CSE 158/258, Fall 2019: Homework 4

Instructions

Please submit your solution **by the beginning of the week 9 lecture (Nov 25).** Submissions should be made on **gradescope**. Please complete homework **individually**.

Download the Assignment 1 data from the course webpage: `http://cseweb.ucsd.edu/classes/fa19/cse258-a/files/assignment1.tar.gz`. We will use the reviews in `train_Category.json.gz`.

Code is provided on the course webpage (week5.py) and in the lecture notebook showing how to load and perform simple processing on the data. Executing the code requires a working install of Python 3.0 with the scipy packages installed.

Tasks

Using the code provided on the webpage, read the *first 10,000* reviews from the corpus, and read the reviews **without capitalization or punctuation**.

1. How many unique bigrams are there amongst the reviews? List the 5 most-frequently-occurring bigrams along with their number of occurrences in the corpus (1 mark).

2. The code provided performs least squares using the 1000 most common unigrams. Adapt it to use the 1000 most common *bigrams* and report the MSE obtained using the new predictor (use bigrams *only*, i.e., not unigrams+bigrams) (1 mark). Note that the code performs *regularized* regression with a regularization parameter of 1.0. The prediction target should be the 'rating' field in each review.

3. Repeat the above experiment using unigrams *and* bigrams, still considering the 1000 most common. That is, your model will still use 1000 features (plus an offset), but those 1000 features will be some combination of unigrams and bigrams. Report the MSE obtained using the new predictor (1 mark).

4. What is the *inverse document frequency* of the words 'stories', 'magician', 'psychic', 'writing', and 'wonder'? What are their *tf-idf* scores in the first review (using log base 10, unigrams only, following the first definition of tf-idf given in the slides) (1 mark)?

5. Adapt your unigram model to use the tfidf scores of words, rather than a bag-of-words representation. That is, rather than your features containing the word *counts* for the 1000 most common unigrams, it should contain tfidf scores for the 1000 most common unigrams. Report the MSE of this new model (1 mark).

6. Which other review has the highest cosine similarity compared to the first review, in terms of their tf-idf representations (considering unigrams only). Provide the review_id, or the text of the review (1 mark)?

7. Implement a validation pipeline for this same data, by randomly shuffling the data, using 10,000 reviews for training, another 10,000 for validation, and another 10,000 for testing.[1] Consider regularization parameters in the range $\{0.01, 0.1, 1, 10, 100\}$, and report MSEs on the *test* set for the model that performs best on the *validation* set. Using this pipeline, compare the following alternatives in terms of their performance (all using 1,000 dimensional word features):

   - Unigrams vs. bigrams
   - Removing punctuation vs. preserving it. The model that preserves punctuation should treat punctuation characters as separate words, e.g. "Amazing!" would become ['amazing', '!']
   - tfidf scores vs. word counts

   In total you should compare $2 \times 2 \times 2 \times 5 = 40$ models (8 models and 5 regularization parameters), and produce a table comparing their performance (2 marks)

[1] You may use smaller samples of the data if experiments are taking too long.