Fall 2018 DS 4400

Homework 1

Due date: Friday, September 28 2018

Instructions

Please make sure your name appears in all the files and source code you submit. Submit a PDF file named "%LASTNAME%_HW1.pdf" in Gradescope that will include the following:

- Link to Jupyter notebook with Python or R code. You can store the code in an online service (such as Google drive, Dropbox, or private github).
- Link to a simple README file with instructions on how to run your code.
- Answers to the questions. You can use Latex or Word to generate the PDF.

Course policy on collaboration and cheating:

- You may discuss the concepts with your classmates, but write up the answers entirely on your own.
- You cannot share your code with your classmates.
- You cannot use code from the Internet for your assignment.

Dataset: The dataset for this assignment is available at: `https://drive.google.com/open?id=19-8Vj_ReERinugT-0aB892--8eHYrMLB`

The prediction task is to predict the price of a house (column price) given the other features. Please ignore the columns id and date, as well as the categorical column zipcode. File "kc_house_data.csv" includes all the records in the dataset. The training file "train.csv" and testing file "test.csv" include each 1000 records extracted from the dataset.

Problem 1 [Average, variance, and correlation] 10 points

In this problem, we will perform some exploratory data analysis.

(a) For each feature, write code to compute the average value, the min and max values, as well as its variance.

(b) Compute the correlation coefficient of each feature with the response. Which feature are positively correlated (i.e., have positive correlation coefficient) and which ones are negatively correlated with the response? Which features have highest correlation with the response (both positive and negative)?

Problem 2 [Linear regression] - 20 points

In this problem, you will use an existing package of your choice for training and testing a linear regression model.

(a) Use an existing package to train a linear regression model on the training set. Report the coefficients of the linear regression models and the 3 metrics of interest: MSE, RSS, and R^2.

(b) Perform feature standardization so that each feature has mean 0 and variance of 1. Train again a linear regression model on the training data. Compare the results with the previous models in terms of the metrics of interest: MSE, RSS, and R^2.

(c) Evaluate both models on the testing set. Report the same metrics (MSE, RSE, and R^2) on the testing set.

(d) Interpret the results in your own words. Which features contribute mostly to the linear regression model? Is the model fitting the data well? How large is the model error?

Problem 3 [Closed-form solution for linear regression] 25 points

In this problem, you will implement your own linear regression model, using the closed-form solution we derived in class. You will also compare your model with the one trained with the package.

(a) Implement simple linear regression and train a model for one feature (sqft_living) using the training set. Write code to predict a response for a new single-dimensional data point in the testing set.

(b) Implement multiple linear regression using matrix operations and train a model on the training set. Write code to predict a response for a new multi-dimensional data point in the testing set.

(c) Compare the models given by your implementation with those trained in Problem 2 by the R or Python packages. Report the MSE, RSE, and R^2 metrics for the models you implemented. Compare the coefficients output by your model with the ones computed by the package.

Problem 4 [Gradient descent] 25 points + Extra Credit 20 points

In this problem, you will implement your own gradient descent algorithm and apply it to linear regression. Use the scaled dataset.

(a) Write code for gradient descent for training linear regression using the algorithm from class.

(b) Vary the value of the learning rate (5 different values) and the number of iterations (5 different values)) and report the value of θ for each of the 25 combinations, as well as the MSE metric on the training set. Report the MSE on the testing set.

(c) Tune your implementation to obtain results close to those obtained with the package. Write some observations: How does the objective change with different learning rates; how many iterations are needed, etc.

(d) **Extra credit - 10 points** You will get extra credit if your GD implementation of linear regression achieves MSE very close to the least-square solution given by the package.

(e) **Extra credit - 10 points** You will extra credit if your GD implementation of linear regression can run on the entire "kc_house_data.csv" dataset efficiently. Report the running time of your training algorithm for the entire dataset and compare that with the running time of the package.

Problem 5 [Ridge regression] 20 points

In this problem, you will derive the optimal parameters for ridge regression and implement a ridge regression model. In ridge regression, the loss function includes a regularization term:

$$J(\theta) = \sum_{i=1}^{n} [h_\theta(x^{(i)}) - y^{(i)}]^2 + \lambda \sum_{j=1}^{d} \theta_j^2$$

(a) Write the derivation of the closed form solution for parameter θ that minimizes the loss function $J(\theta)$ in ridge regression.

(b) Modify your linear regression implementation to handle ridge regression. Compare the results of linear regression and ridge regression on the dataset. Take several values of the regularization parameter λ and output the MSE, RSE, and R^2 metrics. Which model performs better? Interpret the results in your own words.