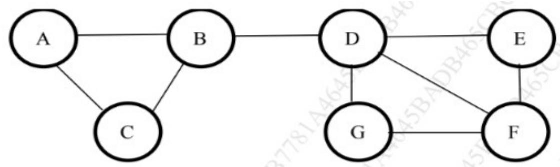
MODULE 6

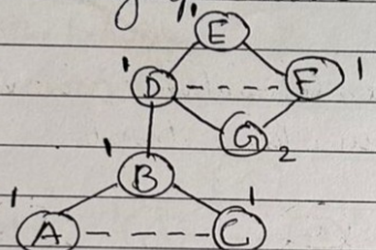
DEC' 22

Q5 a) Determine communities for the given social network graph using Girvan- Newman algorithm. [10]



- To calculate shortest path for finding between edges, we use Girvann newman algorithm, which visits each ode X once & computes the no of shortest paths from X to each of the other nodes that go through each of the edges.
- The algorithm begins by performing a BFS of the graph from node X.
- Edges going between node at the same level can never be part of shortest path from X & Edges DAG edge will be part of atleast 1 shortest path from root X ~~form~~ for the graph above.

Step1 -



Label Root E with 1. At level 1 are the nodes D & F.. Each has only E as a parent; so they too are labelled 1. Nodes B & G are at level 2. B has only D as the parent, so B's label is same as the label of D, which is 1. However, G has parents D & F, so its label is sum of their labels or 2. Finally, at level 3, A & C each have only parent B, so their labels are the label of B, which is 1.

Step2 - Credit Calculation

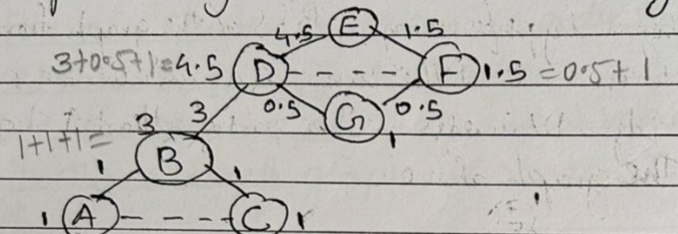
• Each node other than root is given a credit of 1. This credit may be divided among nodes & edges above.

Rules for calculation:-

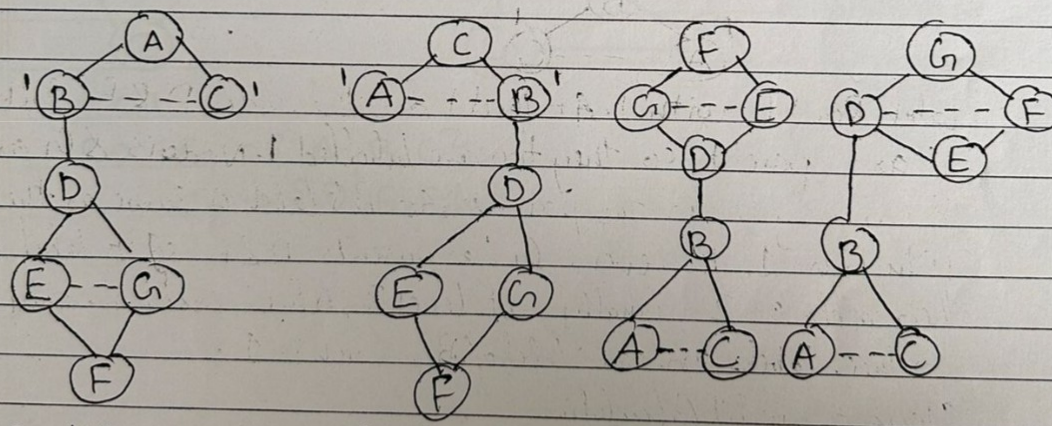
(1) Each leaf in DAG gets a credit

(2) Each non leaf node gets a credit equals to: 1 + Σ credits of DAG Edges from that node to the level.

- DAG edge e entering node Z from the level above is given a share of credit Z, proportional to the fraction of shortest path from the root to Z that go through e.
- After performing credit, calculate with each node as root, we sum credits for each edge. As each shortest path will have been discovered twice, we must divide the result/credit for each edge by 2.
- Graph showing Calculation for DAG, starting from node E.



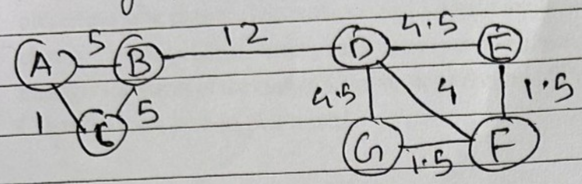
- To complete the betweenness calculation, we have to repeat this calculation for every node as the root & sum the contributions



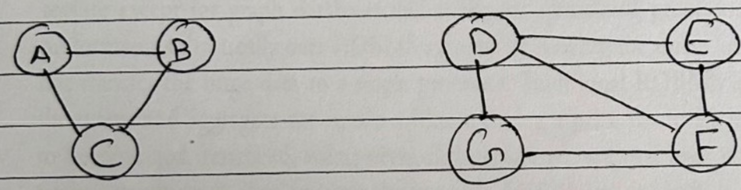
Root A   Root C   Root F   Root G

Step 3 Using betweenness to find communities.

