CS F469, Information Retrieval: Assignment-2

Instructor: Vinti Agarwal (vinti.agarwal@pilani.bits-pilani.ac.in)
Topic: CF based recommender system

Due Date: 21^{st} **November,** 2020

The objective of the assignment is to get your hands on designing recommender system for movies data using collaborative filtering (CF) personalization techniques and suggest target user top-5 movies he/she will most likely to watch.

There are two popular types of CF: User-based (UB) and item based(IB). In Part A of the assignment students are advised to start working with UB-CF. Later in Part B, you are encouraged to come up with some improvements and new methods to overcome the challenges of UB-CF. The idea of UB-CF is based on the cooperation among a group of users to glean the opinions about various unobserved items. In order to decide on people from whom opinion are sought, similarity function is used that determines closed similar neighbors and ratings assigned by them. These rating are aggregated to predict the affinity toward an item for the target user.

Dataset statistics

The dataset is collected from MovieLens (`http://movielens.org`), a movie recommendation service. It contains 100836 ratings from 610 users and 3683 tag applications across 9742 movies. Every user has rated at least 20 movies. The dataset can be found on the given google drive link
`https://drive.google.com/drive/folders/13P_JmROmW60OEZLqNSaMrlCEumcTfpRH?usp=sharing`

The files contained in data are:

1. **ratings.csv:** All ratings of the movies are stored in this file, where row represents one rating of one movie by one user, and has the following format:
   userId, movieId, rating, timestamp

2. **tags.csv:** All tags are contained in this file. Each row of this file has the following format:
   userId, movieId, tag, timestamp Tags are user-generated metadata about movies such as a single word or short phrase. The meaning, value, and purpose of a particular tag is determined by each user.

3. **movie.csv** All the information about movies is contained in this file, where each line has the following format:
   movieId, title, genres

Part A

Design an overall user-based CF recommender system and evaluate its performance. In order to design the model, kindly proceed with the following instructions:

1. Construct Utility matrix (user vs. movies), where user profile is considered as a vector of ratings it has assigned to movies.

2. Neighborhood generation using Pearson correlation coefficient. (Neighborhood selection criteria can be based on either threshold or top-N user or both).

3. Prediction of the rating for target user u for an unseen movie m using resnick prediction formula

$$r_{u,m} = R(u) + \frac{\sum_{v \in N(u)} sim(u,v) r_{v,m}}{\sum_{v \in N(u)} | \ sim(u,v)|}$$

where $R(u)$ represents the average rating of user u and $N(u)$ represents the set of users similar to u.

Once the system design is complete, you need to evaluate the model using k-fold (using k=5) cross validation scheme and measure the effectiveness of prediction using mean absolute error (MAE). You are advised to use tabular format for presenting MAE values corresponding to each fold.
After this, you need to list the top-5 movies predicted by the system on at least 10 randomly chosen users from the data (which were not used during training phase). For each user, list the top-5 predicted movies along with predicted ratings, and the highly ranked (where r>3) movies the user has seen in the past as per the following format:

Test User	Predicted Movies		Movies seen in past	
	Movies	Ratings	Movies	Ratings (> 3)
user1	M11	R11	M21	R21
	M12	R12	M22	R22
	M13	R13	M23	R23
	M14	R14	M24	R24
	M15	R15	M25	R25
user2				

Table 1: Output structure to be saved in output.csv

Part B

In this part, you need to come up with some improvements over the previously used approach in terms of the recommended movies list. Students can attempt this using additional information(such as movie genre, tagging etc) provided in the data. You are required to propose and implement at least two improvements.For each improvement, answer the following questions briefly:

1. What are the challenges in CF-RS built in part A?
2. What improvements are you proposing?
3. How will the proposed improvement address the issues?
4. A corner case (if any) where this improvement might not work or can have an adverse effect.
5. Demonstrate the actual impact of the improvement. Give three examples, where the improvement yields better results compared to the part A implementation.

   There are some additional questions (optional) you can try to answer based on Part-A:

6. What is the significance of multiplying the value of $r_{v,m}$ by the similarity of user u to user v, in resnick prediction formula?
7. What could the additional possible information from data to be exploited to improve existing system developed in part A?

Note: You need to work on the assignment using python programming language. Display of the results in the terminal/notebook would be sufficient. No need to design GUI.

Code submission guidelines:

There must be two main script files (along with data and results) in the folder submitted by the student:

1. *RS_main.py*: This file will be main file to run recommender system including user-item matrix generation, neighborhood generation, prediction and and performance evaluation. This should take command line arguments: input as rating.csv and save the output of MAE performance evaluation in eval.csv
   e.g. *python RS_main.py −−input rating.csv −−output eval.csv*

2. *test.py* : This file will take input as a list of test user (already saved by students in *test_user.txt*) and save output in *output.csv* as the list of the top-5 recommended movies along with previously seen in the format suggested in table 1.
   e.g. *python test.py −−input test_user.txt −−output output.csv*

Deliverables:

1. Well commented code: The purpose and intent of each method, class and module should be mentioned appropriately. Code submissions that are not well-commented will lead to a loss of marks.

2. Report: A report describing all important assumptions made for implementing, limitations, algorithms used etc. It must include part A evaluation results and the answers to part B questions.

3. Readme file having all the steps for running your code.

4. Innovations: You are encouraged to deploy some novel methods or techniques that will allow your code to run efficiently. This could include, but not limited to using different data structures, use of accurately placed skip pointers etc. Make sure to mention all such innovations in your report.

FAQ

- Q1: How will this assignment be evaluated?
  A1: Three parameters:
  1. The working code.
  2. The report.
  3. Your understanding of the code, report and about the overall working of the system.

- Q2: Can I use an XX library?
  A2: You are free to use any library which is open source.

- Q3: What is the weightage in course?
  A3: 15%

- Q4: What are the weightage of part A and part B?
  A4: Part A and Part B both carry equal weightage (i.e. 50% and 50%). In case, a student does not answers/partially answers questions from Part B, some extra marks can be awarded if optional questions (6 and 7) are answered, otherwise not.

- Q5: What is the evaluation scheme for the assignment?
  A5: The evaluation scheme will be as follows:.

Task	Marks
Program code (Part A + Part B)	5+5
Results and Output (Part A + Part B)	2.5 + 2.5
Report	5
Viva and Demo	10
Total Marks	30

There will be a **Viva and Demo** for the evaluation of this assignment, the details of which will be announced later.

Assignment submission deadline is **21st November 2020 5pm.** This is a **hard deadline** since Canvas won't accept submissions beyond this date and time. No late submissions will be entertained.

In case of any the doubts related to the assignment; You can join me via google meet at the given link `meet.google.com/ysr-apjw-rot` on October 23, 2020 between 12pm-1pm.

I wish you the best!!