CSE 158/258, Fall 2021: Homework 4

Instructions

Please submit your solution **by Monday Nov 21.** Submissions should be made on **gradescope**. Please complete homework **individually**.

You should submit two files:

`answers_hw4.txt` should contain a python dictionary containing your answers to each question. Its format should be like the following:

```
{ "Q1": 1.5, "Q2": [3,5,17,8], "Q2": "b", (etc.) }
```

The provided code stub demonstrates how to prepare your answers and includes an answer template for each question.

`homework4.py` A python file containing working code for your solutions. The autograder *will not execute your code*; this file is required so that we can assign partial grades in the event of incorrect solutions, check for plagiarism, etc. Your solution should **Clearly document which sections correspond to each question and answer**.

You may build your solution on top of code from the textbook:

Text Mining: `https://cseweb.ucsd.edu/~jmcauley/pml/code/chap8.html`

Content and Structure: `https://cseweb.ucsd.edu/~jmcauley/pml/code/chap6.html`

You will need the following files:

Homework 4 stub : `https://cseweb.ucsd.edu/classes/fa22/cse258-a/stubs/`

GoodReads Young Adult Reviews (20,000) : `https://cseweb.ucsd.edu/classes/fa22/cse258-a/data/young_adult_20000.json.gz`

Tasks: Text Mining

Use the first half (10,000) of the book review corpus for training and the rest for testing (code to read the data is provided in the stub). Process reviews **without capitalization or punctuation** (and without using stemming or removing stopwords).

1. Build a *sentiment analysis model* that estimates star ratings from a 1,000 word bag-of-words model (based on the most popular words). Compare models based on:

   (a) the 1,000 most common unigrams;

   (b) the 1,000 most common bigrams;

   (c) a model which uses a combination of unigrams and bigrams (i.e., some bigrams will be included if they are more popular than some unigrams, but the model dimensionality will still be 1,000).

   You may use a Ridge regression model (`sklearn.linear_model.Ridge`) with a regularization coefficient of $\lambda = 1$). Report the MSE on the test set for each of the three variants, along with the five most negative and most positive tokens for each variant (2 marks).

2. Which review has the highest cosine similarity compared to the first review in the dataset, in terms of their tf-idf representations (using only the training set, and considering unigrams only)? Report the cosine similarity and the text of the review (2 marks).

Tasks: Content, Structure, and Sequences

For these tasks, you may consider the entire set of 20,000 reviews.

3. Using the *word2vec* library in *gensim*, fit an item2vec model, treating each 'sentence' as a temporally-ordered[1] list of items per user. Use parameters `min_count=1`, `size=10`, `window=3`, `sg=1`.[2] Report the 5 most similar items to the book from the first review along with their similarity scores (your answer can be the output of the `similar_by_word` function) (1 mark).

4. The above model's item representations can be accessed via `model.wv[itemID]`. Implement a rating prediction function of the form

$$r(u,i) = \bar{R}_i + \frac{\sum_{j \in I_u \setminus \{i\}} (R_{u,j} - \bar{R}_j) \cdot \mathrm{Sim}(i,j)}{\sum_{j \in I_u \setminus \{i\}} \mathrm{Sim}(i,j)},$$

   using the cosine similarity between item representations as your similarity function. Report the MSE of this model, *on the first 1,000 ratings only* (1 mark).

5. By making modifications to your item2vec model (e.g. changing the model parameters) or otherwise, improve the model from the previous question in terms of the MSE. Describe your modification in a few words, and report its MSE (again on the first 1,000 ratings) (2 marks).

[1] You may use `dateutil.parser.parse` to parse the date string.

[2] The `size` argument might be `vector_size` in some library versions.