Final graph with betweenness value



- We can cluster by taking in order to increase betweenness & add them to the graph at a time.
- We can remove edge with highest value to cluster the graph.
- In the example graph, we remove BD to get 2 communities as follows:



C1

So $C_1 = \{A, B, C\}$

$C_2 = \{D, E, F, G\}$

Q.6 b) How recommendation is done based on properties of product? Elaborate with a suitable example. [10]

Content-Based Recommendation Systems

Content-based recommendation systems focus on the properties of items themselves to determine similarities and make recommendations. This approach works by creating profiles for each item, which are collections of records representing important characteristics of that item.

Key points to consider:

- - Each item gets a profile with relevant characteristics
- - Similarity between items is measured based on these characteristics
- - Users are recommended items that match their preferences based on item properties

Example: Movie Recommendations

Let's consider a movie recommendation system as an example:

1. Item Profiles:
   - Movie "The Shawshank Redemption"
     - Actors: Tim Robbins, Morgan Freeman
     - Director: Frank Darabont
     - Year: 1994
     - Genre: Drama

   - Movie "Goodfellas"
     - Actors: Robert De Niro, Joe Pesci, Ray Liotta
     - Director: Martin Scorsese
     - Year: 1990
     - Genre: Crime/Drama

2. User Preferences:
   - User A prefers movies directed by Frank Darabont
   - User B likes crime dramas from the 1990s

3. Recommendation Logic:
   - For User A, "The Shawshank Redemption" matches their preference for Frank Darabont's direction
   - For User B, "Goodfellas" fits their interest in 1990s crime dramas

Implementation

In a real-world implementation, steps followed:

1. Collect product data (in this case, movie attributes)
2. Store this data in a database or knowledge base
3. Create algorithms to compare item profiles and user preferences
4. Generate personalized recommendations based on these comparisons

Summary and Best Practices

Content-based recommendation systems rely on item properties to make recommendations. They are particularly useful when item attributes are readily available and meaningful to users. Some best practices include:

- - Choose relevant attributes for your items
- - Regularly update item profiles with new information
- - Consider combining content-based recommendations with collaborative filtering for more robust results
- - Test different similarity measures to optimize recommendations

By leveraging item properties, content-based systems can provide targeted recommendations that align closely with user interests and preferences.

Real world example:

Goodreads: A Real-World Application of Content-Based Recommendation Systems

Goodreads is an excellent example of how content-based recommendation systems are used in a real-world application to enhance user experience and engagement 1.

Key Features of Goodreads' Recommendation System:

- Book Matching: Goodreads suggests books centered around a user's reading history, ratings, and reviews
- Personalization: The algorithm considers factors such as:
  - User's reading habits
  - Preferred genres
  - Favorite authors
- Continuous Learning: As users interact with books and leave reviews, the system updates its understanding of their preferences over time.

How It Works:

- Data Collection: Goodreads collects data on books, including attributes like author, genre, publication year, and publisher.
- User Profiles: Users create profiles based on their reading history, ratings, and reviews.
- Similarity Measures: The system uses algorithms to compare book attributes with user preferences.
- Recommendation Generation: Based on these comparisons, Goodreads suggests books that match the user's interests and reading patterns.

Example Scenario:

Let's say Sarah has been actively using Goodreads for a few months. She has read several books in the fantasy genre, particularly enjoying authors like J.R.R. Tolkien and George R.R. Martin. Here's how Goodreads might work for her:

- Initial Recommendations: When Sarah first joins, she might see recommendations based on popular fantasy books.
- Personalized Suggestions: As she reads and reviews books, Goodreads starts suggesting titles that match her preferences:
  - Books similar to Tolkien's works
  - New releases in the fantasy genre
  - Authors writing in a style similar to G.R.R. Martin
- Dynamic Updates: If Sarah suddenly expresses interest in historical fiction, Goodreads might start recommending books in this new genre.
- Author-Specific Recommendations: If Sarah starts following a new author, Goodreads might suggest other books by that author or similar authors.

Benefits for Users:

- Time-Saving: Users don't have to manually search for books they might enjoy.
- Discovery: They can discover new authors and genres beyond their usual interests.
- Community Engagement: Users can share and discuss book recommendations with others who have similar tastes.

