CSE 4/574: Introduction to Machine Learning
Summer 2021

Instructor: Nitin Kulkarni

Assignment 1 - Linear Regression

Checkpoint: July 23, Fri, 11:59pm

Due Date: July 30, Fri, 11:59pm

1 Assignment Overview

The goal of the assignment is to work with linear models for regression. In the first part of the assignment we will implement linear regression using the Ordinary Least Squares equation we derived in class. In the second part of the assignment we will implement linear regression using gradient descent methods. The purpose of this assignment is to understand how linear models for regression work and the benefits that gradient descent based methods can provide.

Note: You are not allowed to use any Python libraries or built-in functions that directly perform regression or give you the RMSE. Use of ML libraries like sklearn will result in 0 points for the assignment.

Dataset

You have been given a medical insurance dataset. There are 1338 samples in the dataset. The features include information about an individual such as their age, sex, number of children, whether they smoke and the region they are based in. The target label is the insurance charge.

You will figure out which features you want to use and split the dataset into a train and test dataset such that the train dataset contains 80% of the data and test dataset contains 20% of the data.

Part 1 [40 points] - Implementing Linear Regression using Ordinary Least Squares Estimate

Implement linear regression using ordinary least squares method to do direct minimization of the squared loss function.

$$J(\boldsymbol{w}) = \sum_{i=1}^{N} (y_i - \boldsymbol{w}^\top x_i)^2$$

Plan of Work

1. Calculate the weight vector ($\boldsymbol{w}$) using the Ordinary Least Squares Estimate.
2. Using the weight vector ($\boldsymbol{w}$) you computed get the predictions for the test data.
3. Calculate the RMSE for the test dataset.
4. Show a plot comparing the predictions vs the actual test data targets.

In your report for Part 1:

1. Discuss the benefits/drawbacks of using OLS estimate for computing the weights.
2. Mention the weight vector that you calculated.
3. Mention the RMSE value for the test data.
4. Show the plot comparing the predictions vs the actual test data.

Part 2 [60 points] - Implementing Linear Regression using Gradient Descent Methods

2.1 Implementing Gradient Descent

Implement linear regression from scratch using gradient descent algorithm.

2.2 Implementing Stochastic Gradient Descent

Implement linear regression from scratch using stochastic gradient descent algorithm.

Plan of Work

1. Initialize the weight vectors with random values then update them using the following equation: $\boldsymbol{w} = \boldsymbol{w} - \alpha \nabla J(\boldsymbol{w})$ ($\alpha =$ learning rate)
2. Calculate the weight vector ($\boldsymbol{w}$) using gradient descent and stochastic gradient descent.
3. Using the weight vector ($\boldsymbol{w}$) you computed get the predictions for the test data.
4. Calculate the RMSE for the test dataset.
5. Show a plot comparing the predictions vs the actual test data targets.

In your report for Part 2:

1. Discuss the benefits/drawbacks of using gradient descent based methods over OLS estimate.
2. Provide a comparison of the three different methods and their results i.e., OLS estimate, gradient descent and stochastic gradient descent.
3. Mention the weight vectors that you calculated with gradient descent and stochastic gradient descent.
4. Mention the RMSE value for the test data for gradient descent and stochastic gradient descent.
5. Show the plot comparing the predictions vs the actual test data for gradient descent and stochastic gradient descent.

Extra Points [max + 5 points]

- **Implement Ridge Regression [5 points]**
  Implement ridge regression from scratch using direct minimization.

Plan of Work

1. Calculate the weight vector ($\boldsymbol{w}$) by direct minimization.
2. Using the weight vector ($\boldsymbol{w}$) you computed get the predictions for the test data.
3. Calculate the RMSE for the test dataset.
4. Show a plot comparing the predictions vs the actual test data targets.

In your report, mention the weight vectors that you calculated. Mention the RMSE value for the test data. Show the plot comparing the predictions vs the actual test data.

3 Deliverables

Submit your work using UBLearns group in both cases if you work individually or in a team of two. There are two parts in your submission:

3.1 Report

The report should be delivered as a separate pdf file, and it is recommended for you to use the NIPS template to structure your report. You may include comments in the Jupyter Notebook, however you will need to duplicate the results in the separate pdf file. For the final submission, combine the reports for both Part 1 and Part 2 into one file.

3.2 Code

Python is the only code accepted for this project. You can submit the code in Jupyter Notebook (.ipynb) or Python script (.py). You can submit multiple files, but they all need to have a clear name. After executing command python main.py in the first level directory or Jupyter Notebook, it should generate all the results and plots you used in your report and should be able to be printed out in a clear manner. Additionally you can submit the trained parameters, so that the grader can fully replicate your results. For the final submission you can combine the code from both parts into one.

4 References

- NIPS Styles (docx, tex)
- Overleaf (LaTex based online document generator) - a free tool for creating professional reports
- Lecture slides

5 Checkpoint Submission [Due date: July 23]

Complete Part 1 and submit the code and draft report. To submit your work, add your pdf, ipynb/python script to zip file with UBIT $TEAMMATE1\_TEAMMATE2\_assignment1\_checkpoint.zip$ (e.g. $nitinvis\_soumyyak\_assignment1\_checkpoint.zip$ and upload it to UBlearns (Assignments section). Checkpoint will be evaluated after the final submission.

6 Final Submission [Due date: July 30]

Add your combined pdf and ipynb/python script for Part 1 and Part 2 to a zip file $TEAMMATE1\_TEAMMATE2\_assignment1\_final.zip$ (e.g. $nitinvis\_soumyyak\_assignment1\_final.zip$) and upload it to UBlearns using group submission (Assignments section).

7 Important Information

This assignment can be completed in groups of two or individually. The standing policy of the Department is that all students involved in any academic integrity violation (e.g. plagiarism in any way, shape, or form) will receive an F grade for the course. The catalog describes plagiarism as "Copying or receiving material from any source and submitting that material as one's own, without acknowledging and citing the particular debts to the source, or in any other manner representing the work of another as one's own.". Updating the hyperparameters or modifying the existing code is not part of the assignment's requirements and will result in a zero. Please refer to the UB Academic Integrity Policy.

8 Late Days Policy

You can use up to 3 late days throughout the course toward any assignments' checkpoint or final submission. You don't have to inform the instructor, as the late submission will be tracked in UBlearns. If you work in teams the late days used will be subtracted from both partners. E.g. you have 3 late days and your partner has 2 days left. If you submit one day after the due date, you will have 2 days and your partner will have 1 days left.

9 Important Dates

July 23, Friday 11:59pm - Checkpoint is Due

July 30, Friday, 11:59pm - Assignment 1 is Due