Stony Brook University
CSE512 – Machine Learning – Spring 21
Homework 1, Due: Feb 23 at midnight 11:59pm

This homework contains three questions. The last two questions require programming. The maximum number of points is 100 points. For this homework, you should review some material on probability and linear regression.

1 Question 1 – Probability (30 points)

Let X_1 and X_2 be independent continuous random variables uniformly distributed from 0 to 1. Let $X = \max(X_1, 2X_2)$. Compute:

1. The expectation $E(X)$
2. The variance $Var(X)$
3. The covariance: $Cov(X, X_1)$.

2 Question 2 – Feature examination (25 points)

Analyzing the range and distribution of feature values are important in machine learning. In this question, you are asked to write a function to approximate the distribution of each individual features by an univariate Gaussian distribution.

2.1 Question 2.1 (10 points)

Write a Python file hw1.py that contains a function with the following signature:

$$[mu0, var0, mu1, var1] = get\_mean\_and\_variance(\mathbf{X}, \mathbf{y})$$

where

Inputs:

- $\mathbf{X}$: a two dimensional Numpy array of size $n \times d$, where n is the number of data points, and d the dimension of the feature vectors.
- $\mathbf{y}$: a Numpy vector of length n. $\mathbf{y}[i]$ is a binary label corresponding to the data point $\mathbf{X}[i, :]$.

Outputs:

- $mu0$: a Numpy vector of length d, $mu0[j]$ is the mean of $\mathbf{X}[i, j]$ for all i where $\mathbf{y}[i] = 0$. Basically, $mu0[j]$ is the mean of the j^{th} feature for all the negative data points.
- $var0$: a Numpy vector of length d, $var0[j]$ is the variance of $\mathbf{X}[i, j]$ for all i where $\mathbf{y}[i] = 0$.
- $mu1$: a Numpy vector of length d, $mu1[j]$ is the mean of $\mathbf{X}[i, j]$ for all i where $\mathbf{y}[i] = 1$.
- $var1$: a Numpy vector of length d, $var1[j]$ is the variance of $\mathbf{X}[i, j]$ for all i where $\mathbf{y}[i] = 1$.

2.2 Question 2.2 (15 points)

For this question, you will use the data provided in the file $covid19\_metadata.csv$. This is a subset of the COVID-19 image data collection (`github.com/ieee8023/covid-chestxray-dataset`).

Inspect your data carefully: Each row corresponds to a patient, which was suspected positive for COVID-19. The first column corresponds to their age (continuous value) and the second one to their gender (F for female or M for male). The last column includes information if they recovered (Y) or not (N).

The first step is to load the data and pre-process them. (Tip: To load the data from the provided csv file, you can use numpy.genfromtxt or csv.reader). In this case, the feature matrix $\mathbf{X}$ should include the first two columns, namely the age and gender of the patients. You will need to convert their gender to integer values (e.g. 1 for female and 0 for male). The labels vector $\mathbf{y}$ is the last column. You will need to convert it to integer values, namely 1 for Y (survived) and 0 for N (not survived).

Then, run the function of the Question 2.1 on this data.

(a) (5 points) Report the values of $mu0$, $var0$, $mu1$, $var1$.

(b) (5 points) For each feature j, plot the Gaussian distribution with mean $mu0[j]$ and variance $var0[j]$ in black color. On the same graph, plot the Gaussian distribution with mean $mu1[j]$ and variance $var1[j]$ in blue. You can use Python packages matplotlib and scipy.

(c) (5 points) Is it a good idea to approximate gender by a Gaussian distribution? Why or why not?