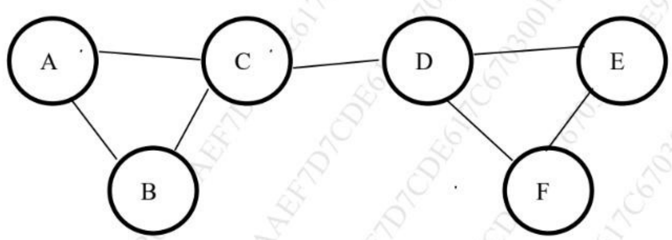
Technical Implementation:

- Natural Language Processing (NLP) to analyze book descriptions and user reviews.
- Machine learning models to learn user preferences over time.
- Database systems to store and retrieve vast amounts of book and user data efficiently.

This example demonstrates how content-based recommendation systems can be effectively applied in real-world applications, providing personalized experiences for users across various domains.

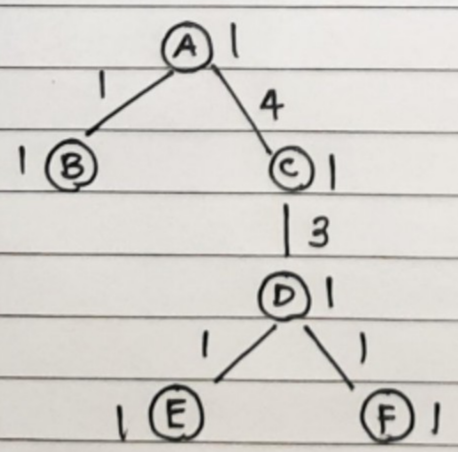
MAY' 23

Q5 a) Determine communities for the given social network graph using Girvan-Newman algorithm. **[10]**



Edges	EDGE BETWEENESS						Total
	A	B	C	D	E	F	
AB	1	1	0	0	0	0	2
BC	0	4	1	1	1	1	8
AC	4	0	1	1	1	1	8
CD	3	3	3	3	3	3	18
DE	1	1	1	1	4	0	8
EF	0	0	0	0	1	1	2
FD	1	1	1	1	0	4	8

step 1 : consider node A as the root node .



- $ED = \frac{1}{1} = 1$

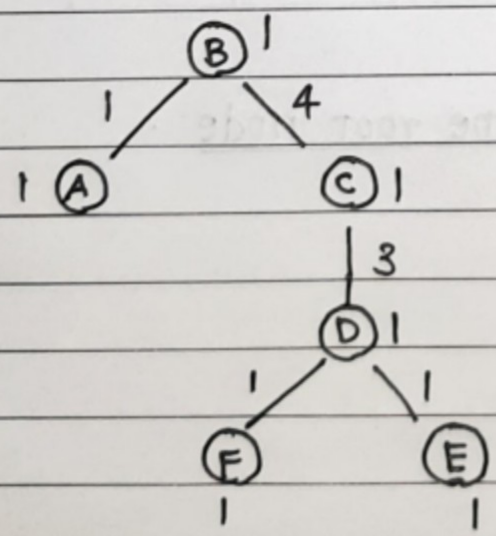
- $FD = \frac{1}{1} = 1$

- $DC = \frac{1+1+1}{1} = 3$

- $CA = \frac{1+3}{1} = 4$

- $BA = \frac{1}{1} = 1$

- <u>step 2 :</u> <u>consider node B as root node.</u>



- $FD = \frac{1}{1} = 1$

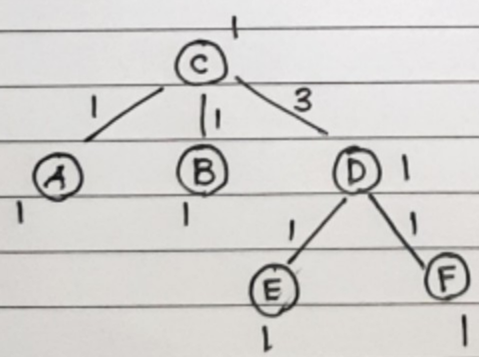
- $ED = \frac{1}{1} = 1$

- $DC = \frac{1+1+1}{1} = 3$

- $CB = \frac{1+3}{1} = 4$

- $AB = \frac{1}{1} = 1$

- Consider Node C as root Node.



