Homework 7: Text Bag-of-Words Search and Classification

About

This homework focuses on implementing a bag-of-word based pipeline to retrieve similar documents and classify reviews.

Due

Monday 4/1/19, 11:59 PM CST

Goal

The intent of this homework is to become familiar with using bag-of-words text representation to perform several tasks. Specifically, we will explore text retrieval by simple nearest neighbor queries and classification of using Logistic regression.

Submission

Submission will be through gradescope (https://www.gradescope.com)

Code and External Libraries

The assignment can be done using any programming language. Python is recommended.

External libraries can be used for all part of the assignment, although we limit how you can use a couple of libraries below.

Problems

Total points: 100

Part 1: Preprocessing with Bag-of-Words

Points: 20

At http://courses.engr.illinois.edu/cs498aml/sp2019/homeworks/yelp_2k.csv (http://courses.engr.illinois.edu/cs498aml/sp2019/homeworks/yelp_2k.csv)you will find a dataset of Yelp reviews. Theoriginal dataset (https://www.kaggle.com/yelp-dataset/yelp-

dataset/version/4) contains 5,261,668 reviews and we select 2000 from them, where half of them for reviews with 1 and 5 stars respectively.

1. Download and import the dataset. And then extract text and stars columns as your X (data) and y (label). You may find pandas or numpy package helpful (if you are using Python). Both have functions to load CVS files.
2. Convert the text into lower case then into bag-of-words representation. You can use a library such as one found in sklearn (if you are using Python). **Do not** use a pre-existing list of stop-words.
3. Bag-of-words Analysis and Repreprocessing.
    - Graph the distribution of words counts vs word rank.

      Example graph (make yours larger):

    - Identify the set of common stop words by looking at the words. What stop words did you choose?
    - Choose a max document frequency theshold for word occurances and a minimum word occurance to cull the less useful words.
    - Reprocess your data using the stop-words list you determined, the max document frequency and the minimum word occurance.
    - Graph the updated words counts vs word rank.
4. After removing stop-words, convert all the data into bag-of-words vectors for use in the next part.

Part 2: Text-Retrieval

Points: 30

In this section, we look at finding similar documents to a query phrase.

1. Using nearest neighbor with a cos-distance metric, find 5 reviews matching *Horrible customer service*.
2. Print the original text from the review along with the associated distance score. You can truncate the review to 200 words so we don't get a page of text.
3. Looking at all the distance scores, how many documents do you think are good matches to the query?

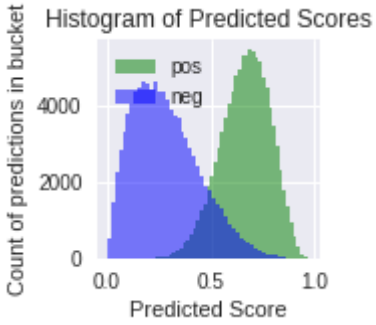
Part 3: Classification with Logistic Regression

Points: 50

Here, we attempt to classify good reviews vs bad reviews using Logistic Regression.

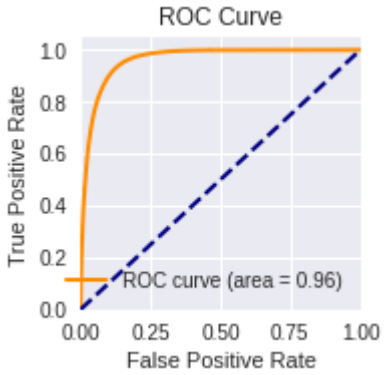
1. Separate your data in train and test sets. Use 10% of the data for test.
2. Create a classifier based on Logistic Regression. Feel free to use whatever packages you like. In Python, sklearn should have what you need.
3. What is the accuracy on the training set and the test set of your classifier?
4. Logistic Regession returns a probability of the positive label. Although it is common to use 0.5 as the threshold to call something positive or negative, depend on your use case and the data, sometimes a different threshold is better.
    - Plot a histogram of the scores on the training data.

      Example (make yours bigger):

    - Choose a new threshold based on your plot and report the accuracy on the training set and the test set. Did it improve?
    - An ROC curve shows the trade-off of correct positive predictions (true positives) vs incorrect positive predictions (false positives) as the classification threshold is adjusted. Plot the ROC curve for your classifier.

      Example (make yours larger):

    - At what false positive rate would your classifier minimize false positives while maximizing true positives?

Submission

Your submission should be a PDF with the following pages.

Page 1 Distribution graph (5 points)

Show the distribution graph of words counts vs word rank.

Page 2 Identify the stop words (5 points)

List the stop words you choose as well as the frequency threshold.

Page 3 Distribution graph again (5 points)

After chosing the stop words, show the distribution graph of words counts vs word rank.

Page 4 Code snippets (15 points)

Show the snippet of your code that you convert all the reviews into bag-of-words formulation using your chosen stop words and your code for nearest-neighbours with cos-distance.

Page 5 Reviews with score (10 points)

Show the original reviews with the distance scores

Page 6 Query results (10 points)

Show your document results and explain the reasons that you choose them.

Page 7 Accuracy with threshold 0.5 (10 points)

Show your code for creating classifier. Report the accuracy on train and test dataset with threshold 0.5.

Page 8 Predicted scores (10 points)

Show your code for plotting predicted scores and show the figure.

Page 9 Accuracy again and curve (20 points)

Report the accuracy on train and test dataset with a different threshold. Explain why you choose that threshold.

Plot the ROC curve.

Page 10 Best threshold (10 points)

Choose the threshold that minimizes false positives while maximizing true positives. Explain your reason.