CSE 258, Fall 2018: Homework 4

Instructions

Please submit your solution **by the beginning of the week 9 lecture (Nov 26).** Submissions should be made on **gradescope**. Please complete homework **individually**.

Download the "50,000 beer reviews" data from the course webpage: `http://jmcauley.ucsd.edu/cse158/data/beer/beer_50000.json`. Code is provided on the course webpage (week5.py) showing how to load and perform simple processing on the data. Executing the code requires a working install of Python 2.7 or Python 3.0 with the scipy packages installed.

Tasks

Using the code provided on the webpage, read the *first 5000* reviews from the corpus, and read the reviews **without capitalization or punctuation**.

1. How many unique bigrams are there amongst all of 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.
3. What is the *inverse document frequency* of the words 'foam', 'smell', 'banana', 'lactic', and 'tart'? What are their *tf-idf* scores in the first review (using log base 10) (1 mark)?
4. What is the cosine similarity between the first and the second review in terms of their tf-idf representations (considering unigrams only) (1 mark)?
5. Which other review has the highest cosine similarity compared to the first review (provide the beerId and profileName, or the text of the review) (1 mark)?
6. Adapt the original model that uses the 1000 most common unigrams, but replace the features with their 1000-dimensional tf-idf representations, and report the MSE obtained with the new model.
7. Implement a validation pipeline for this same data, by randomly shuffling the data, using 5000 reviews for training, another 5000 for validation, and another 5000 for testing. 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:

   - 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 vs. word counts

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