- $ED = \frac{1}{1} = 1$

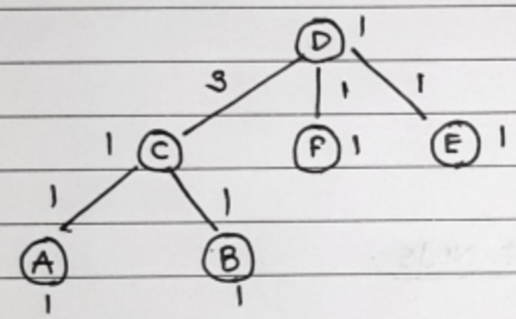
- $FD = \frac{1}{1} = 1$

- $DC = \frac{1+1+1}{1} = 3$

- $BC = \frac{1}{1} = 1$

- $AC = \frac{1}{1} = 1$

- **Step 4: Consider Node D as root Node.**



- $AC = \frac{1}{1} = 1$

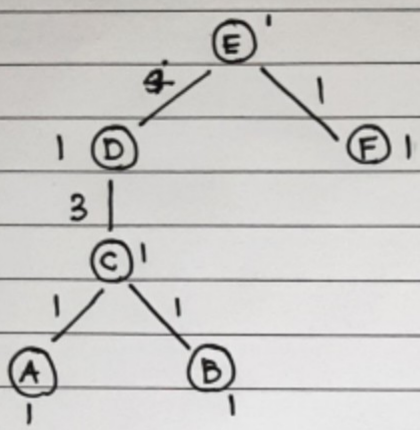
- $BC = \frac{1}{1} = 1$

- $CD = \frac{1+1+1}{1} = 3$

- $ED = \frac{1}{1} = 1$

- $FD = \frac{1}{1} = 1$

- Step 5 : Consider Node E as the root Node.



- $AC = \frac{1}{1} = 1$

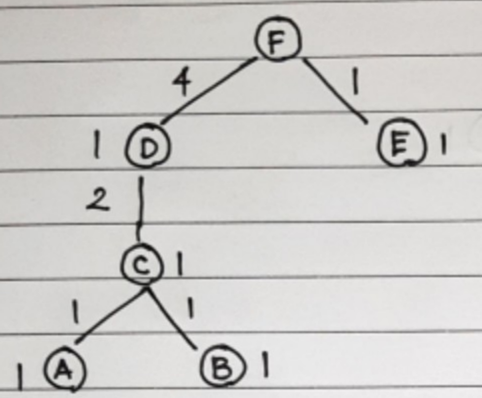
- $BC = \frac{1}{1} = 1$

- $CD = \frac{1+1+1}{1} = 3$

- $DE = \frac{1+3}{1} = 4$

- $FE = \frac{1}{1} = 1$

- Consider Node F as root Node.



- $AC = \frac{1}{1} = 1$

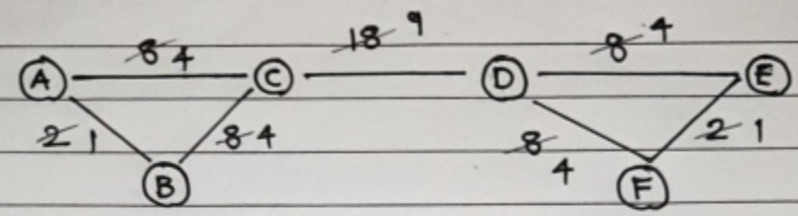
- $BC = \frac{1}{1} = 1$

- $CD = \frac{1+1+1}{1} = 3$

- $DF = \frac{1+3}{1} = 4$

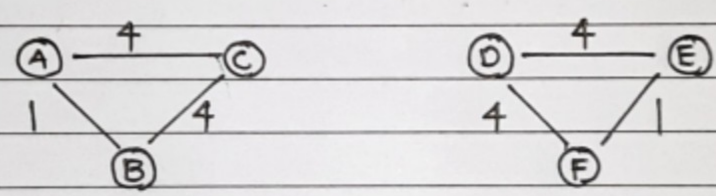
- $EF = \frac{1}{1} = 1$

- Now mapping the values,



since the graph is undirected, divide the edge sum by 2

Eliminate edge with highest sum value.



These are the final communities.

Q.6 b) Define collaborative filtering. Using an example of an e-commerce site like flipkart or amazon describe how it can be used to provide recommendation to users. **[10]**

Collaborative filtering is a technique used in recommendation systems to predict the preferences of a user by analyzing the behavior of similar users. Here's a detailed explanation of collaborative filtering, along with an example of how it can be used in an e-commerce site like Flipkart or Amazon:

Definition of Collaborative Filtering

Collaborative filtering is a method of generating recommendations based on the behavior of similar users. The key aspects of collaborative filtering are:

1. It identifies similarities between customers based on their site interactions .
2. It recommends relevant products to customers across digital properties .
3. It doesn't use features of the item itself to make recommendations; instead, it focuses on the similarity of user ratings for two items .

4. Users are grouped into clusters based on their similar browsing or purchasing habits .

How Collaborative Filtering Works

The process of collaborative filtering typically involves the following steps:

1. Data Collection: Gather user interaction data (e.g., purchase history, browsing patterns).
2. User Profiling: Create profiles for each user based on their interactions.
3. Similarity Measurement: Determine the similarity between users or items.
4. Recommendation Generation: Provide recommendations based on the behavior of similar users or items.

Example: Collaborative Filtering in E-commerce (Flipkart/Amazon)

Let's consider a scenario where a user named Rohan is shopping on Flipkart:

1. Data Collection: Flipkart tracks Rohan's browsing and purchase history.

2. User Profiling: Flipkart creates a profile for Rohan based on his interactions:
   - Rohan has purchased smartphones and accessories in the past.
   - He tends to browse products in the electronics category.
   - His average purchase price is around ₹25,000.

