AIE425 Intelligent Recommender Systems
Assignment #2: Significance Weighting-based Neighborhood CF Filters
Submission Date: During Week 12 Lab (Tuesday, December 17, 2024)

1. Instructions

- The total number of pages if this file is 6 pages.
- Students complete the assignment in order.
- Each student must use his/her GitHub account that was created in assignment 1.
- Each student **MUST** prepare the homework solution and write the report **BEFORE ARRIVING WEEK 12 LAB**. Your write-up includes the requirements stated in Section 3.4. HANDWRITTEN scans for your write-up ARE NOT ACCEPTED.
- **Each student must finish the assignment during week 12 lab and post the solution and the pre-prepared report to GitHub.**
- Failure to attend week 12 lab and complete the assignment will result in a ZERO in this assignment.

2. Introduction

This assignment consists of the following TWO parts.

Part 1: in the case of user-based CF, you need to demonstrate the significance of weighting schemes on the performance of the recommender systems.

Part 2: in the case of item-based CF, you need to demonstrate the significance of weighting schemes on the performance of the recommender systems.

3. Assignment Description and requirements

3.1. General requirements for the TWO parts:

1. Use the dataset you generated in assignment 1.
2. Adjust the rating on a 1-to-5 scale.
3. Count the total number of users in your dataset and save it a variable called "*tnu*".
4. Count the total number of items in your dataset and save it a variable called "*tni*".
5. Count the number of ratings for every product in this dataset.
6. Pick three users (U1, U2 and U3) and consider them active users. One user with 2 missing ratings, another user with 3 missing ratings, and the other user with 5 missing ratings.
7. Pick two items (I1 and I2) and consider them target items. One item with 4% missing ratings, and the other item with 10% missing ratings.
8. Count the number of users who have co-rated items with the active user ("*No_common_users*"), and the number of co-rated items ("*No_coRated_items*").
9. Create a 2-D array in which the first column contains the "*No_common_users*" in descending order, the corresponding "*No_coRated_items*" go in the second column.
10. Draw a curve that illustrates the quantity of ratings for every item in your dataset.
11. Determine the maximum number of users who have co-rated at least 30% of items with each of the active users and consider it the threshold β in each case.
    Note: finding different threshold β for each active user is expected.
12. Save all the above results for later use.

3.2. Part 1 requirements and questions:

Part 1 includes the following THREE case studies.

Case study 1.1:

1.1.1. Apply user-based collaborative filtering algorithms using Cosine similarity without considering the bias adjustment effect of mean-centering to compute the similarity between each active user and other users.
1.1.2. Use the Cosine similarity from point 1.1.1 to determine the top 20% closest users to each active user.
1.1.3. Use the results from point 1.1.2 to compute the prediction for each active user to find whether the user will like or dislike the not yet seen or rated items.
1.1.4. Compute the discount factor (DF) then the discounted similarity (DS), for each of the active users considering the threshold β in each case.
1.1.5. Use the discounted similarity to determine the top 20% closest users to each active user.
1.1.6. Use the results from point 1.1.5 to compute the prediction for each active user to find whether the user will like or dislike the not yet seen or rated items.
1.1.7. Compare the results of point 1.1.2 with results of point 1.1.5. Comment on your answer.
1.1.8. Compare the results of point 1.1.3 with results of point 1.1.6. Comment on your answer.

Case study 1.2:

1.2.1. Apply user-based collaborative filtering algorithms using Cosine similarity considering the bias adjustment effect of mean-centering to compute the similarity between each active user and other users.
1.2.2. Use the Cosine similarity from point 1.2.1 to determine the top 20% closest users to each active user.
1.2.3. Use the results from point 1.2.2 to compute the prediction for each active user to find whether the user will like or dislike the not yet seen or rated items.
1.2.4. Compute the discount factor (DF) then the discounted similarity (DS), for each of the active users considering the threshold β in each case.
1.2.5. Use the discounted similarity to determine the top 20% closest users to each active user.
1.2.6. Use the results from point 1.2.5 to compute the prediction for each active user to find whether the user will like or dislike the not yet seen or rated items.
1.2.7. Compare the results of point 1.2.2 with results of point 1.2.5. Comment on your answer.
1.2.8. Compare the results of point 1.2.3 with results of point 1.2.6. Comment on your answer.

Case study 1.3:

1.3.1. Apply user-based collaborative filtering algorithms using Pearson Correlation Coefficient (PCC) to compute the similarity between each active user and other users.
1.3.2. Use the PCC from point 1.3.1 to determine the top 20% closest users to each active user.
1.3.3. Use the results from point 1.3.2 to compute the prediction for each active user to find whether the user will like or dislike the not yet seen or rated items.
1.3.4. Compute the discount factor (DF) then the discounted similarity (DS), for each of the active users considering the threshold β in each case.
1.3.5. Use the discounted similarity to determine the top 20% closest users to each active user.
1.3.6. Use the results from point 1.3.5 to compute the prediction for each active user to find whether the user will like or dislike the not yet seen or rated items.
1.3.7. Compare the results of point 1.3.2 with results of point 1.3.5. Comment on your answer.
1.3.8. Compare the results of point 1.3.3 with results of point 1.3.6. Comment on your answer.

- Compare and comment on the results of Case study 1.1, 1.2, and 1.3.

3.3. Part 2 requirements and questions:

Part 1 includes the following THREE case studies.

Case study 2.1:

2.1.1. Apply item-based collaborative filtering algorithms using Cosine similarity without considering the bias adjustment effect of mean-centering to compute the similarity between each target item and other items.
2.1.2. Use the Cosine similarity from point 2.1.1 to determine the top 25% closest items to each target item.
2.1.3. Use the results from point 2.1.2 to compute the prediction of the missing ratings for each target item.
2.1.4. Compute the discount factor (DF) then the discounted similarity (DS), for each of the target items considering the threshold β in each case.
2.1.5. Use the discounted similarity to determine the top 20% closest items to each target item.
2.1.6. Use the results from point 2.1.5 to compute the prediction of the missing ratings for each target item.
2.1.7. Compare the results of point 2.1.2 with results of point 2.1.5. Comment on your answer.
2.1.8. Compare the results of point 2.1.3 with results of point 2.1.6. Comment on your answer.

Case study 2.2:

2.2.1. Apply item-based collaborative filtering algorithms using Cosine similarity considering the bias adjustment effect of mean-centering to compute the similarity between each target item and other items.
2.2.2. Use the Cosine similarity from point 2.2.1 to determine the top 20% closest items to each target item.
2.2.3. Use the results from point 2.2.2 to compute the prediction of the missing ratings for each target item.
2.2.4. Compute the discount factor (DF) then the discounted similarity (DS), for each of the target items considering the threshold β in each case.
2.2.5. Use the discounted similarity to determine the top 20% closest items to each target item.
2.2.6. Use the results from point 2.2.5 to compute the prediction of the missing ratings for each target item.
2.2.7. Compare the results of point 2.2.2 with results of point 2.2.5. Comment on your answer.
2.2.8. Compare the results of point 2.2.3 with results of point 2.2.6. Comment on your answer.

Case study 2.3:

2.3.1. Apply item-based collaborative filtering algorithms using Pearson Correlation Coefficient (PCC) to compute the similarity between each target item and other items.
2.3.2. Use the PCC from point 2.3.1 to determine the top 20% closest items to each target item.
2.3.3. Use the results from point 2.3.2 to compute the prediction of the missing ratings for each target item.
2.3.4. Compute the discount factor (DF) then the discounted similarity (DS), for each of the target items considering the threshold β in each case.
2.3.5. Use the discounted similarity to determine the top 20% closest items to each target item.
2.3.6. Use the results from point 2.3.5 to compute the prediction of the missing ratings for each target item.
2.3.7. Compare the results of point 2.3.2 with results of point 2.3.5. Comment on your answer.
2.3.8. Compare the results of point 2.3.3 with results of point 2.3.6. Comment on your answer.

- Compare and comment on the results of Case study 2.1, 2.2, and 2.3.

3.4. Discussion and Conclusion:

During the homework and **BEFORE YOU COME TO LAB 12**, prepare a report that include the following

1- A section called "Outcomes of Section 3.1" summarizes the outcomes of each of the requirements in Section 3.1.

2- A section called "Summary of the Comparison of part 1 and 2", summarizes your comments about the comparisons you made in part 1 and part 2, emphasizing the impact of applying significance weighting to the top-N list and rating prediction in both parts.

3- A "conclusion" section, which summarizes your own comments and conclusion, shows the impact of the significance weighting and addresses any improvements from your perspective.

4. Method and significance of the assessment

- This coursework accounts for **5%** of the course mark and is done **INDIVIDUALLY**.
- This assessment will demonstrate progression of the student's academic ability and assess students' achievement intended learning outcomes (ILOs) of the course.
- The marks will be awarded pro-rata, depending on the details, evidence of academic and technical talents, professionalism, information organization, and discussion skills.
- A **40%** reduction will be applied if evidence of thoroughness is not adequate.
- If the student appears to be performing it right but makes a mistake, they will receive a maximum of 70% for any of the requirements.

5. Marking schema:

- 10% of the mark for the report as described in Section 3.4, including evidence of the presented knowledge, topic understanding, completeness of the information, analysis, and the contribution of the student.
- 80% of the mark for the implementation, experiments, and results according to the requirements.
- 10% of the mark for posting the report, code, dataset, and the PLAGIARISM REPOR on GitHub.

6. Report (paper) Format:

Report must follow the following.

- be word-processed,
- a report format standard,
- use correct paragraphing, formal grammar, tenses, and spelling,
- be submitted on A4,
- The title page includes the Assignment title <AIE425 Intelligent Recommender Systems, Fall Semester 24/25> on the first line, <Assignment #2: Significance Weighting-based Neighborhood CF Filters> on the second line, and <Student ID, Full Name> on the third line. Use double line space, centered contents and without page numbering.
- All other pages are 1.5 line spaced,

- use 12-point Arial font size for normal text,
- use 12-point Arial font size, bolded for headings,
- have page numbers centered on the bottom of each page in the format <Page X of Y>,
- section headings are with the same font, size and alignment.

7. Plagiarism and Academic Honesty

- INTEGRITY and COLLABORATION: Student are encouraged to discuss issues related to the assignment with other students, but genuine collaboration on all or part of the assignment must be explicitly acknowledged, or he/she will be penalized.
- PLAGIARISM is strictly prohibited and may result in failure in this course.
- This is an exercise, so submit your own work. If you submit material that is not entirely your own, you must state this clearly in your submission.
- A PLAGIARISM REPORT is required for the submitted report. A similarity ratio of greater than 30% is not acceptable.
- Code and/or written material should NOT be shared. Even a single line of code without reference will be considered plagiarism.
- Please do not use external code unless authorized. This is easier to detect than you might think. If you must do so for any standard initialization and embedding parts, this must be referenced and properly marked/highlighted.
- ASK Teaching Staff If you have any queries or are confused whether specific activities constitute dishonesty. It's better to be safe than sorry.

8. Feedback given to students in response to assessed work.

- Specific oral feedback on the assessed components.
- The written component will be assessed directly through annotations on the page.
- Feedback for programming will also be placed on the coursework assessment sheet returned with the coursework mark.
- During contact hours, students will receive oral generic comments on every part of their assignment..
- If students need more input, they are encouraged to speak with the teaching staff.

9. Deliverables

- The whole assignment (report, dataset, and code) will be submitted on GitHub during week 12 lab.
  - All files MUST be named as follow: <StudentID_StudentFirstName_FileName>.
  - Put your name, and ID at the top of each file even in programs.
- Late submission, if there is a solid reason, is permissible during the next day subject to the arrangement with the teaching staff, however you will lose 50% of the coursework mark.
- more than 48 hours delay is not accepted, and you will get **ZERO** in this assignment.

----------- END -----------