vNational University of Computer & Emerging Sciences, Karachi
Spring 2023
School of Computing
Final Examination [Solution]
27th May 2023, 09:00 am – 12:00 noon

Course Code: CS4053 / AI4006	**Course Name:** Recommender System
Course Instructor: Syed Zain Ul Hassan	
Student ID:	**Section:**

Instructions:

- Return the question paper after exam.
- There are **6 questions** on **2 pages** with **3 sides**.
- In case of any ambiguity, you may make assumption. But your assumption should not contradict any statement in the question paper.

Time: 180 minutes **Max Marks**: 100 Points

Question 1: Short Q/A (CLO: 1, 2)	**20 points** (2 x 10)

Briefly answer the following questions:

a) How does information overload affect the recommendation problem?
Information overload i.e., having too many options to choose from, makes it all the more important to have a recommendation algorithm given that it efficiently takes into account user preferences and/or usage patterns.

b) Name any four paradigms of recommender systems.
   i. Collaborative
   ii. Content-based
   iii. Demography-based
   iv. Trust-based

c) Can Collaborative Filtering handle outdated ratings? Justify your answer with a reason.
No. It cannot handle outdated ratings. It only takes into account similarity between users or items without taking into account any temporal information.

d) Explain the terms contextual pre-filtering, contextual modelling, and contextual post-filtering.
Contextual Prefiltering: Filtering out irrelevant items before recommendation engine starts processing.
Contextual Modelling: Filtering out irrelevant items during the recommendation engine process as part of the training.
Contextual Postfiltering: Filtering out irrelevant results after the recommendations have been generated.

e) Why do we use significance weighting?
To take into account not only similarity with other users but also the number of items they have rated. The reason being that not all users are "*important*" enough.

f) What is a long-tail problem?
It is a common problem when most of the users have only rated a very small subset of items while the rest of them remain unrated.

g) What is the difference between model-based and memory-based recommender systems?

Model-based Systems: Use data only once for training and then make predictions without needing to use the data again.

Memory-based Systems: Use the entire data every time a new prediction is to be made.

h) How well does a content-based system perform in terms of scalability?

Content-based systems are scalable in terms of users as only a new user profile is to be constructed.

i) Name two major advantages of Naïve Bayes Filtering over regular Collaborative Filtering.

i. It provides more accurate recommendations compared to vanilla collaborative filtering.

ii. It can provide ranking of predicted items more easily.

j) Will Pearson Correlation be an appropriate similarity measure to use if the underlying distribution of ratings is uniform? Justify your answer with a reason.

No. The correlation coefficient flattens (becomes zero) if the underlying distribution of ratings is uniform.

Question 2: Collaborative Filtering (CLO: 1) **20 points** (8 + 8 + 4)

a) Consider the following interaction matrix extracted from *MovieLens* dataset:

	John Wick	Wall-E	Jaws	Avengers
Rohail	5	1	?	2
Saif	1	5	2	5
Ahsan	2	?	3	4
Qasim	4	3	5	?

Use item-based collaborative filtering (with adj. cosine) to predict ***R*(Ahsan, Wall-E)**.

Solution:

For k = 1:

Mean(Rohail) = 2.66, Mean(Saif) = 3.25, Mean(Ahsan) = 3, Mean(Qasim) = 4

	John Wick	Wall-E	Jaws	Avengers
Rohail	2.34	-1.66	?	-0.66
Saif	-2.25	1.75	-1.25	1.75
Ahsan	-1	?	0	1
Qasim	0	-1	1	?

Sim(Wall-E, John Wick) = -0.923

Sim(Wall-E, Jaws) = -0.987

Sim(Wall-E, Avengers) = 0.921

Avengers are Wall-E are most similar according to Adjust Cosine measure.

***R*(Ahsan, Wall-E)** = (4 x 0.921) / |0.921| = **4**

b) For the interaction matrix presented in Part (a), write Python code snippet to perform the following operations:

I. Read the matrix from ratings.csv into a dataframe named "*inter_matrix*".

```
import pandas as pd
inter_matrix = pd.read_csv('ratings.csv')
```

II. Print all the ratings given by the user "*Saif*".

```
print(inter_matrix.iloc[1].values)
```

c) For the matrix in Part (a), find the user that is most similar to the active user (*Ahsan*) using Pearson Correlation.

Corr(Ahsan, Rohail) = -0.87
Corr(Ahsan, Saif) = 0.960
Corr(Ahsan, Qasim) = 0
Most similar user is Saif.

Question 3: Probabilistic Filtering (CLO: 1) **20 points** (15 + 5)

a) Consider the following matrix which represents data for the performance rating given by auditors to each of the branches. The rating system is {1, 2, 3}.

	Branch 1	Branch 2	Branch 3
Auditor 1	1	1	2
Auditor 2	2	?	1
Auditor 3	3	1	3

Find the missing rating using Naïve Bayes Collaborative Filtering based on users.

