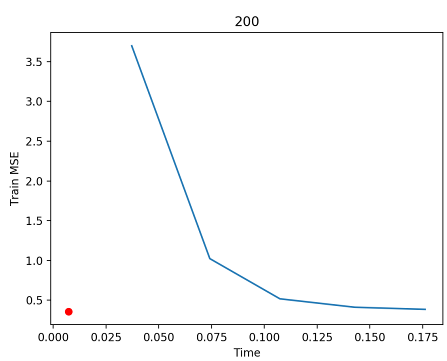
(lr, bs, epoch): (0.0001, 1, 5), (0.01, 200, 5),(0.1, 1000, 5),(1, N, 5)

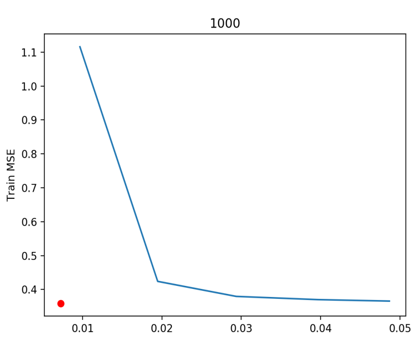
Red point is the close form solution:

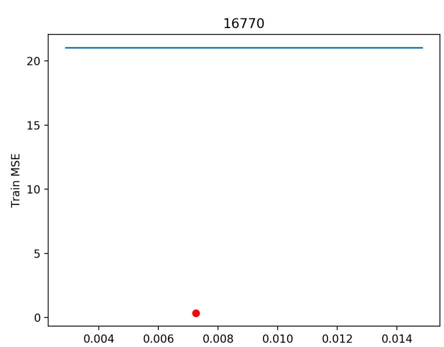
```
{0: {'time': 0.007259845733642578, 'train-mse': 0.3590447177262758, 'test-mse': 0.30271971818002}}
```

Train:

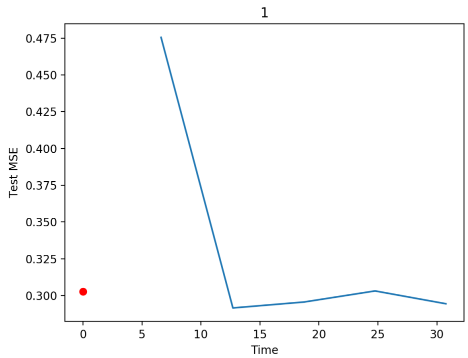


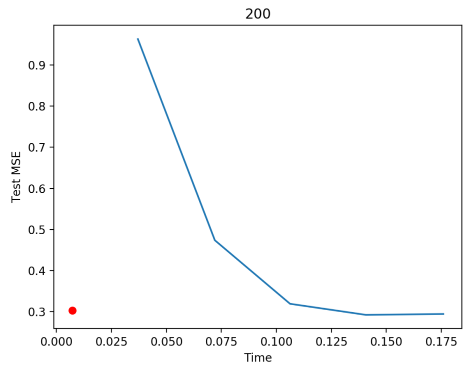


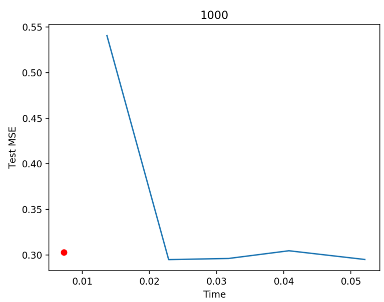


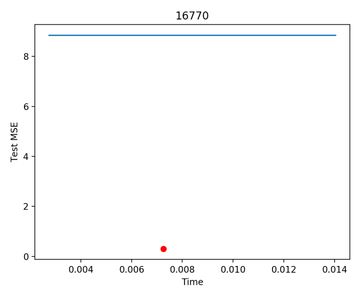


Test:

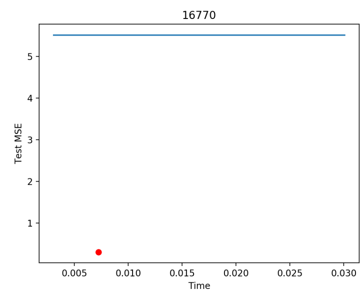








(1, N ,5)



(0.0001, N, 10)

(b) What are your observations based on the plot in (a)? What are the trade-offs between different batch sizes as well as the closed form solution?

Answer:

Computational time decrease with the increase of batch sizes but large batch size’s optimal learning rate is large, and easily cause diverge result.

Closes form solution is the fastest and accurate. But if our data is a noninvertible matrix, we are not able to use close form solution.