3. Similar User Identification: Flipkart identifies users who have similar browsing and purchasing patterns to Rohan:
   - User A: Also purchases smartphones and accessories.
   - User B: Frequently browses electronics but hasn't made a purchase yet.
   - User C: Purchases high-end smartphones regularly.

4. Recommendation Generation: Based on these similarities, Flipkart generates recommendations for Rohan:

   - Product Recommendations:
     - Latest smartphone models
     - High-quality phone cases
     - Wireless earbuds compatible with recent smartphone models

   - Category Recommendations:
     - Electronics
     - Mobile Accessories

   - Price Range Recommendations:
     - ₹15,000 - ₹35,000 (based on Rohan's average purchase price)

5. Dynamic Updates: As Rohan interacts with these recommendations, Flipkart continuously updates his profile and adjusts future recommendations accordingly.

Benefits of Collaborative Filtering in E-commerce

1. Personalization: Recommendations are tailored to individual user preferences.
2. Improved User Experience: Users discover products they might not have found otherwise.
3. Increased Sales: Relevant recommendations can lead to higher conversion rates.
4. Cold Start Tolerance: Can provide initial recommendations even with minimal user data.

Challenges and Considerations

1. Privacy Concerns: Users may be hesitant to share their browsing history.
2. Data Sparsity: Initial recommendations may be less accurate with limited user data.
3. Cold Start Problem: Recommending to new users with little interaction data.
4. Shilling Attacks: Malicious users could manipulate the system by artificially inflating ratings.

EXTRA

There are several limitations to the accuracy of collaborative filtering recommendations.

1. Latency Problem

The latency problem is specific to collaborative filtering approaches and occurs when new items are frequently inserted into the database . This leads to:

- Failure to recommend new items promptly
- Reduced system performance due to frequent updates

Solutions include:
- Performing calculations in an offline environment
- Using clustering-based techniques to increase performance

2. Scalability Problem

As the amount of data increases rapidly, scalability becomes a significant challenge :

- Recommender systems require large amounts of training data
- Processing becomes slower as the database grows larger

Solutions include:
- Dimensionality reduction techniques
- Clustering-based approaches to group similar users

3. Sparsity Problem

Data sparsity occurs when active users rate very few items, reducing recommendation accuracy . This leads to:

- Limited information for accurate predictions
- Difficulty in identifying patterns in sparse data

Solutions include:
- Demographic filtering
- Singular Value Decomposition (SVD)
- Model-based collaborative techniques

4. Synonymy Problem

Similar or related items may have different entries or names, leading to reduced accuracy . Examples include "babywear" and "baby cloth".

Solutions include:
- Automatic term expansion
- Singular Value Decomposition (SVD)
- Demographic filtering

5. Cold Start Problem

Collaborative filtering struggles with recommending to new users who have little interaction data .

Solutions include:
- Content-based filtering for initial recommendations
- Hybrid approaches combining collaborative and content-based filtering

6. Shilling Attacks

Malicious users could artificially inflate ratings to manipulate the system .

Solutions include:
- Anomaly detection algorithms
- Diverse user groups for recommendations
- Regular audits of user behavior

7. Context-Awareness Limitations

Traditional collaborative filtering often fails to capture contextual factors that influence preferences .

Solutions include:
- Incorporating contextual information (time, location, device)
- Hybrid approaches combining collaborative filtering with content-based filtering

8. Overfitting

The model may become too specialized to the training data, failing to generalize well to new situations .

Solutions include:
- Cross-validation techniques
- Regularization methods
- Ensemble methods combining multiple models

9. Privacy Concerns

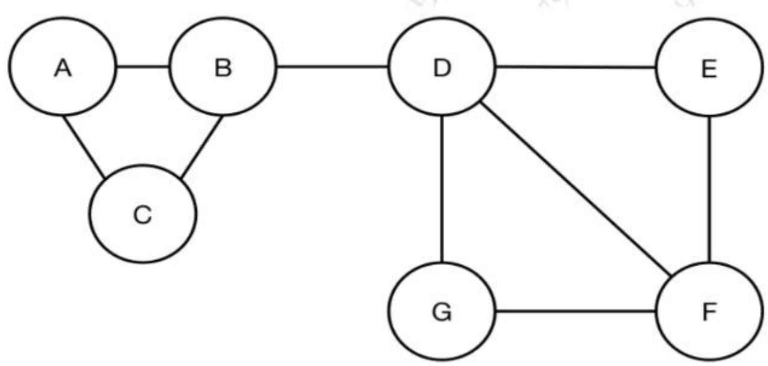
Users may be hesitant to share their browsing history due to privacy concerns .

