CSE 158/258, Fall 2021: Homework 4

Instructions

Please submit your solution **by the beginning of the week 9 lecture (Nov 22).** Submissions should be made on **gradescope**. Please complete homework **individually**.

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`

Temporal Recommendation: `https://cseweb.ucsd.edu/~jmcauley/pml/code/chap7.html`

We'll base our solution on data from Goodreads Comic Book reviews:
`https://cseweb.ucsd.edu//classes/fa21/cse258-b/data/goodreads_reviews_comics_graphic.json.gz`

Tasks: Text Mining

You may conduct these exercises using just the first *first 20,000* reviews from the corpus. Use the first half for training and the rest for testing. 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 positive and most negative 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)? Provide the ID and text of the review (2 marks).

Tasks: Content, Structure, and Sequences

For these tasks, consider the complete dataset, but *discard any users who have fewer than three interactions.* Build a training set by considering *all but the last* interaction from each user (in order of timestamp). The test set will then contain *each user's most recent interaction.*

3. Using the fastFM library[1] compare three model variants to predict ratings:[2]

   (a) A regular latent-factor model, i.e., one which includes a user term and an item term ($f(u, i)$);

   (b) A non-personalized Markov Chain model, i.e., one which combines a one-hot encoding of the previous item with a one-hot encoding of the current item ($f(i, i^{(\text{prev})})$).

   (c) A model which includes both the user, the item, and the previous item ($f(u, i, i^{(\text{prev})})$).

   Report the training and test MSE for each variant; you may follow the same hyperparameter settings as in the starter code (3 marks).[3]

[1] Or adapting the FPMC Tensorflow code from Chapter 7.

[2] You may also perform ranking if you prefer, depending on which code you're adapting; e.g. the Tensorflow code implements a Bayesian Personalized Ranking-style model. If using the Tensorflow code you can report and estimate the AUC by measuring how often the 'predict' function returns a positive score (positive item has a higher score than the negative item). Please just specific what library you used and how you approached the problem.

[3] Don't be too surprised if the Markov Chain model doesn't work well for this task—not every dataset is 'Markovian'!

4. Experiment with the factorization machine code to see whether performance of the above models can be improved by incorporating any other features from the data (1 mark).[4] **If you can't get any FM library working, you may skip this question and simply describe what features you might try.**

[4]Though ideally, you should avoid using the review text, which wouldn't be available at test time in a real application!