Solution:

(Assuming alpha/beta to be 0.05)

P(ri2 = 1) = 2+0.05 / 2+0.05 = 1
P(ri2 = 2) = 0+0.05 / 2+0.05 = 0.024
P(ri2 = 3) = 0+0.05 / 2+0.05 = 0.024
P(X|ri2=1) = (0+0.05 / 2+0.05) x (0+0.05 / 2+0.05) = 0
P(X|ri2=2) = (0+0.05 / 0+0.05) x (0+0.05 / 0+0.05) = 1
P(X|ri2=3) = (0+0.05 / 0+0.05) x (0+0.05 / 0+0.05) = 1

P(ri2=1|X) = 1 x 0 = 0
P(ri2=2|X) = 0.024 x 1 = 0.024
P(ri2=3|X) = 0.024 x 1 = 0.024

We arbitrarily pick P(ri2=2|X) hence **R(Auditor2, Branch2) = 2**

b) Explain the role of serendipity in probabilistic collaborative filtering approach.

Unlike in vanilla collaborative filtering, we cannot control serendipity by adjusting the hyperparameter value in probabilistic techniques. However, one way to control serendipity is by filtering only the relevant users based on their significance or total items rated. Though it still cannot give us full control over the degree of serendipity.

Question 4: Matrix Factorization (CLO: 3) **10 points** (2.5 x 4)

Consider the following matrix for anime recommendation where 1 and -1 represent *like* and *dislike*, respectively. Answer the given questions:

	Bleach	Berserk	Erased	Akira
User 1	1	?	1	-1
User 2	-1	1	1	1
User 3	?	?	-1	-1
User 4	-1	1	1	?

a) Is it possible to make more than three factors of this matrix? Defend your answer with a proof.

Let us assume that we want to break the matrix into four matrices such that:

M = F1 * F2 * F3 * F4

The factors we can make are:

F1 is a 4x2 matrix

F2 is a 2x2 matrix

F3 is a 2x1 matrix

F4 is a 1x4 matrix

b) For what values of *m, k, l* and *n*, will the three factors (*m* x *k*), (*k* x *l*) and (*l* x *n*) consume less memory in total than the original interaction matrix?

For m = 4, k = 1, l = 2, n = 4 i.e., factors (4x1) (1x2) (2x4) it will take a storage size of 14 compared to the original matrix of size 16.

c) We used interactions instead of ratings in the given matrix. Can this approach help gradient descent converge quicker?

Yes. Using binary ratings (interaction) can help the algorithm converge faster due to reduced feasible solution space.

d) Does matrix factorization suffer from cold-start problem as severely as Collaborative Filtering? Justify your answer.

Matrix factorization does suffer from cold-start problem and in many cases, it is even more severe than collaborative filtering. To find meaningful factors, we require the matrix to be as less sparse as possible in matrix factorization technique.

Question 5: Content-based Filtering (CLO: 2) **20 points** (15 + 5)

a) Consider the following interaction table for news articles and keywords where "*Article 2*" is the current (active) item. The table is populated with pre-computed TF-IDF values. Perform the given tasks.

	Covid-19	Tesla	Economy	Export
Article 1	3.05	5.07	0.10	1.48
Article 2	0.30	3.39	5.06	0.90
Article 3	0.40	4.66	0.30	1.32
Article 4	0.44	2.50	1.30	0.07

I. Calculate normalized TF-IDF values for this matrix.

	Covid-19	Tesla	Economy	Export	Magnitude
Article 1	3.05	5.07	0.10	1.48	6.099
Article 2	0.30	3.39	5.06	0.90	6.164
Article 3	0.40	4.66	0.30	1.32	4.869
Article 4	0.44	2.50	1.30	0.07	2.852

	Covid-19	Tesla	Economy	Export	Magnitude
Article 1	0.50	0.83	0.01	0.24	6.099
Article 2	0.04	0.54	0.82	0.14	6.164
Article 3	0.08	0.95	0.06	0.27	4.869
Article 4	0.15	0.87	0.45	0.02	2.852

II. Find the article that should be recommended to the user.

Sim(A2, A1) = 0.514

Sim(A2, A3) = 0.612

Sim(A2, A4) = 0.861

Based on similarity **Article 4** should be recommended to the user.

b) For the interaction table presented in Part (a), write Python code snippet to save it to a dataframe named "*news*" and then calculate the pairwise cosine similarity on it.

```
import pandas as pd
from sklearn.metrics.pairwise import cosine_similarity
news = pd.DataFrame(interaction_matrix)
similarities = cosine_similarity(news)
```

Question 6: Neural Recommendations (CLO: 4) **10 points** (5 + 5)

a) Explain why the following ratings data cannot be properly split into training and testing samples for neural recommendations:

	I1	*I2*	*I3*	*I4*	*I5*	*I6*	*I7*	*I8*	*I9*	*I10*
U1	2	1		3	5	4		2	1	2
U2										
U3	4	5		3	2	4			5	1
U3		1	4	3	4	2		1	1	5
U5	4	2		1	3	2		1	2	3
U6	3			1		2		3	2	2

Note: *Each row represents a unique user and each column is a unique item.*

The given dataset has a missing record for I7 and U2. We cannot split it as is into training and testing sets unless we preprocess it and handle the missing values.

b) Explain why we are usually interested in interactions more than ratings when working with recommendations generated by a GAN.

Training a GAN and its convergence is resource intensive. If we use ratings instead of interactions the resource requirements will significantly be more. Moreover, in some cases we want to know whether the user would like to interact with a particular item more than we are concerned with the rating they would give.

--. --- --- -.. / .-.. ..- -.-. -.-