PHASE 1

In phase 1, we learn about recommendation system and its theoretical background.

Question 1

Explain Recommender Systems according to the references.

Question 2

Write some of the applications of recommender systems in different areas.

Question 3

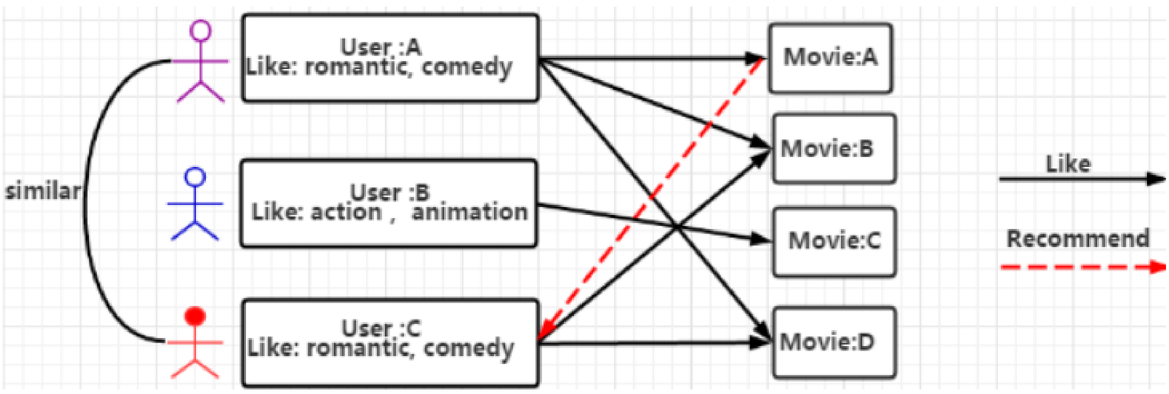
There are various challenges faced by Recommendation Systems. Write these challenges.

Question 4

Explain similarity-based methods for recommender systems.

Question 5

Explain classes of hybridization.



Question 6

In this question, we learn the utility matrix.

Question 7 continued on next page...

(a) What is the utility matrix?

(b) For real world data, is it an sparse matrix?

(c) Is this sentence true? "The goal of a recommendation system is to predict all the blanks in the utility matrix"

(d) Write the two general approaches to discovering the value users place on items to build the utility matrix.

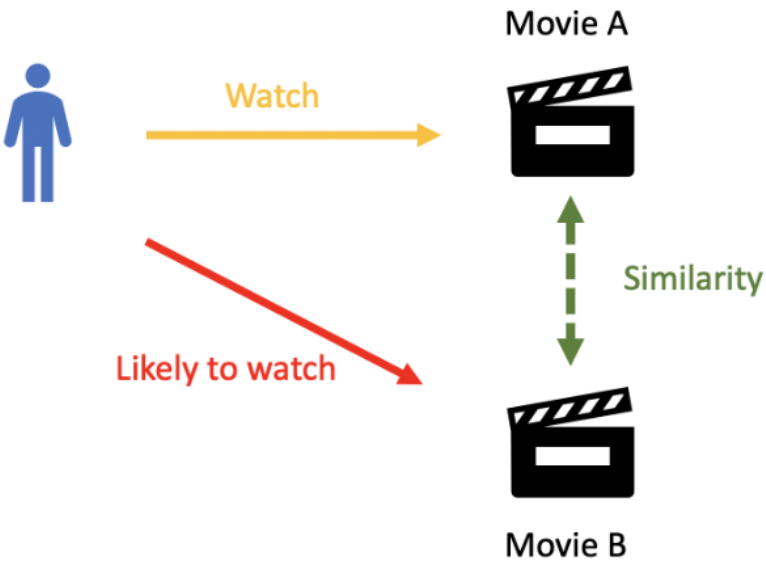
Question 7

In this question, we investigate content-based recommendations.

(a) Explain content-based recommendations.

In a content-based system, we must construct for each item a profile, which is a record or collection of records representing important characteristics of that item.

(b) Write 10 examples of features of a movie that might be relevant to a recommendation system.



Representing Item Profiles: Our ultimate goal for content-based recommendation is to create both an item profile consisting of feature-value pairs and a user profile summarizing the preferences of the user, based of their row of the utility matrix. We shall try to generalize this vector approach to all sorts of features. It is easy to do so for features that are sets of discrete values. For example, if one feature of movies is the set of actors, then imagine that there is a component for each actor, with 1 if the actor is in the movie, and 0 if not. Likewise, we can have a component for each possible director, and each possible genre. All these features can be represented using only 0's and 1's. There is another class of features that is not readily represented by Boolean vectors: those features that are numerical. For instance, we might take the average rating for movies to be a feature.

(c) Suppose the only features of movies are the set of actors and the average rating. Consider two movies with five actors each. Two of the actors are in both movies. Also, one movie has an average rating of 3 and the other an average of 4. Write the vectors and calculate the cosine of the angle between the vectors and then discuss about it.

