CSE 158/258, Fall 2021: Homework 2

Instructions

Please submit your solution **by the beginning of the week 5 lecture (Oct 25).** Submissions should be made on **gradescope**. Please complete homework **individually**.

This specification includes both questions from the undergraduate (CSE158) and graduate (CSE258) classes. You are welcome to attempt questions from both classes but will only be graded on those for the class in which you are enrolled.

You may base your solution on the Chapter 4 code samples:

`https://cseweb.ucsd.edu/~jmcauley/pml/code/chap4.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`

(code to read the dataset can be found in the sample above)

Please include the code of (the important parts of) your solutions.

Tasks — Similarity Functions:

1. Which 10 items have the highest Jaccard similarity compared to the first item (i.e., the item from the first review, '18471619')? Report both similarities and item IDs (1 mark).

2. There are several ways similar-item recommendations could be used to make personalized recommendations for a particular user. For instance we could:

   (a) Choosing the N items most similar to the user's favorite (i.e., highest rated) item.

   (b) Finding the N most similar users, and recommending each of their *their* favorite (highest rated) items.

   Implement these two strategies for user 'dc3763cdb9b2cae805882878eebb6a32' (i.e., the user from the first review); in both cases use the Jaccard similarity as your measure of item-to-item or user-to-user similarity. In case of ties, always select the alphabetically first user or item. Report the top 10 items (and associated scores) in each case; note that you should avoid recommending items the user has already interacted with (2 marks).[1]

3. In class we briefly discussed whether the Pearson similarity should be implemented (a) only in terms of *shared* items (i.e., $U_i \cap U_j$) in the denominator; or (b) in terms of all items each user consumed (i.e., U_i or U_j for each term in the denominator). (See last slide on Pearson similarity). Implement versions of the Pearson similarity based on both definitions, and report the 10 most similar items to the same query item from Question 1 (2 marks).

Tasks — Rating Prediction:

4. Implement a rating prediction model based on the similarity function

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

   (there is already a prediction function similar to this in the starter code, you can either start from scratch or modify the solution in the starter code). Report the MSE of this rating prediction function when $\text{Sim}(i,j) = \text{Jaccard}(i,j)$ (1 mark).[2]

5. **CSE158 only** Modify the similarity function from Question 4 to use:

   (a) The cosine similarity;

   (b) The two definitions of Pearson similarity from Question 3;

[1] In the event that you have to skip a user or item due to seeing an already-consumed item, you may take the next-highest-rated item from that user, or simply skip that user; feel free to describe your strategy in ambiguous or edge cases.

[2] If computing the full MSE is prohibitively slow, you may report the MSE on a subset of the data, e.g. 10,000 samples; just clearly state how you compute the MSE.

(c) The Jaccard similarity, but interchanging users and items (i.e., in terms of the similarity between users $\text{Sim}(u, v)$ rather than $\text{Sim}(i, j)$).

Report the MSE of each strategy (2 marks).[3]

6. **CSE258 only** Later in the quarter, we'll explore recommender systems based on temporal dynamics. Here, we'll explore a simple form of temporal recommendation known as *time-weight collaborative filtering*. Here, interactions are weighted in terms of their recency, i.e.,

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

where $t_{u,j}$ is the timestamp associated with the rating $R_{u,j}$. For example a decay function might be based on exponential decay, i.e., $f(t) = e^{-\lambda t}$, where λ is a controllable parameter. Design a decay function that outperforms (in terms of the MSE) the trivial function $f(t_{u,j}) = 1$, documenting any design choices you make. You are welcome to consider the timestamp of the target rating $R_{u,i}$ in your function (e.g. you might choose a function of the form $f(|t_{u,i} - t_{u,j}|)$, or similar) (2 marks).[4]

[3] If a particular heuristic performs poorly in 'edge cases', you are welcome to alter your strategy, e.g. by simply predicting $\bar{R}_i$ or otherwise; document any such modifications you make.

[4] Any minimal improvement in the MSE is acceptable.