Homework 4

You should submit a single file containing your code ("homework4.py") to the autograder

Each question is worth two marks.

Text mining

We'll use part of the 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 λ = 1). Compute the MSE on the test set for each of the three variants.
2. Find the reviews with 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).

Content, Structure, and Sequences

For these tasks, you may consider the entire dataset (i.e., no train/test splits).

3. Using the word2vec library in gensim, fit an item2vec model, treating each 'sentence' as a temporally-ordered list of items per user. Use parameters `min_count=1,size=10,window=3,sg=1` (already in stub). Find the 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).
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. Compute the MSE of this model. Next, 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 (as usual, it should be better than the above model, *or*

always predicting the average). The easiest improvement is to adjust the normalization (denominator) in the above expression, which is highly unstable.