3 Question 3 – Linear Regression (45 points)

In this question, you will use Linear Regression to forecast the number of COVID-19 deaths for the next day based on the numbers of the total cases and deaths in the last seven days. Suppose you have collected the data for the past n days $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$, where x_i and y_i are the number of the aggregated cases and deaths for day i respectively. You hypothesize that there is a linear relationship between the number of deaths for day t based on the data from the previous days, and you derive a model that predict the number of deaths for day t based on:

$$\hat{y}_t = \sum_{i=1}^{7} w_i x_{t-i} + \sum_{i=1}^{7} w_{7+i} y_{t-i} + b, \text{ for } 8 \leq t \leq n. \tag{1}$$

3.1 Question 3.1 (25 points)

Write a Python file hw1.py that contains a function with the following signature:

$$[\mathbf{w}, b] = learn\_reg\_params(\mathbf{x}, \mathbf{y})$$

where

Inputs:

- $\mathbf{x}$: a Numpy vector of length n, where n is the number of days
- $\mathbf{y}$: a Numpy vector of length n

Outputs:

- $\mathbf{w}$: a Numpy vector of length 14
- b: a scalar value for the intercept of Linear Regression model.

Tip: you can use $sklearn.linear\_model.LinearRegression$ from the scikit-learn library.

3.2 Question 3.2 (20 points)

The file $covid19\_time\_series.csv$ contains the dataset that you will use. The first row is the csv header. The second row contains the aggregated counts of the confirmed positive cases in the US. The third row contains the aggregated counts of deaths in the US. These data are a subset of "JHU CSSE COVID-19 Data" gathered at Johns Hopkins University (`https://github.com/CSSEGISandData/COVID-19`). The first step is to load the data into $\mathbf{x}$ and $\mathbf{y}$. Use the function that you wrote in Question 3.1 to calculate the parameters $\mathbf{w}$ and b of Linear Regression model.

(a) (5 points) Report the learned parameters: the weights $\mathbf{w}$ and the intercept term b.

(b) (5 points) Visualize the actual and predicted death values y_t and $\hat{y}_t$ (for $8 \leq t \leq n$). Display y_t as a function of t and $\hat{y}_t$ as a function of t on the same graph. You can use the library matplotlib.pyplot to plot.

(c) (5 points) Use a Gaussian to approximate the distribution of the errors $y_t - \hat{y}_t$ (for $8 \leq t \leq n$). Report the mean and variance of this Gaussian.

(d) (5 points) Use matplotlib.pyplot.hist to plot the distribution of $y_t - \hat{y}_t$ (for $8 \leq t \leq n$). On the same plot, plot the Gaussian function that approximates this distribution. Is Gaussian a good approximation for the distribution of the errors?

4 What to submit

You will need to submit both your code and your answers to questions on Blackboard. Put the answer file and your python code in a folder named: SUBID_FirstName_LastName (e.g., *10947XXXX_Barack_Obama*). Zip this folder and submit the zip file on Blackboard. Your submission must be a zip file,
i.e, SUBID_FirstName_LastName.zip.

The answer file should be named: hw1-answers.pdf. You can use Latex if you wish, but it is not compulsory. The first page of the hw1-answers.pdf should be the filled cover page at the end of this homework. The remaining of the answer file should contain:

1. Answers to Questions 1
2. Answers to Questions 2.2
3. Answers to Questions 3.2

Your Python code must be named hw1.py. It should contains two functions in Question 2.1 and Question 3.1. For automated testing, it should be possible for us to import your functions using 'from hw1 import learn_reg_params, get_mean_and_variance'. You can submit other python/data files if necessary. Your code hw1.py can also include other functions if necessary.

Make sure you follow the instructions carefully. Your will lose points if:

1. You do not attach the cover page. Your answer file is not named hw1-answers.pdf
2. Your functions do not have the exact signatures as instructed
3. Your functions cannot be imported for automatic testing
4. Your functions crash or fail to run

5 Cheating warnings

Don't cheat. You must do the homework yourself, otherwise you won't learn. You cannot ask and discuss with students from previous years. You cannot look up the solution online.

Cover page for answers.pdf
CSE512 Spring 2021 - Machine Learning - Homework 1

Your Name:

Solar ID:

NetID email address:

Names of people whom you discussed the homework with: