CS 434: Implementation assignment 1

Due Saturday April 11th 11:59PM, 2020

General instructions

1. The assignment should be implemented in Python. You should make sure that your code can be run on the flip server.
2. You can work in team of up to 3 people.
3. You need to submit your source code (self contained, well documented and with clear instruction for how to run) and a report on canvas. In your submission, please clearly indicate your team members' information.
4. Be sure to answer all the questions in your report. Your report should be typed, submitted in the pdf format. You will be graded based on both your code as well as the report. In particular, the clarity and quality of the report will be worth 10 points. So please write your report in clear and concise manner. Clearly label your figures, legends, and tables.

1 Linear regression

Data You will use the Boston Housing dataset of the housing prices in Boston suburbs. The goal is to predict the median value of housing of an area (in thousands) based on 13 attributes describing the area (e.g., crime rate, accessibility etc). The file housing_desc.txt describes the data. Data is divided into two sets: (1) a training set housing_train.txt for learning the model, and (2) a testing set housing_test.txt for evaluating the performance of the learned model. Your task is to implement linear regression and explore some variations with it on this data.

1. (10 pts) Load the training data into the corresponding X and Y matrices, where X stores the features and Y stores the desired outputs. The rows of X and Y correspond to the examples and the columns of X correspond to the features. Introduce the dummy variable to X by adding an extra column of ones to X (You can make this extra column to be the first column. Changing the position of the added column will only change the order of the learned weight and does not matter in practice). Compute the optimal weight vector $\mathbf{w}$ using $\mathbf{w} = (X^T X)^{-1} X^T Y$. Feel free to use existing numerical packages (e.g., numpy) to perform the computation. Report the learned weight vector.
2. (10 pts) Apply the learned model to make predictions for the training and testing data respectively and compute for each case the average squared error(ASE), defined by $1/n \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$, which is the sum of squared error normalized by n, the total number of examples in the data. Report the training and testing ASEs respectively. Which one is larger? Is it consistent with your expectation?

   Write your code so that you get the results for questions 1 and 2 using the following command:

   python q1_2.py housing_train.txt housing_test.txt

   The output should include:

   - the learned weight vector
   - ASE over the training data
   - ASE over the testing data

3. (10 pts) Remove the dummy variable (the column of ones) from X, repeat 1 and 2. How does this change influence the ASE on the training and testing data? Provide an explanation for this influence.

   Write your code so that you get the results for question 3 using the following command:
   python q1_3.py housing_train.txt housing_test.txt

   The output should include:

   - the learned weight vector
   - ASE over the training data
   - ASE over the testing data

4. (20 pts) Modify the data by adding additional random features. You will do this to both training and testing data. In particular, generate 20 random features by sampling from a standard normal distribution. Incrementally add the generated random features to your data, 2 at a time. So we will create 20 new train/test datasets, each with d of random features, where $d = 2, 4, ..., 20$. For each version, learn the optimal linear regression model (i.e., the optimal weight vector) and compute its resulting training and testing ASEs. Plot the training and testing ASEs as a function of d. What trends do you observe for training and testing ASEs respectively? In general, how do you expect adding more features to influence the training ASE? How about testing ASE? Why?

   Write your code so that you get the results for question 4 using the following command:
   python q1_4.py housing_train.txt housing_test.txt

   The output should include:

   - plot of the training ASE (y-axis) as a function of d (x-axis)
   - plot of the testing ASE (y-axis) as a function of d (x-axis)

2 Logistic regression with regularization (to come)