CMPE 258, Spring 2018

Assignment #2

Due 11:59pm on Sunday, February 13th, 2018

Notes

This programming assignment should be submitted in Canvas as a format of ipython notebook (assignment_2_yourFirstName_LastnName.ipynb).
You can discuss how to solve the problem with other students or search internet or other resources, but the work should be your own.
The submitted ipynb should be executable without any extra work.

1 (30pts). Polynomial regression / overfitting / regularization

Using Jupyter notebook, load the data (ex2data1.csv).

1-1. Fit the data using linear (1st order) regression model (matrix form, gradient descent method).
Plot the data with the fitted line. Print optimized weights. Print Root Mean Squared Error (RMSE).

1-2. Fit the data using 2nd order polynomial regression model (matrix form, gradient descent method).
Plot the data with the fitted line. Print optimized weights. Print Root Mean Squared Error (RMSE).
Note: Do not forget feature normalization.

1-3. Fit the data using 4th order polynomial regression model (matrix form, gradient descent method).
Plot the data with the fitted line. Print optimized weights. Print Root Mean Squared Error (RMSE).
Note: Do not forget feature normalization.

1-4. Fit the data using 16th order polynomial regression model (matrix form, gradient descent method).
Plot the data with the fitted line. Print optimized weights. Print Root Mean Squared Error (RMSE).
Note: Do not forget feature normalization.

1-5. Fit the data using 16th order polynomial regression model with ridge (L2 penalty) regularization (matrix form, gradient descent method).
You need to try at least 3 different L2 penalty (for example, $\lambda = 0.1, 1, 10$). Plot the data with the fitted line. Print optimized weights. Print Root Mean Squared Error (RMSE)
Note: Do not forget feature normalization.

1-6. Fit the data using 16th order polynomial regression model with scikit-learn Ridge model.
You need to try at least 3 different L2 penalty (for example, $\lambda = 0.1, 1, 10$). Plot the data with the fitted line. Print optimized weights. Print Root Mean Squared Error (RMSE)

1-7. Fit the data using 16th order polynomial regression model with scikit-learn Lasso model.
You need to try at least 3 different L1 penalty (for example, $\lambda = 0.1, 1, 10$). Plot the data with the fitted line. Print optimized weights. Print Root Mean Squared Error (RMSE)

2 (30pts). Polynomial regression with train/validation/test

Using Jupyter notebook, load the data (ex2data2.csv).
The first column is the size of the house (in square feet), the second column is the price of the house.

You need to split the data into training/validation/testing data set as 60% / 20% / 20%.
Please use np.random.seed(1) to have consistent data for evaluation.

2-1. Fit the data using linear (1st order) regression model (matrix form, gradient descent method).
Plot the training data with the fitted line. Using the optimized weights, please calculated Root Mean Squared Error (RMSE) of training and testing data.

2-2. Fit the data using 2nd order polynomial regression model (matrix form, gradient descent method).
Plot the training data with the fitted line. Using the optimized weights, please calculated Root Mean Squared Error (RMSE) of training and testing data.
Note: Do not forget feature normalization.

2-3. Fit the data using 4th order polynomial regression model (matrix form, gradient descent method).
Plot the training data with the fitted line. Using the optimized weights, please calculated Root Mean Squared Error (RMSE) of training and testing data.
Note: Do not forget feature normalization.

2-4. Fit the data using 16th order polynomial regression model (matrix form, gradient descent method).
Plot the training data with the fitted line. Using the optimized weights, please calculated Root Mean Squared Error (RMSE) of training and testing data.
Note: Do not forget feature normalization.

2-5. Fit the data using 16th order polynomial regression model with ridge (L2 penalty) regularization.
You need to try at least 3 different L2 penalty (for example, $\lambda = 0.1, 1, 10$). Plot the training and validation data with the fitted line. Search optimum L2 penalty based on Root Mean Squared Error (RMSE) of validation data. Print optimized weight coefficients. Plot weight coefficients with L2 penalty. Print Root Mean Squared Error (RMSE) for training/validation/test data.
Note: Do not forget feature normalization.

2-6. Fit the data using 16th order polynomial regression model with scikit-learn Ridge model.
You need to try at least 3 different L2 penalty (for example, $\lambda = 0.1, 1, 10$). Plot the data with the fitted line. Print optimized weight coefficients. Plot weight coefficients with L2 penalty. Print Root Mean Squared Error (RMSE)

2-7. Fit the data using 16th order polynomial regression model with scikit-learn Lasso model.
You need to try at least 3 different L1 penalty (for example, $\lambda = 0.1, 1, 10$). Plot the data with the fitted line. Print optimized weight coefficients. Plot weight coefficients with L2 penalty. Print Root Mean Squared Error (RMSE)

3 (40pts). Regularization with Tensorflow

Using Jupyter notebook, load the data (ex2data3.csv).
This is California housing dataset. The original database is available from http://lib.stat.cmu.edu
The data contains 20,640 observations on 9 variables. This dataset contains the average house value as target variable and the following input variables (features): average income, housing average age, average rooms, average bedrooms, population, average occupation, latitude, and longitude
(R. Kelley and Ronald Barry, Sparse Spatial Autoregressions,\ Statistics and Probability Letters, 33 (1997) 291-297) .

You need to split the data into training/validation/testing data set as 60% / 20% / 20%.
Please use np.random.seed(1) to have consistent data for evaluation.

3-1. Fit the training data using regression model with ridge (L2 penalty) regularization with scikit-learn Ridge model. You need to try at least 3 different L2 penalty (for example, $\lambda = 0.1, 1, 10$). Search optimum L2 penalty based on Root Mean Squared Error (RMSE) of validation data. Print optimized weight coefficients. Plot weight coefficients with L2 penalty. Print Root Mean Squared Error (RMSE) for training/validation/test data.
Note: Do not forget feature normalization

3-2. Fit the training data using regression model with lasso (L1 penalty) regularization with scikit-learn Lasso model. You need to try at least 3 different L1 penalty (for example, $\lambda = 0.1, 1, 10$). Search optimum L1 penalty based on Root Mean Squared Error (RMSE) of validation data. Print optimized weight coefficients. Plot weight coefficients with L2 penalty. Print Root Mean Squared Error (RMSE) for training/validation/test data.
Note: Do not forget feature normalization

3-3. Fit the training data using regression model with ridge (L2 penalty) regularization using TensorFlow.
You need to make gradient descent method instead of open source algorithm.
You need to try at least 3 different L2 penalty (for example, $\lambda = 0.1, 1, 10$). Search optimum L2 penalty based on Root Mean Squared Error (RMSE) of validation data. Print optimized weight coefficients. Plot weight coefficients with L2 penalty. Print Root Mean Squared Error (RMSE) for training/validation/test data.
Note: Do not forget feature normalization.

3-4. Fit the training data using regression model with lasso (L1 penalty) regularization using TensorFlow.
You need to make gradient descent method instead of open source algorithm.
You need to try at least 3 different L1 penalty (for example, $\lambda = 0.1, 1, 10$). Search optimum L1 penalty based on Root Mean Squared Error (RMSE) of validation data. Print optimized weight coefficients. Plot weight coefficients with L2 penalty. Print Root Mean Squared Error (RMSE) for training/validation/test data.
Note: Do not forget feature normalization.