Course › Unit 1 Linear Classifiers and Generalizations (2 weeks) › Project 1: Automatic Review Analyzer › 1. Introduction

1. Introduction

The goal of this project is to design a classifier to use for sentiment analysis of product reviews. Our training set consists of reviews written by Amazon customers for various food products. The reviews, originally given on a 5 point scale, have been adjusted to a +1 or -1 scale, representing a positive or negative review, respectively.

Below are two example entries from our dataset. Each entry consists of the review and its label. The two reviews were written by different customers describing their experience with a sugar-free candy.

Review	**label**
Nasty No flavor. The candy is just red, No flavor. Just plan and chewy. I would never buy them again	-1
YUMMY! You would never guess that they're sugar-free and it's so great that you can eat them pretty much guilt free! i was so impressed that i've ordered some for myself (w dark chocolate) to take to the office. These are just EXCELLENT!	1

In order to automatically analyze reviews, you will need to complete the following tasks:

1. Implement and compare three types of linear classifiers: the **perceptron** algorithm, the **average perceptron** algorithm, and the **Pegasos** algorithm.

2. Use your classifiers on the food review dataset, using some simple text features.

3. Experiment with additional features and explore their impact on classifier performance.

Setup Details:

For this project and throughout the course we will be using Python 3.6 with some additional libraries. We strongly recommend that you take note of how the NumPy numerical library is used in the code provided, and read through the on-line NumPy tutorial. **NumPy arrays are much more efficient than Python's native arrays when doing numerical computation. In addition, using NumPy will substantially reduce the lines of code you will need to write.**

1. *Note on software: For this project, you will need the **NumPy** numerical toolbox, and the **matplotlib** plotting toolbox.*

2. Download sentiment_analysis.tar.gz and untar it in to a working directory. The `sentiment_analysis` folder contains the various data files in **.tsv** format, along with the following python files:

- **project1.py** contains various useful functions and function templates that you will use to implement your learning algorithms.

- **main.py** is a script skeleton where these functions are called and you can run your experiments.

- **utils.py** contains utility functions that the staff has implemented for you.

- **test.py** is a script which runs tests on a few of the methods you will implement. Note that these tests are provided to help you debug your implementation and are not necessarily representative of the tests used for online grading. Feel free to add more test cases locally to further validate the correctness of your code before submitting to the online graders in the codeboxes.

Tip: Throughout the whole online grading system, you can assume the NumPy python library is already imported as `np`. In some problems you will also have access to python's `random` library, and other functions you've already implemented.

This project will unfold both on MITx and on your local machine. You are welcome to implement functions locally and run **test.py** to validate basic functionality, and then copy+paste your code into the MITx code boxes to fully check correctness and receive your grade for individual function implementations. Alternatively, you can also implement the functions online first and after finishing, copy+paste the solution to your local **project1.py** file. Be wary of the number of attempts you have for each problem, especially if you choose the second development flow.

How to Test Locally: In your terminal, navigate to the directory where your project files reside. Execute the command `python test.py` to run all the available tests.

How to Run your Project 1 Functions Locally: In your terminal, enter `python main.py`. You will need to uncomment/comment the relevant code as you progress through the project.

Discussion

Hide Discussion

Topic: Unit 1 Linear Classifiers and Generalizations (2 weeks):Project 1: Automatic Review Analyzer / 1. Introduction

Add a Post

Show all posts ▼ by recent activity ▼

? [Staff] Postponing the Project 1 due date
Hello Staff and everybody, Would it be possible to postpone the due date of Project 1 by at least a couple of days ? (current due date coincides with Homework 2)
9

💬 Thanks for creating such an amazing structure
I really liked the idea of test.py. I want to thank all the staff for creating the structure.
5

? [Staff]Due Date Project 1
In the last mail, when notifying of the extension for homework 1 and project 0, it was pointed out that the second part of Unit 1 was due con Tuesday but on the webpage it s...
1

💬 Python 3.6?
Does this mean the grader is running Python 3.6? The current version is 3.7, and I expect most folks who just installed Python have 3.7. This likely won't cause issues, so just c...
Community TA
10

💬 Project file
Hi. I often miss up when I have to start a project. I would like to write some code on my own IDE before implementing it on the exercises. I downloaded and opened the file gi...
6

💬 Can't run test.py
3

💬 Staff[Grader]
Hi. Not sure this is the right place to argue this nuance, but I'd like to let you know about it. I noticed if I save my code I no longer can visualize my output graded. Previously I ...
3

☑ How do I use test.py?
I am completely clueless when it comes to Python. I'm using a Jupyter notebook. I updated project1.py with my "hinge_loss_single" function and saved it, and then used the m...
13

Learn About Verified Certificates