COEN 346 Natural Language Processing

Programming Assignment 1: Sentiment Analysis

Due on Camino: Tuesday, April 16, 5pm, 2024

In this assignment, you will do sentiment classification by implementing the gradient descent algorithm for logistic regression in Python.

You are provided with a dataset comprising several thousand single-sentence reviews gathered from three different domains, namely imdb.com, amazon.com, and yelp.com. Each review consists of a sentence and a binary label indicating whether the sentiment of the sentence is positive or negative, with 1 representing positive and 0 representing negative sentiment. Below are some example reviews:

Amazon	the only VERY DISAPPOINTING thing was there was NO SPEAKERPHONE!!!!
IMDB	But the convoluted plot just didn't convince me, and much of the film was watched with a weird, questioning glance.
Yelp	Now the burgers aren't as good, the pizza which used to be amazing is doughy and flavorless.

You are given the data in CSV file format, with 2,400 input, output pairs in the training set, and 600 inputs in the test set. Please download the dataset at the links below:

https://www.cse.scu.edu/~yfang/coen346/p1/x_train.csv
https://www.cse.scu.edu/~yfang/coen346/p1/y_train.csv
https://www.cse.scu.edu/~yfang/coen346/p1/x_test.csv

Training set of 2,400 examples

- x_train.csv : input data.
    - Column 1: 'website_name': one of ['imdb', 'amazon', 'yelp']
    - Column 2: 'text': string sentence which represents the raw review
- y_train.csv : binary groundtruth labels
    - Column 1: 'is_positive_sentiment': 1 = positive sentiment, 0 = negative

Test set of 600 examples

- x_test.csv: input for test data

As discussed in class, there are different approaches to feature representation, the process of transforming a natural language document into a feature vector of a standard length. In this assignment, you will use the TF-IDF (term frequency inverse document frequency) "Bag-of-Words" representation with a fixed, finite-size vocabulary of V words. In other words, each review document is represented as a TF-IDF vector of length V, where entry at each dimension is the TF-IDF value of the corresponding vocabulary word in the document.

You are allowed to use sklearn Python package to perform tokenization and vector representation. Standard Python libraries and NumPy are also allowed. However, you cannot use built-in logistic regression model implementation in any libraries. You should implement the gradient descent algorithm from scratch by yourself.

You can check the reference below for more details about tokenization and TF-IDF representation from sklearn python package.

- feature_extraction.text.TfidfVectorizer: https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html#sklearn.feature_extraction.text.TfidfVectorizer

What to submit to Camino:

- yprob_test.txt: plain text file. Please use the exactly same file name in your submission.
  - Each line contains float probability that the relevant example should be classified as a positive example given its features. Please follow the order in the x_test.csv file for the test examples.
  - Should be loadable into NumPy as a 1D array via this snippet:

    ```
    np.loadtxt('yprob_test.txt')
    ```

  - Will be thresholded at 0.5 to produce hard binary predicted labels (either 0 or 1) for testing after submission
- Either .py or .ipynb file for the source code

Before you submit your predicted results, it is recommended that you use cross-validation to test how well your model performs by splitting the training data into your own training and validation datasets.