Lab Course Machine Learning
Exercise 1

Prof. Dr. Dr. Lars Schmidt-Thieme,
Mofassir ul Islam Arif
Information Systems and Machine Learning Lab
University of Hildesheim
Submission: 01.11.2019 LearnWeb 3115

October 26, 2019

Instructions

Please following these instructions for solving and submitting the exercise sheet.

1. You should submit a zip or a tar file containing two things a) python scripts and b) a pdf document.
2. In the pdf document you will explain your approach (i.e. how you solved a given problem), and present your results in form of graphs and tables.
3. The submission should be made before the deadline, only through learnweb.
4. Unless explicitly mentioned, you are not allowed to use scikit, sklearn or any other library for solve any part. All implementations must be done yourself.

1 Exercise Sheet 1

1.1 Python and Numpy (8 Points)

This task puts you in the position that I end up at the end of every semester. Which is, grading your work and issuing the grades. In this task you are required to use the 'Grades.csv' File that has been provided on learnweb.

- **Grading Program**:
- Read the data from the csv.
- For each student,
  - Compute the sum for all subjects for each student.

GRADING SYSTEM

% Range	Grade	# of students
96 - 100	A+	1
90 - 95	A	0
86 - 89	A-	0
80 - 85	B+	1
76 - 79	B	1
70 - 75	B-	0
66 - 69	C+	0
60 - 65	C	0
56 - 59	D	0
0 - 55	F	0

Figure 1: Grading Rubric

  - Compute the average of the point for each student. (total points are 500).
  - Compute the standard deviation of point for each student.
  - Plot the average points for all the students (in one figure).
  - For each student assign a grade based on the following rubric.
  - Plot the histogram of the final grades.

- **Matrix Multiplication**: Using numpy you are required to use numpy for operation on matrices. Create a matrix A of dimensions $n \times m$, where n = 100 and m = 20. Initialize Matrix A. Create a vector v of dimension $m \times 1$. Initialize the matrix with random values and vector with normal distribution using $\mu = 5$ and $\sigma = 0.01$. Perform following operation on them
  - Iterative multiply (element-wise) each row of matrix A with vector v and sum the result of each iteration in another vector c
  - Find the mean and standard deviation of the new vector c
  - Plot histogram of vector c using 5 bins
  - NOTE: You are no allowed to use ANY BUILTIN FUNCTIONS FOR CALCULATING THE MATRIX MULTIPLICATION. Implement it yourself.

1.2 Linear Regression through exact form. (12 Points)

In this exercise, you will implement linear regression that was introduced in the introduction Machine Learning Lecture. The method we are implementing here today is for a very basic univariate linear regression. We will approach it using two examples.

1.2.1 OLS using an artificial dataset 6 Points

- Generate some simple data. i.e. a matrix A with dimensions 100×2. Initialize it with normal distribution $\mu = 2$ and $\sigma = [1]$ (0 points)

- Implement LEARN-SIMPLE-LINREG algorithm and train it using matrix A to learn values of β_0 and β_1 (1 point)
- Implement PREDICT-SIMPLE-LINREG and calculate the points for each training example in matrix A. (1 point)
- Plot the training data (use plt.scatter) and your predicted line (use plt.plot). (1 point)
- Put β_0 to zero and rerun the program to generate the predicted line. Comment on the change you see for the varying values of σ (1 point)
- Put β_1 to zero and rerun the program to generate the predicted line. Comment on the change you see for the varying values of σ (1 point)
- Use numpy.linalg lstsq to replace step 2 for learning values of β_0 and β_1. Explain the difference between your values and the values from numpy.linalg lstsq.(1 point)

1.2.2 OLS using a Real dataset 6 Points

DATA:

A : Download the MPG data from this **link**. (Use the **auto-mpg.data** file)[1]. This dataset was taken from the StatLib library which is maintained at Carnegie Mellon University. The dataset was used in the 1983 American Statistical Association Exposition. Read the dataset description to understand which columns exist in the dataset and how to use them.

- Load the dataset and split it into our uni-variate case. use 'Displacement' as the independent variable and 'MPG' as the dependant variable.
- Implement LEARN-SIMPLE-LINREG algorithm and train it using matrix A to learn values of β_0 and β_1. (1 point)
- Implement PREDICT-SIMPLE-LINREG and calculate the points for each training example in matrix A. (1 point)
- Plot the training data (use plt.scatter) and your predicted line (use plt.plot). (1 point)
- Repeat the above but change the indepedant variable to ["horsepower","weight","acceleration"], Dependant variable stays as MPG.
- Comment on the behavior you notice for each of the independent variable.
- use scikit learn to implement OLS Linear Model. (2 point)
- Plot the training data (use plt.scatter) and your predicted line (use plt.plot). (1 point)

References

[1] Dheeru Dua and Casey Graff. *UCI Machine Learning Repository*. 2017. URL: http://archive.ics.uci.edu/ml.