Course › Week 3... › Week 3... › Week 3...

Week 3 Project

ACADEMIC HONESTY

As usual, the standard honour code and academic honesty policy applies. We will be using automated **plagiarism detection** software to ensure that only original work is given credit. Submissions isomorphic to (1) those that exist anywhere online, (2) those submitted by your classmates, or (3) those submitted by students in prior semesters, will be detected and considered plagiarism.

INSTRUCTIONS

This assignment has two parts. You should write one function that implements both.

PART 1: In this part you will implement the ℓ_2-regularized least squares linear regression algorithm we have been discussing (ridge regression). Recall from the lectures that this takes the form:

$$w_{RR} = \arg\min_w \|y - Xw\|^2 + \lambda\|w\|^2.$$

Your task will be to write code that takes in data y and X and outputs w_{RR} for an arbitrary value of λ.

PART 2: In the same code, you will also implement the active learning procedure discussed in Lecture 5. For this problem, we will provide you with an arbitrary setting of λ and σ^2 and ask you to provide us with the first 10 locations you would measure from a set $\mathcal{D} = \{x\}$ given a set of measured pairs (y, X). Please look over the slides carefully to remind yourself about the *sequential evolution* of the sets $\mathcal{D}$ and (y,X).

More details about the inputs we provide and the expected outputs are given below.

Sample starter code to read the inputs and write the outputs: Download hw1_regression.py

WHAT YOU NEED TO SUBMIT

You can use either **Python (version 3.6.4)** or **Octave** coding languages to complete this assignment. Octave is a free version of Matlab. Your Matlab code should be able to directly run in Octave, but you should not assume that advanced built-in functions will be available to you in Octave. Unfortunately we will not be supporting other languages in this course.

.

Depending on which language you use, we will execute your program using one of the following two commands.

.

Either

```
$ python3 hw1_regression.py lambda sigma2 X_train.csv
y_train.csv X_test.csv
```

Or

```
$ octave -q hw1_regression.m lambda sigma2 X_train.csv
y_train.csv X_test.csv
```

.

You must name your file as indicated above for your chosen language. If both files are present, we will only run your Python code. We will create and input the csv data files to your code. The value of lambda ("λ" above) will be a non-negative integer that we choose. The value of sigma2 ("σ^2" above) will be an arbitrary positive number.

.

The values of lambda and sigma2 will be input as strings. You must convert them to a number for your code to work. All numbers should be double-precision floating-point format. (In Matlab, "str2double()" is the way to do it.)

.

The csv files that we will input into your code are formatted as follows:.

1. **X_train.csv:** A comma separated file containing the covariates. Each *row* corresponds to a single vector x_i. *The last dimension has already been set equal to 1 for all data.*
2. **y_train.csv:** A file containing the outputs. Each row has a single number and the *i*-th row of this file combined with the *i*-th row of "X_train.csv" constitutes the training pair (y_i, x_i).
3. **X_test.csv:** This file follows exactly the same format as "X_train.csv". No response file is given for the testing data.

WHAT YOUR PROGRAM OUTPUTS

When executed, you should have your code write the output for *both* PART 1 & 2 to the files listed below. It is required that you follow the formatting instructions given below. For our chosen value of λ and σ^2, you will create the following two files containing the following information:

1. `wRR_[lambda].csv`: A file where the value in each dimension of the vector w_{RR} is contained on a new line. This file corresponds to your output for PART 1 above.
2. `active_[lambda]_[sigma2].csv`: A comma separated file containing the *row index* of the first 10 vectors you would select from X_test.csv starting with the measured values in X_train.csv and y_train.csv. ***Please make sure your indexing starts at 1 and not at 0 (i.e., the first row is index 1).*** This file should contain one line with a "," separating each index value. This file corresponds to your output for PART 2 above.

For example, if $\lambda = 2$ and $\sigma^2 = 3$, then the files you create will be named "wRR_2.csv" and "active_2_3.csv". If your code then learns that w = [3.2; -3.63; 1.4141; -0.78], then wRR_2.csv should look like:

```
3.2
-3.63
1.4141
-0.78
```

If the first 10 index values you would choose to measure are 724, 12, 109, 42, 23, 96, 342, 594, 123, 414, then active_2_3.csv should look like:

```
724,12,109,42,23,96,342,594,123,414
```

Note on Correctness

Please note that for both of these problems, there is one and only one correct solution. Therefore, we will grade your output based on how close your results are to the correct answer. We strongly suggest that you test out your code on your own computer before submitting. The UCI Machine Learning Repository (http://archive.ics.uci.edu/ml/) has a good selection of datasets for regression. While you still may not have the ground truth, you can build confidence that the outputs of your code are reasonable. For example, you can verify that your vector w_{RR} makes reasonable predictions and that your 10 selected measurement indexes are all unique.

USE OF VOCAREUM

This assignment uses Vocareum for submission and grading. Vocareum comes equipped with an editing environment that you may use to do your development work. You are **NOT** required to use the editor. In particular, you are free to choose your favorite editor / IDE to do your development work on. When you are done with your work, you can simply upload your files onto Vocareum for submission and grading.

However, your assignments will be graded on the platform, so you **MUST** make sure that your code passes at least the submission test cases. In particular, do not use third-party libraries and packages. We do not guarantee that they will work on the platform, even if they work on your personal computer. For the purposes of this project, everything that comes with the standard Python or Matlab libraries should be more than sufficient.

After you submit your code, the system will generate a file named ***SubmissionReport.txt***, which indicates any output formatting issues.

Once you submit your assignment, you will find submission report and grading report in your terminal. You will also find the same reports under **Details**.

- **You will have unlimited opportunities to submit your code for grading**
- **You can test your code in the terminal without submitting it**
- **You will get graded once you click “Submit”**
- **The assignment due date is May 3rd 2020, 23:30 UTC. Assignments submitted before March 8th 2020, 23:30 UTC are eligible for bonus points (we count grades on your latest submission). Due to edX policy, all assignment grades are capped at 100%.**

WORK ON PROJECT (ML.T) (External resource)

(25.0 / 25.0 points)

Your email address will be used to identify your submission entry.

Launch Project