COMP-551 : Applied Machine Learning | Given on: Sep 17, 09:00 pm

Programming Assignment #1 | Due on : Oct 01, 10:00 pm

- This is an individual assignment. However, you are allowed to discuss the problems with other students in the class. But you should write your own code and report.
- If you have any discussion with others, you should acknowledge the discussion in your report by mentioning their name.
- You have to submit the `pdf` copy of the report on gradescope before the deadline. If you handwrite your solutions, you need to scan the pages, merge them to a single `pdf` file and submit. Mark page 1 for outline items *'Late Submission'* and *'Verbosity'*.
- When asked to report the parameters, save the corresponding parameters in a text file (`.txt`) with the following template for the name: `Assignment1_<McGill ID>_<section>_<sub-question>_<sub-sub-question>`. Submit all the text files and codes compressed to a single file `<your-mcgill-ID>.zip` on **MyCourses**.
  Note: gradescope doesn't accept .zip.
- Be precise with your explanations in the report. Unnecessary verbosity will be penalized.
- After the deadline, you have 3 days to submit your assignment with 30% penalty.
- You are free to use libraries with general utilities, such as matplotlib, numpy and scipy for python. However, you should implement the algorithms yourself, which means **you should not use pre-existing implementations of the algorithms as found in SciKit learn, Tensorflow, etc.**!
- If you have questions regarding the assignment, you can ask for clarifications in the corresponding reddit thread.

---

1 Sampling

A grad student's daily routine is defined as a multinomial distribution, p, over the set of following activities:

- Movies: 0.2
- COMP-551: 0.4
- Playing: 0.1
- Studying: 0.3

1. Every morning, he/she wakes up and randomly samples an activity from this distribution and do that for the rest of the day. Provided that you can only sample from uniform distribution over (0,1), write a pseudocode to sample from the given multinomial distribution.

2. Implement your sampling algorithm and use it to sample his/her routine for 100 days. Report the fraction of days spent in each activity. Now use it to sample for 1000 days. Report the fraction of days spent in each activity. Compare these fractions to the underlying multinomial distribution.

2 Model Selection

You have to use Dataset-1 for this experiment. Dataset-1 consists of train, validation, and test files. The input is a real valued scalar and the output is also a real valued scalar. The dataset is generated from an n-degree polynomial and a small Gaussian noise is added to the target.

1. Fit a 20-degree polynomial to the data.

   (a) Report the training and validation MSE (Mean-Square Error). Do not use any regularization.

   (b) Visualize the fit.

   (c) Is the model overfitting or underfitting ? Why?

2. Now add L2 regularization to your model. Vary the value of λ from 0 to 1.

   (a) For different values of λ, plot the training MSE and the validation MSE.

   (b) Pick the best value of λ and report the test performance for the corresponding model.

   (c) Also visualize the fit for the chosen model.

   (d) Is the model overfitting or underfitting ? Why?

3. What do you think is the degree of the source polynomial? Can you infer that from the visualization produced in the previous question?

3 Gradient Descent for Regression

You have to use Dataset-2 for this experiment. Dataset-2 consists of train, validation, and test files. The input is a real valued scalar and the output is also a real valued scalar.

1. Fit a linear regression model to this dataset by using stochastic gradient descent (one example at a time).

   (a) Use the step size of 1e-6. Compute the MSE on validation set for every epoch.

(b) Plot the learning curve i.e. training and validation MSE for every epoch. [Note: Plot the learning curve until the learning saturates.]

2. Try different step sizes and choose the best step size by using validation data.

   (a) Report in a table the validation performance with different step-sizes.

   (b) Report the test MSE of the chosen model.

3. Visualize the fit for every epoch and report 5 visualizations chosen at random to illustrate how the regression fit evolves during the training process.

4 Real life dataset

For this question, you will use the Communities and Crime Data Set from the UCI repository (`http://archive.ics.uci.edu/ml/datasets/Communities+and+Crime`).

1. This is a real-life data set and as such would not have the nice properties that we expect. Your first job is to make this dataset usable, by filling in all the missing values.

   (a) Use the sample mean of each column to fill in the missing attribute. Is this is a good choice? Explain why or why not.

   (b) What else might you use to fill in the missing attributes?

   (c) If you have a better method, describe it, and use it for filling in the missing data. Does your method provide improvement in performance ? Explain why this method is better.

   (d) Turn in the completed data set.

2. Fit the above data using linear regression. Report the 5-fold cross-validation error: The MSE of the best fit achieved on test data, averaged over 5 different 80-20 splits, along with the parameters learnt for each of the five models.

   [**Note:** The split should be done at random.]

3. Use Ridge-regression on the above data. Repeat the experiment for different values of λ. Report the MSE for each value, on test data, averaged over 5 different 80-20 splits, along with the parameters learnt.

   (a) Which value of λ gives the best fit ? Show the comparison in a plot. [x-axis: λ, y-axis: Average Test MSE.]

   (b) Is it possible to use the information obtained during this experiment for feature selection? If so, explain how?

   (c) Show the results of the best fit you achieve with a reduced set of features?

   (d) How different is the performance of the model with reduced features compared to the model using all the features? Comment about the difference.

Instructions on how to use 80-20 splits

1. Make 5 different 80-20 splits in the data and name them as $CandC{-}train\,\langle num\rangle\,.csv$ and $CandC-test\,\langle num\rangle\,.csv$.
2. For all 5 datasets that you have generated, learn a regression model using the 80% data and test it using 20% data.
3. Report the average MSE over these 5 different runs.

Instruction for code submission

1. Submit a single zipped folder with your McGill id as the name of the folder. For example if your McGill ID is 12345678, then the submission should be 12345678.zip.
2. You can only use Python 3 and you must submit your solution as a jupyter notebook.
3. Make sure all the data files needed to run your code is within the folder and loaded with relative path. We should be able to run your code without making any modifications.

Instruction for report submission

1. You report should be brief and to the point. When asked for comments, your comment should not be more than 3-4 lines.
2. Report all the visualizations (learning curves, regression fit).
3. Do not include your code in the report!