User Profile: We not only need to create vectors describing items; we need to create vectors with the same components that describe the user's preferences

(d) Suppose items are movies, represented by Boolean profiles with components corresponding to actors. Also, the utility matrix has a 1 if the user has seen the movie and is blank otherwise. If 20 percent of the movies that user U likes have Julia Roberts as one of the actors, then what will be the user profile for U? Why?

If the utility matrix is not Boolean, e.g., ratings 1–5, then we can weight the vectors representing the profiles of items by the utility value. It makes sense to normalize the utilities by subtracting the average value for a user. That way, we get negative weights for items with a below-average rating, and positive weights for items with above-average ratings.

(e) Consider the same movie information as in part d, but now suppose the utility matrix has non-blank entries that are ratings in the 1–5 range. Suppose user U gives an average rating of 3. There are three movies with Julia Roberts as an actor, and those movies got ratings of 3, 4, and 5. On the other hand, user V gives an average rating of 4, and has also rated three movies with Julia Roberts (it doesn't matter whether or not they are the same three movies U rated). User V gives these three movies ratings of 2, 3, and 5. Then, what will be the component for Julia Roberts value in user profile of U and V?

Three movies m_1, m_2 and m_3 has the following values for features f_1, f_2 and f_3

	m_1	m_2	m_3
f_1	2.98	2.55	2.87
f_2	400	250	600
f_3	5	3	5

Table 1

We may imagine these values as defining a vector for each movie; for instance, m_1's vector is [2.98, 400, 5]. We can compute the cosine distance between any two of the vectors, but if we do not scale the components, then the second feature(f_2) will dominate and make differences in the other components essentially invisible. Let us use 1 as the scale factor for feature f_1, α for feature f_2, and β for feature f_3.

(f) In terms of α and β, compute the cosines of the angles between the vectors for each pair of the three movies.

(g) What are the angles between the vectors if $\alpha = \beta = 1$?

(h) What are the angles between the vectors if $\alpha = 0.01$ and $\beta = 0.5$?

(i) One fair way of selecting scale factors is to make each inversely proportional to the average value in its component. What would be the values of α and β, and what would be the angles between the vectors?

An alternative way of scaling components of a vector is to begin by normalizing the vectors. That is, compute the average for each component and subtract it from that component's value in each of the vectors.

(j) Normalize the vectors for the three movies described in the previous question.

When all components are nonnegative, as they are in the data of the previous question, no vectors can have an angle greater than 90 degrees. However, when we normalize vectors, we can (and must) get some negative components, so the angles can now be anything, that is, 0 to 180 degrees. Moreover, averages are now 0 in every component, so the suggestion in part (j) that we should scale in inverse proportion to the average makes no sense.

(k) Suggest a way of finding an appropriate scale for each component of normalized vectors. How would you interpret a large or small angle between normalized vectors? What would the angles be for the normalized vectors derived from the data in the previous question?

A certain user has rated the three movies m1, m2 and m3 as follows:

m_1 ★★★★
m_2 ★★
m_3 ★★★★★

(l) Normalize the ratings for this user.

(m) Compute a user profile for the user, with components for f_1, f_2 and f_3, based on the data of table1.

Question 8

In this question, we take up a significantly different approach to recommendation. Instead of using features of items to determine their similarity, we focus on the similarity of the user ratings for two items. That is, in place of the item-profile vector for an item, we use its column in the utility matrix. Further, instead of contriving a profile vector for users, we represent them by their rows in the utility matrix. Users are similar if their vectors are close according to some distance measure such as Jaccard or cosine distance. Recommendation for a user U is then made by looking at the users that are most similar to U in this sense, and recommending items that these

Question 8 continued on next page...

users like. The process of identifying similar users and recommending what similar users like is called collaborative filtering.
Table 2 is a utility matrix, representing the ratings, on a 1–5 star scale, of eight items, m_1 through m_8, by three user u_1, u_2, and u_3. Compute the following from the data of this matrix.

	m1	m2	m3	m4	m5	m6	m7	m8
u1	5	4		4	1		3	2
u2		2	4	3	1	2	1	
u3	2		1	3		4	4	3

Table 2

(a) Treating the utility matrix as boolean, compute the Jaccard distance between each pair of users.