Solutions include:
- Anonymous data collection
- Differential privacy techniques
- Opt-in/opt-out mechanisms for data sharing

While collaborative filtering remains a powerful technique for personalized recommendations, these limitations highlight areas where improvements can be made. Addressing these challenges often involves combining collaborative filtering with other techniques or incorporating domain-specific knowledge. The effectiveness of any recommendation system depends on carefully balancing these factors and continuously refining the approach based on user feedback and performance metrics.

DEC' 23

Q5 a) Determine communities for the given social network graph using Girvan-Newman algorithm. [10]



b) How recommendation is done based on properties of the product? Explain with the help of an example. [10]

Q.6

Content-based filtering is a recommendation technique that focuses on the attributes or features of products to determine similarities and make recommendations. This approach differs from collaborative filtering, which relies on user behavior data.

Key aspects of content-based filtering include:

1. Product Profiles: Each product is represented by a set of attributes or features. These can include text descriptions, genres, keywords, or any other relevant characteristics of the item.

2. User Profiles: Users are typically represented by their preferences or interests, which are derived from their interaction history with products.

3. Similarity Measurement: The system calculates the similarity between product profiles and user profiles based on shared attributes.

4. Recommendation Generation: Items that match the user's preferences are selected for recommendation.

Implementation of content-based filtering involves several steps:

1. Feature Extraction: Relevant features are extracted from product descriptions and other attributes. This often employs natural language processing techniques like TF-IDF (Term Frequency-Inverse Document Frequency) vectorization.

2. User Profile Creation: User preferences are identified based on their interaction history with products.

3. Similarity Computation: Cosine similarity or other distance metrics are used to measure the similarity between product and user profiles.

4. Recommendation Generation: Products with the highest similarity scores to the user's profile are selected for recommendation.

This approach offers several advantages:

1. Cold start tolerance: It can work well even with limited initial data.
2. Transparency: Users can understand why they're being recommended certain products.
3. Scalability: It can handle new products easily once their attributes are known.

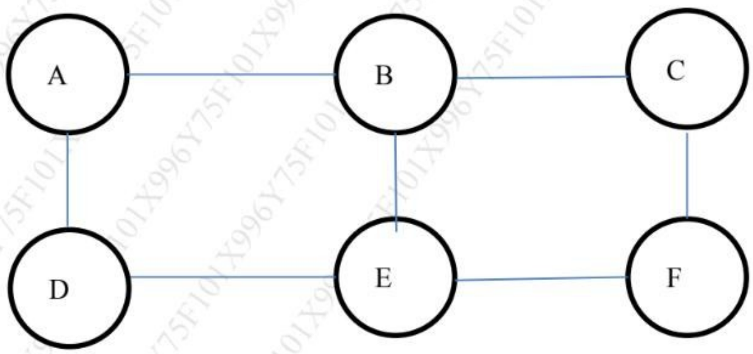
However, content-based filtering also has limitations:

1. Sparsity problem: It struggles with sparse data where users haven't interacted much with similar products.
2. Overfitting: It may become too specialized to training data.
3. Lack of diversity: It might recommend very similar products repeatedly.

To address these challenges, hybrid approaches combining content-based filtering with collaborative filtering techniques are often employed in real-world recommendation systems.

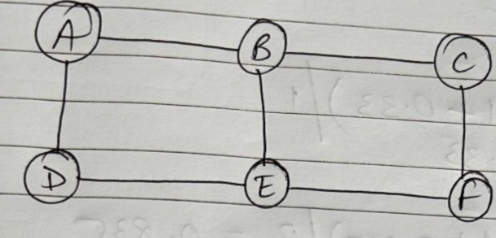
MAY' 24

Q5 a) Determine communities for the given social network graph using Girvan- Newman algorithm. [10]



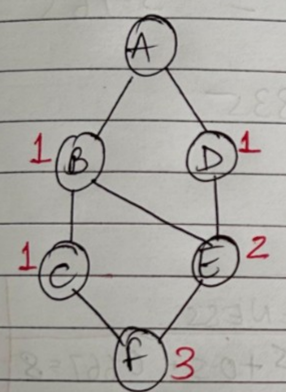
GN Algo.

May '24



Step 1

Start with Node A.



The given graph has 6 nodes and 7 edges so 6 iterations will be performed including each node as a starting point.

Assign score to every node based on the number of paths available to reach the respective node from the root node.

Assign score to every edge by taking the ratio and moving backwards.

