Assignment 9

Regression and Classification

Date Due: Thursday March 30, 5:00pm **Total Marks: 13**

General Instructions

- **This assignment is individual work.** You may discuss questions and problems with anyone, but the work you hand in for this assignment must be your own work.
- If you intend to use resources not supplied by the instructor, please request permission prior to starting. **You should not use any resource that substantially evades the learning objectives for this assignment.** You must provide an attribution for any external resource (library, API, etc) you use. If there's any doubt, ask! No harm can come from asking, even if the answer is no.
- Each question indicates what the learning objective is, what to hand in, and how you'll be evaluated.
- **Do not submit folders, or zip files, even if you think it will help.**
- Assignments must be submitted to Canvas

Question 1 (2 points):

Purpose: To think about data for Linear Regression

In this assignment, you will be implementing **linear regression** and running it on a data set. But first, it's useful to take quick look at the data itself.

Both of the provided data sets have the same format. Each line of each data file is a single **labelled sample**, and each sample is comprised of just two numbers. The first number is the **input feature** for that sample, and the second number is the **output value**. Recall that the task for linear regression is to predict the output value based on the input feature, assuming a linear relation between the two.

The first file `easydata.txt` is just some warm-up data that will help you determine if your linear regression is working. Look at the data and determine, by hand and without any program, the **equation of the linear function** that you think would have produced this data.

The second file `gamesite.txt` contains data for a fictional website where players log in to play boardgames. The input feature for this data is the number of **unique user logins** that occurred on a particular day, and the output feature is the number of actual games played on the site during that day. Look at the data and make a prediction about whether you think linear regression will **work well** on this data. No need to write a lot, your first impression is good enough.

What to Hand In

Your data together with your written answer in a file called `a9q1.pdf` (or .doc, .txt).

Evaluation

- 1 mark: The equation for the easy data is correct
- 1 mark: The impression for the game site data shows critical thought

Question 2 (7 points):

Purpose: To understand linear regression

For this question, your task is to implement the training algorithm for **linear regression**.

Implementation

A starter code file has been provided for you. Your task is to implement the `train()` function. The coding here is quite straightforward; the main challenge is understanding the mathematical equations for linear regression well enough to implement it. In particular, consider section 19.6 of AIMA, and the lecture slide notes for "Regression and Classification", especially slide 11.

Run on easy data

Run the regression on `easydata.txt`. You will note that to run linear regression, you must decide on a number of **training steps** to use and on a value for **alpha**, the learning rate. Experiment to find values for these that work well.

Report the values you used for the steps and for alpha, and the **weights that get learned** as a result. If everything is working, you should get weights that are very close to what you calculated by hand for this data set.

Run on gamesite data

Run the regression on `gamesitedata.txt`. Again, find a number of steps and a value for alpha that seem to produce plausible results and report those values.

You will notice that the provided code splits the data into two halves, and then reports results using the first half for training and the second half for testing, and then vice versa. Report these results and then answer the following questions:

- Is there a significant difference between the **weights** that were learned by using the two training sets? If so, why do you think that is?
- Is there a significant difference between the **training error** and the **test error**? If so, what does that tell you?
- Based on your results here, how well would you expect your regression to work on unseen data?

What to Hand In

- Your updated code in a file called `linear_regression.py`
- Your data together with your written answer in a file called `a9q2.pdf` (or .doc, .txt)

Evaluation

- 3 marks: `train()` is correctly implemented
- 1 mark: Results are reported for easy data
- 3 marks: Results are reported and analyzed for gamesite data

Question 3 (4 points):

Purpose: To appreciate the importance of cross-validation

Improve Data Partitioning

In the provided code you were given for linear regression, the `partition_data()` function from `data_manager.py` is very bad. For this question, your task is to improve it by implementing **cross-validation** (see "Machine Learning" lecture slides, slide 18).

In particular you need to do the following:

- Randomize the data before splitting it
- **Generalize** the function by adding a parameter k for the number of sets to use

Rerun the regression

Choose some reasonable value for the number of randomized cross-validation sets to create and rerun the linear regression. Report the **test error** for each of the randomized sets and the overall average test error. Then answer these questions:

- How confident are you in the reported test error now, as compared to your results from the previous question?
- Do you see any evidence of **over-fitting** in your results here?

What to Hand In

- Your updated data partitioning code in a file called `data_manager.py`
- Your data together with your written answer in a file called `a9q3.pdf` (or .doc, .txt)

Evaluation

- 2 marks: the cross validation is correctly implemented
- 2 marks: Results are reported and critically analyzed