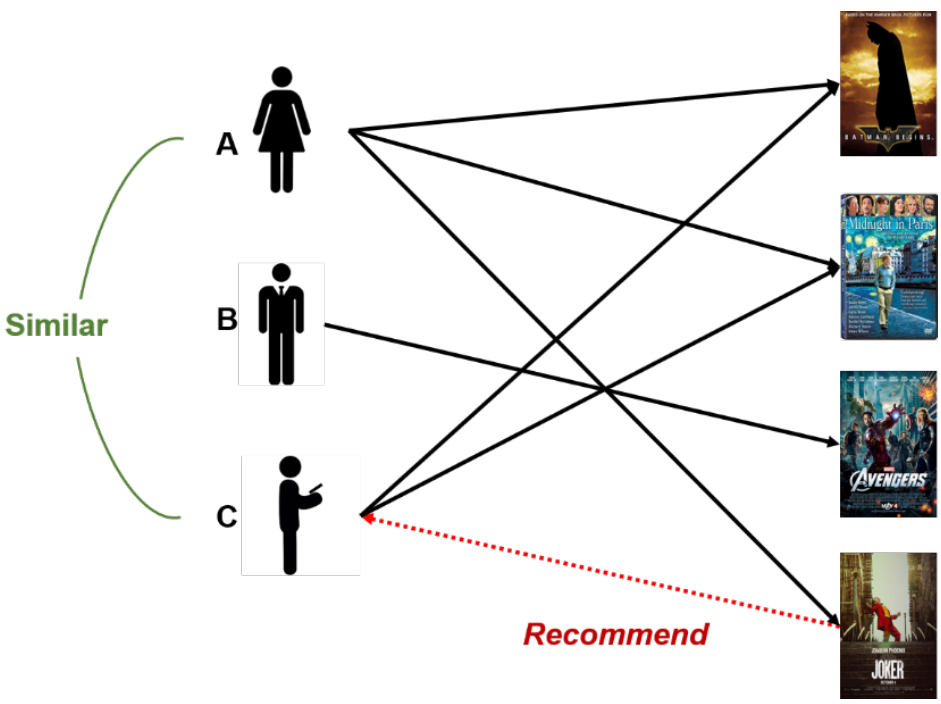
(b) Repeat Part (a), but use the cosine distance

(c) Treat ratings of 3, 4, and 5 as 1 and 1, 2, and blank as 0. Compute the Jaccard distance between each pair of users.

(d) Repeat Part (c), but use the cosine distance.

(e) Normalize the matrix by subtracting from each nonblank entry the average value for its user.

(f) Using the normalized matrix from Part (e), compute the cosine distance between each pair of users.

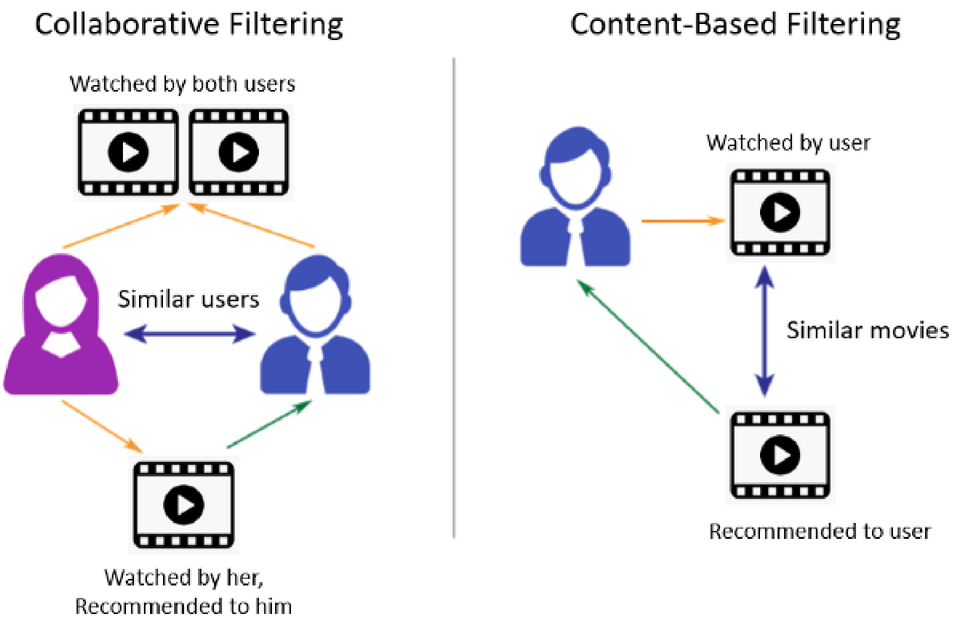


Now we cluster movies in the matrix of Table 2. Do the following steps.

(g) Cluster the eight movies hierarchically into four clusters. The following method should be used to cluster. Replace all 3's, 4's, and 5's by 1 and replace 1's, 2's, and blanks by 0. Use the Jaccard distance to measure the distance between the resulting column vectors. For clusters of more than one element, take the distance between clusters to be the minimum distance between pairs of elements, one from each cluster.

(h) Then, construct from the original matrix of Table 2 a new matrix whose rows correspond to users, as before, and whose columns correspond to clusters. Compute the entry for a user and cluster of items by averaging the nonblank entries for that user and all the items in the cluster.

(i) Compute the cosine distance between each pair of users, according to your matrix from Part(b).



Question 9

Explain Dimensionality Reduction (with examples).