FC = (node score of C) / (node score of F)
= 1/3
= 0.33

PAGE No.
DATE / /

$FE = 2/3 = 0.667$

$CB = (\text{node score of } B + \text{edge score of } CF) / \text{node score of } C$

$= (1 + 0.33)/1$

$CB = 1.33$

$EB = (1 + 0.667)/2 = 0.835$

$ED = (1 + 0.667)/2 = 0.835$

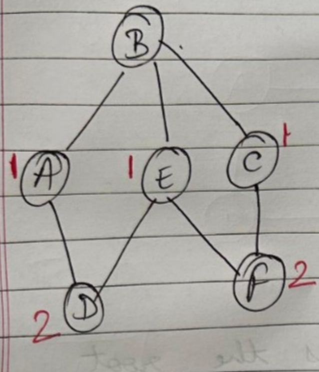
$BA = (1 + 1.33 + 0.835)/1 = 3.165$

$DA = (1 + 0.835)/1 = 1.835$

EDGES	EDGE BETWEENESS.
AB	$3.165 + 1.5 + 1.33 + 0.835 + 0.5 + 0.667 = 8$
AD	$1.835 + 0.5 + 0.33 + 1.835 + 0.5 + 0.33 = 5.33$
BC	$1.33 + 1.5 + 3.165 + 0.667 + 0.5 + 0.835 = 8$
BE	$0.835 + 2 + 0.835 + 0.835 + 2 + 0.835 = 7.34$
CF	$0.33 + 0.5 + 1.835 + 0.33 + 0.5 + 1.835 = 5.33$
DE	$0.835 + 0.5 + 0.667 + 3.165 + 1.5 + 1.33 = 8$
EF	$0.667 + 0.5 + 0.835 + 1.33 + 1.5 + 3.165 = 8$

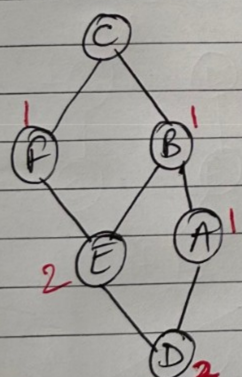
A B C D E F

Step 2 Consider node B as a Root



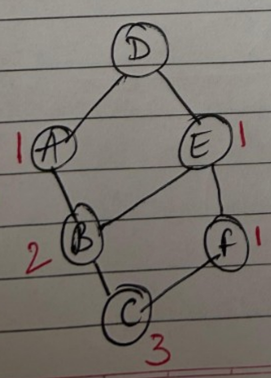
$DA = 1/2 = 0.5$
$DE = 1/2 = 0.5$
$FE = 1/2 = 0.5$
$FC = 1/2 = 0.5$
$AB = (1+0.5)/1 = 1.5$
$EB = (1+0.5+0.5)/1 = 2$
$CB = (1+0.5)/1 = 1.5$

Step 3 Consider node C as the root



$DE = 2/3 = 0.667$
$DA = 1/3 = 0.33$
$EF = (1+0.667)/2 = 0.835$
$EB = (1+0.667)/2 = 0.835$
$AB = (1+0.33)/1 = 1.33$
$FC = (1+0.835)/1 = 1.835$
$BC = (1+0.835+1.33)/1 = 3.165$

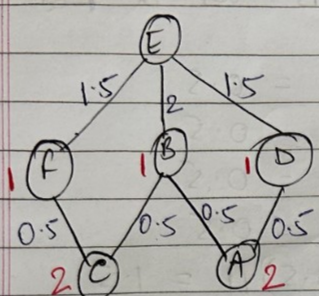
Step 4 Consider node D as the root



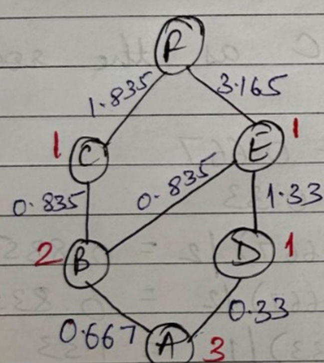
$CB = 2/3 = 0.667$
$CF = 1/3 = 0.33$
$BA = (1+0.667)/2 = 0.835$
$BE = (1+0.667)/2 = 0.835$
$FE = (1+0.33)/1 = 1.33$
$AD = (1+0.835)/1 = 1.835$
$DE = (1+0.835+1.33)/1 = 3.165$

PAGE No.
DATE / /

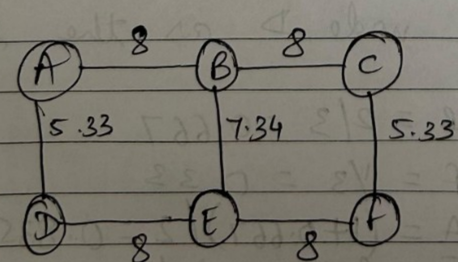
Step 5 Considering E as the ~~node~~ root



Step 6 Considering F as the root

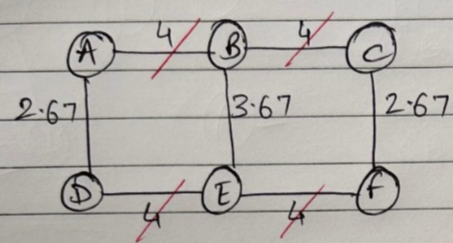


Final Graph.

