CS534 — Implementation Assignment 1 — Due 11:59PM Oct 19th, 2019

General instructions.

1. The following languages are acceptable: Java, C/C++, Python, Matlab

2. You should work in a team of two or three people. Each team will only need to submit one copy of the source code and report. You need to explicitly state each member's contribution in percentages (a rough estimate)[1].

3. Your source code and report will be submitted through TEACH

4. You need to submit a readme file that contains the programming language version you use (e.g. python 2.7) and the command to run your code (e.g. python main.py).

5. Please make sure that you can be run code remotely on the server (i.e. babylon01) especially if you develop your code using c/c++ under visual studio.

6. Be sure to answer all the questions in your report. You will be graded based on your code as well as the report. In particular, **the clarity and quality of the report will be worth 10 pts**. So please write your report in clear and concise manner. Clearly label your figures, legends, and tables.

7. In your report, the results should always be accompanied by discussions of the results. Do the results follow your expectation? Any surprises? What kind of explanation can you provide?

[1] Grades will be assigned assuming all members of the team contribute equally. If there is clear deviation from this expectation, adjustment will be made to individual team member's grade.

Linear regression with L_2 regularization (total points: 80 pts + 10 report pts + 10 result pts)

For the first part of the assignment, you need to implement linear regression with L_2 (quadratic) regularization, which learns from a set of N training examples $\{\mathbf{x}_i, y_i\}_{i=1}^N$ an weight vector $\mathbf{w}$ that optimize the following regularized Sum of Squared Error (SSE) objective:

$$\sum_{i=1}^{N}(y_i - \mathbf{w}^T\mathbf{x}_i)^2 + \lambda\|\mathbf{w}\|^2 \tag{1}$$

To optimize this objective, you need to implement the gradient descent algorithm. Because some features have very large values, for part of the assignment you are asked to normalize the features to the range between zero and one. This will have an impact on the convergence behavior of gradient descent.

Data. The dataset consisted of historic data on houses sold between May 2014 to May 2015. You need to build a linear regression that can be used to predict the house's price based on a set of features. You are provided with three data files: **train**, **test** and **validation**, all in csv format. You are provided with a description of the features as well. The first column of each file contains the dummy feature taking the constant value of 1 for all examples. The last column in the files **train** and **validation** stores the target y values for each example, We omitted y values from **test** file. You need to learn from the training data and tune your parameters with the provided validation data to chose the best model. Your submission will include a prediction file of the testing data that has the predicted y values generated by the best model you learned.

General guidelines for training. For all parts, you should train your model until the convergence condition is met, i.e., the norm of the gradient is less than $\epsilon = 0.5$. If you find that this specific threshold makes the training time too long for some learning rate values, feel free to use higher values and report the value you used. It is a good practice to monitor the norm of the gradient during the training. You need to report the SSE (the first term in the Eq. 1) on the training data and the validation data respectively for each value of the hyperparamter you tune (e.g. learning rate, λ). Use the best model you learned to do predction on the test data and submit the prediction file.

Part 0 (10 pts) : Preprocessing and simple analysis. Perform the following preprocessing of the your data.

(a) Remove the ID feature. Why do you think it is a bad idea to use this feature in learning?

(b) Split the date feature into three separate numerical features: *month*, *day* , and *year*. Can you think of better ways of using this date feature?

(c) Build a table that reports the statistics for each feature. For numerical features, please report the mean, the standard deviation, and the range. Several of the features (waterfront, grade, condition (the later two are ordinal)) that are marked numeric are in fact categorical. For such features, please report the percentage of examples for each category.

(d) Based on the meaning of the features as well as the statistics, which set of features do you expect to be useful for this task? Why?

(e) Normalize all features to the range between 0 and 1 using the training data. Note that when you apply the learned model from the normalized data to test data, you should make sure that you are using the same normalizing procedure as used in training.

Part 1 (30 pts). Explore different learning rate for batch gradient descent. For this part, you will work with the preprocessed and normalized data and fix λ to 0 and consider at least the following values for the learning rate: $10^0, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}, 10^{-7}$.

(a) Which learning rate or learning rates did you observe to be good for this particular dataset? What learning rates make the gradient decent explode? Report your observations together with some example curves showing the training SSE as a function of training iterations and its convergence or non-convergence behaviors.

(b) For each learning rate worked for you, Report the SSE on the training data and the validation data respectively and the number of iterations needed to achieve the convergence condition for training. What do you observe?

(c) Use the validation data to pick the best converged solution, and report the learned weights for each feature. Which feature are the most important in deciding the house prices according to the learned weights? Compare them to your pre-analysis results (Part 0 (d)).

Part2 (30 pts). Experiments with different λ values. For this part, you will test the effect of the regularization parameter on your linear regressor. Please exclude the bias term from regularization. It is often the case that we don't really what the right λ value should be and we will need to consider a range of different λ values. For this project, consider at least the following values for λ: $0, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 100$. Feel free to explore other choices of λ using a broader or finer search grid. Report the SSE on the training data and the validation data respectively for each value of λ. Report the weights you learned for different values of λ. What do you observe? Your discussion of the results should clearly answer the following questions:

(a) What trend do you observe from the training SSE as we change λ value?

(b) What tread do you observe from the validation SSE?

(c) Provide an explanation for the observed behaviors.

(d) What features get turned off for λ= 10, 10^{-2} and 0 ?

Part 3 (10 pts). Training with non-normalized data Use the preprocessed data but skip the normalization. Consider at least the following values for learning rate: 1, $0, 10^{-3}, 10^{-6}, 10^{-9}, 10^{-15}$. For each value , train up to 10000 iterations (Fix the number of iterations for this part). If training is clearly diverging, you can terminate early. Plot the training SSE and validation SSE respectively as a function of the number of iterations. What do you observe? Specify the learning rate value (if any) that prevents the gradient descent from exploding? Compare between using the normalized and the non-normalized versions of the data. Which one is easier to train and why?

Submission. Your submission should include the following:
1) Your source code, together with detailed instruction on running your code (see general instruction items 4 and 5);
2) Your report (see general instruction items 6 and 7), which should begin with a general introduction section, followed by one section for each part of the assignment;
3) a prediction file containing the predicted y values for the provided test file, one test example per line. This prediction file will be scored against the ground truth y values and 10% of the grade will be based on this score.