Assignment (1)

Artificial Intelligence and Machine Learning (24-787)

Due date: 2019/09/09 @ 11:59 pm EST

Note: In case a problem requires programming, it should be programmed in Python language. In Programming, you should use plain Python language, unless otherwise stated. For example, if the intention of a Problem is familiarity with numpy library, it will be clearly noted in that problem to use numpy.

There are two steps to submitting your assignment:

1. Submit a pdf of the completed iPython notebooks (Jupyter Notebooks) to **Canvas**. If you are enrolled in the course, then you should have already been automatically added to the course on Canvas. To produce a pdf of your work, you can first convert each of the .ipynb files to HTML. To do this, simply run:

ipython nbconvert –to html FILE.ipynb

for each of the notebooks, where FILE.ipynb is the notebook you want to convert. Then you can convert the HTML files to PDFs with your favorite web browser, and then concatenate them all together in your favorite PDF viewer/editor. Submit this final PDF on Canvas, and be sure to tag the questions correctly!

Important: Please make sure that the submitted notebooks have been run and the cell outputs are visible.

2. Submit the FILE.ipynb and its dependencies. If an assignment has theoretical and mathematical derivation, scan your handwritten solution and make a PDF file. We prefer FILE.zip submission for grouping multiple files. The file name (FILE) for naming should be saved as HW-assignmentnumber-problemnumber-andrew-ID. For example for problem 1 of assignment 1, my **FILE= HW-1-1-afariman**

PROBLEM 1

Topic: Linear Regression

a) (Theoretical problem)- 5 points: Find the equation of the regression line that fits the points (0, 4), (1, 7), (2, 9), (3, 12),and (5, 18) by using and solving the normal equations. (You could use Latex or word and transform it into pdf)

b) (Programming problem)- 10 points: Write a program which takes n numbers of (x,y) as input and outputs b_0 and b_1 for the linear regression equation $y = b_0 + b_1 x$ that fits (x,y)s. Use normal equation. And verify the function you write by printing b_0 and b_1 in a) within the Jupyter cell. You can use numpy and math libraries in Python.

c) (Programming problem)- 15 points: Using the dataset provided in this assignment (data.csv) and the program you wrote in part b, calculate b_0 and b_1.

PROBLEM 2

Topic: Python Programming

a) (Programming problem)- 10 points: Write a function called `includes` which accepts a collection, a value, and an optional starting index. The collection can be a string, a list or a dictionary. The function should return `True` if the value exists in the collection when we search starting from the starting index. Otherwise it should return `False`. If the starting index is given, it is where to search from the collection. If the collection is a dictionary, the function searches for a value among values in the dictionary; since objects in dictionaries have no sort order, the starting index is ignored. You can test your code with the examples below.

```
def includes(item,val,start_ind=None):
    pass


includes([1, 2, 3], 1) # True
includes([1, 2, 3], 1, 2) # False
includes({'a':1,'b':2}, 1) # True
includes({'a':1,'b':2}, 'a') # False
includes('abcd', 'b') # True
```

b) (Programming problem)- 15 points: Create a function `moving_average`. When the function is passed a value, the function returns the current average of all previous function calls. Round all answers to the 1st decimal place. You can test your code with the example below to see if it returns the correct values. You may use numpy but do not use `numpy.ma.average`. *Hint: You might need a inner function for this.

```
def moving_average():
    pass

mAvg = moving_average
print(mAvg(10)) #10.0
print(mAvg(11)) #10.5
print(mAvg(12)) #11.0
```

c) (Programming problem)- 15 points: Write a function called `same_frequency` which accepts two numbers and returns `True` if they contain the same frequency of the digits. Otherwise, return `False`. You can test your code with the examples given.

```
def same_frequency(num1,num2):
    pass

same_frequency(551122,221515) # True
same_frequency(321142,3212215) # False
same_frequency(1212, 2211) # True
```

PROBLEM 3

Topic: Maximum Likelihood Estimate (MLE)

a) (Theoretical problem)- 10 points: In a lab testing, the number of cycles to fail in five tests are x_i = 25, 20, 28, 33, 26. In this case, the distribution parameters are λ and ξ, and follows log-normal distribution: ln(x) $\sim$ N(μ,σ^2). Estimate the distribution parameter using MLE.

b) (Theoretical problem)- 10 points:: For a two parameter distribution, the probability density function is:

$$f(x) = \theta\sigma^{\theta}x^{-\theta-1} \tag{1}$$

What is the MLE for σ and θ ?

c) (Theoretical problem)- 10 points: The probability mass function of Poisson distribution is:

$$P(x) = e^{-\lambda}\frac{\lambda^x}{x!} \tag{2}$$

Find the MLE of parameter λ.