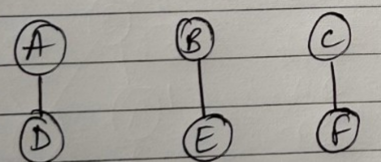


Since it is undirected graph, we divide the score by 2.

final graph with EBC Score



The edges with the highest score will be deleted and sub-graphs will be made which are called as communities.



Q.6 b) Describe collaborative filtering in recommendation system. [10]

Collaborative filtering is a recommendation technique that predicts the preferences of a user by analyzing the behavior of similar users.
Collaborative filtering is a method of generating recommendations based on the behavior of similar users . The core idea is that users who have similar preferences tend to rate items similarly .

Key Components of Collaborative Filtering

1. User Profiles: Each user is represented by a profile based on their ratings or interactions with items .

2. Item Profiles: Items are also represented by profiles based on how users have rated them .

3. Similarity Measurement: Algorithms calculate similarities between user profiles or item profiles .

4. Recommendation Generation: Recommendations are generated based on the behavior of similar users or items .

How Collaborative Filtering Works

The collaborative filtering process typically involves these steps:

1. Data Collection: Gather user interaction data (e.g., ratings, purchases, browsing history) .

2. User Profiling: Create profiles for each user based on their interactions .

3. Similar User Identification: Find users who have similar rating patterns to the target user .

4. Rating Prediction: Predict the rating a user might give to an unrated item based on the ratings of similar users .

5. Recommendation Generation: Provide recommendations based on predicted ratings .

Types of Collaborative Filtering

There are two main types of collaborative filtering:

1. User-based Collaborative Filtering: Finds similar users and recommends items liked by those similar users .

2. Item-based Collaborative Filtering: Finds similar items and recommends items similar to ones the user likes .

Advantages of Collaborative Filtering

1. Personalization: Provides highly personalized recommendations tailored to individual user preferences .
2. Cold Start Tolerance: Can work well even with limited initial data .
3. Scalability: Can handle large datasets efficiently .

Challenges and Limitations

1. Sparsity Problem: Difficulty in finding similar users when there's limited interaction data .
2. Cold Start Problem: Struggles with recommending to new users with little interaction data .
3. Shilling Attacks: Vulnerability to malicious users manipulating the system .

Collaborative filtering remains a powerful technique for personalized recommendations, especially in domains with rich interaction data such as e-commerce sites or streaming platforms. However, addressing its limitations often requires combining it with other techniques or incorporating domain-specific knowledge.

Collaborative filtering is compared to other recommendation techniques in several ways:

1. Data Requirements:
   - Collaborative filtering relies heavily on user-item interaction data (e.g., ratings, purchases, browsing history) .
   - Content-based filtering uses product attributes and user preferences derived from interaction data .

2. Cold Start Problem:
   - Collaborative filtering struggles with recommending to new users who have little interaction data .
   - Content-based filtering can work well even with limited initial data by focusing on product attributes .

3. Scalability:
   - Collaborative filtering can handle large datasets efficiently once enough interaction data is collected .
   - Content-based filtering may struggle with very large product catalogs due to increased dimensionality .

4. Personalization:
   - Collaborative filtering provides highly personalized recommendations based on similar user behavior .
   - Content-based filtering recommends items similar to what the user likes, potentially leading to more diverse recommendations .

5. Transparency:
   - Collaborative filtering doesn't provide clear reasons for recommendations .
   - Content-based filtering allows users to understand why they're being recommended certain products .

6. Handling New Items:
   - Collaborative filtering struggles with recommending new items until enough interaction data is collected .
   - Content-based filtering can recommend new items immediately once their attributes are known .

7. Sparsity Problem:
   - Collaborative filtering faces challenges when there's limited interaction data .
   - Content-based filtering is less affected by sparsity issues but may struggle with high-dimensional product spaces .

8. Shilling Attacks:
   - Collaborative filtering is vulnerable to malicious users manipulating ratings .
   - Content-based filtering is generally immune to shilling attacks .

Hybrid approaches combine collaborative filtering with content-based filtering to address many limitations:

1. Improved Personalization: Combines the strengths of both techniques .
2. Better Cold Start Performance: Can recommend to new users based on item attributes .
3. Enhanced Diversity: Provides recommendations across different dimensions .
4. Robustness: Reduces reliance on any single technique .

Examples of hybrid systems include Netflix, which uses both user behavior and movie characteristics for recommendations .

In conclusion, while collaborative filtering excels at personalization and scalability, content-based filtering offers advantages in cold start scenarios and transparency. Hybrid approaches often provide the best balance of performance and robustness for real-world recommendation systems.

Q:

Such that 1 means most dislike and 5 means most liked. In this matrix customer rating of every movie for each user is given. Now calculate user 1 customer rating which he forgets to mention for movie 5

Solutions: Steps to solve:

Solution:

Formula used to calculate the similarity

Step 2: calculating the similarity between two user using the above formula

S