Assignment 3– Regression, Probability Theory, and Optimization

Assignment overview. This assignment is designed to introduce you the gradient descent regression with regularization and probability theory. This Assignment requires you to load a cleaned version of the dataset ***House Sales***[1], learn and predict using your own implemented linear models, and evaluate models. Follow the steps as indicated and complete the tasks. You are expected to figure out details of syntax by consulting Python's Help. Since the material builds from one question to the next, it will be easiest to do them in order. Please answer each question by copying and commenting as needed on the material that you produce in the IPython console as well as all scripts you are asked to create, or use Jupyter Notebook and download it as a Python file.

The second part of the assignment is a practice with probability theory. You are asked to write a program to plot a histogram for a specific probabilistic process that we can model by drawing samples from a specific distribution. Graduate students will also be asked to provide some analytic solutions.

Submission. Create a folder called ML_Assignment3 and put all the files inside the folder. Compress this folder to create either ML_Assignment3.zip or ML_Assignment3.rar. Submit this compressed folder as your assignment submission on Brightspace.

Submission deadline. Thursday, 28 Sep, 10:00 pm.

Late submission policy. If submitted after the due date, the penalty will be 10% per day.

Academic Integrity. Dalhousie academic integrity policy applies to all submissions in this course. You are expected to submit your own work. Please refer to and understand the academic integrity policy, available at https://www.dal.ca/academicintegrity

Python: We will be using Python for the programming exercises based on scientific Python libraries like

- **numpy** - mainly useful for its *N*-dimensional array objects.
- **matplotlib** - 2D plotting library producing publication quality figures.
- **pandas** - Python data analysis library, including structures such as data-frames

If you have a question: Teaching Assistants (TAs) will be present during the labs to help you with any questions you may have. If you still have questions, feel free to email me at tt@cs.dal.ca.

Questions:

1. **[60 marks, 30 marks for Grads]** This Assignment requires you to write a Python script file called sol1.py. You are not allowed to use any Python public libraries related to regression and metrics. You can compare the results of your program sklearn, numpy, or scipy linear models, but the whole exercise is to write the algorithm yourself.

[1] This dataset contains house sale prices for King County, which includes Seattle. It includes homes sold between May 2014 and May 2015.

1.1. **[5 marks, 3 marks for Grads]** Write a Python script file called sol1_1.py to load the ***House sales*** dataset from the houses.csv file and place them in a data-frame `df`. (Hint: You can use `pandas.read_csv` function. New in pandas? Click here). Then generate and show various statistic summary using `pandas.DataFrame.describe` method. **What** is pandas dataframe? Using `pandas.DataFrame` methods split the dataset into target value `Y` (price) and feature matrix `X` (all feature columns). In addition, extract the `sqft_living` column into a feature vector name `X_1`.

1.2. **[15 marks, 7 marks for Grads]** Write a function named `linear_regression` to implement Linear Regression **without using public libraries related to regression**. The inputs of this function should be predictor values (`X` or `X_1`), a target value (`Y`), a learning rate (`lr`), and the number of iterations (`repetition`). The function must build a linear model using gradient descent and output the model (`params`) and loss values per iteration (`loss`). Set the iteration to 10000 and calculate and show the *mean squared error (MSE)* for the models obtained from both `X` and `X_1` predictors (hint: you might write another function named `predict` to predict the values based on `X` or `X_1` and `params`) and plot the learning curve (`loss`) for both models in one figure (hint: use log scaling plot). Try different learning rates (10, 1, 0.1, 0.01, and 0.001) and compare and show the results.

1.3. **[10 marks, 5 marks for Grads]** Visualize the best-obtained model for `X_1` using a scatter plot to show price vs area and plot the linear model. Then, visualize the best-obtained model for all features (`X`) using a scatter plot to show the predicted vs actual target values. **Does** the scatter plot create a linear line? **Why**?

1.4. **[10 marks, 5 marks for Grads]** Modify the `linear_regression` function in a way that applies Ridge regression and LASSO, and name them `linear_regression_Ridge` and `linear_regression_LASSO` respectively. Then repeat the assignments 1.2 and 1.3. You can thereby use a fixed learning rate that you find appropriate, but you should try different values for the regularization penalty alpha.

1.5. **[10 marks, 5 marks for Grads]** Use `linear_regression_LASSO` and write a function named `linear_regression_LASSO_momentum` in which you add a momentum term. Try different momenta (0.1, 0.5, 0.7, and 0.9) and plot the learning curves with and without momentum for a fixed learning rate.

1.6. **[10 marks, 5 marks for Grads]** Modify the `linear_regression_LASSO_momentum` function in a way that it fits the feature vector `X_1` with a polynomial of order 2 and name it as `polynomial_regression_optimized.` Calculate the MSE, plot the learning curve and show the quadratic model on the scatter plot (price vs area).

2. **[40 marks, 25 marks for Grads]** This Assignment requires you to write a Python script file called sol2.py.

2.1. **[10 marks, 5 marks for Grads]** Write a program that rolls two dice (randomly selects a number between 1 and 6 inclusive for each die). Repeat rolling dice 20 times. For each trial, you should add the two numbers that appear on each die and save it in a vector. Plot a histogram of the values you have gathered in the vector. Based on this histogram, **what** are the estimates for the probabilities of each number?

2.2. **[10 marks, 5 marks for Grads]** Run your code for 1000 times. **What** is your estimation now?

2.3. **[20 marks, 10 marks for Grads]** Change your program and assume that you have one fake die and one correct die. The fake die has 0 instead of 2. This means when you roll this defect die, a random number should be chosen among (1,0,3,4,5, and 6). Plot histogram of the values you have gathered in the vector. **Calculate** the probability of getting a 7 as the sum of the numbers appeared on the dice. **What** is the probability of getting 3?

2.4. **Graduate students only [5 marks]:** Try the same problem with seven dice and plot the distribution together with a Gaussian fit. **Report** the mean and variance.

3. **Graduate students only [25 marks]:** (From Thrun, Burgard and Fox, Probabilistic Robotics) A robot uses a sensor that can measure ranges from 0m to 3m. For simplicity, assume that the actual ranges are distributed uniformly in this interval. Unfortunately, the sensors can be faulty. When the sensor is faulty it constantly outputs a range below 1m, regardless of the actual range in the sensor's measurement cone. We know that the prior probability for a sensor to be faulty is p = 0.01. Suppose the robot queries its sensors N times, and every single time the measurement value is below 1m. What is the posterior probability of a sensor fault, for N = 1, 2, ..., 10? Formulate the corresponding probabilistic model.

4. **Graduate students only [20 marks]:** Theoretical limit of Gaussian classification example. We followed an example of classifications with two Gaussian classes in section 2.4. In this assignment, you should calculate analytically the theoretical limit of the optimal accuracy for the parameters used in the printed program. Provide your answer with a brief outline of the calculation.