Name: Purushothaman. S

Roll No: G24AI1042

Subject: Social Network Analysis.

Subject Code: CSL7390

Exam Type: Trimester II - PGD. Data Engineering - Major Exams.

Date: 19th April 2025 Time: 2.00 PM. - 5.00 PM.

Email ID: g24ai1042@iitj.ac.in

College/university: IIT - Jodhpur.

Course & Batch: PGD. Data Engineering / July 2024.

(1) (a) Convert the incidence Matrix to adjacency Matrix:

given Matrix:

1	0	0
1	1	1
0	1	0
0	0	1

Column 1: Connects Node 1 and Node 2 → edge between 1 and 2

Column 2: Connects Node 2 and Node 3 Edge between 2 & 3.

Column 3: Connects Node 2 & Node 4 → Edge between 2 & 4.

So Adjacency Matrix:

0	1	0	0
1	0	1	1
0	1	0	0
0	1	0	0

(1)(b) Which Network Model Assumes that edges are formed between pairs of Nodes with a uniform probability, independent of Each other.

Ans: (B) Erdős – Rényi (Random Network)

Reason: This Model Connects Node pairs with uniform and independent probability.

(1)(c) In game theory, a situation where No player can improve their Outcome By Unilaterally changing their Strategy, given the strategies of other players is known as:

Ans: c Nash Equilibrium.

Reason: This is when a No Player benefits By changing Strategy unilaterally.

(1)(d) The Tendency for Individuals in Social Network to associate and bond with Similar others is defined as.

Ans: Assortative Mixing

Reason: This describes preferences for Connecting with similar others.

(1)(e) Why Might betweenness centrality be a More relavent Measure than degree centrality for Identifing Critical nodes in a Network transmitting information that Must follow specific path.

Ans: Because it Quantifies how often a Node lies on the shortest paths between other Nodes.

Reason: Betweenness centrality highlights Nodes critical to information flow.

(1) (3) A key finding about Scale-free Networks (like those generated by the Barabasi-Albert Model) is their robustness to random Node failure but vulnerabilities to targeted attacks on hubs, what underlaying property best explains this?

Ans: The presence of Many Nodes with Very high degrees (hubs)

Reason: Hubs keeps the Network connected their loss disrupts connectivity.

(4) (9) In Community detection, optimizing for high Modularity aims to find partitions, where:

Ans: The Number of Intra-Community Edges is significantly higher than expected in a random Network with the same degree sequences.

Reason: High Modularity reflects strong Community Structure.

(1) (b) Consider the two Nodes X, & Y. The Neighbors of X are A, B, C, D. & The Neighbors of y are C, D, E. What is the Jaccard Co-efficient for link prodection between x & Y?

Ans: 2/5.

Reason: Jaccard Co-efficient = $|C \cap D| / |C \cup D|$

$= \{C, D\} \rightarrow 2/\{A, B, C, D, E\} = 2/5$.

(2) (a) In the Context of Information Casecade Models, how does the Activation Mechanism differ fundamentally between the Independent Casecade Model (ICM) and the Linear Threshold Model (LTM)?

Ans: ICM uses edge probabilities; LTM uses weighted Sum (Vs) Threshold.

Reason: ICM is probabilitic per edge; LTM checks if influence exceeds Threshold.

(i) (j) A Standared graph Convolutional Network (GCN) aggregates information from a Node's immidiate Neighbours. Why Might this standared Message-passing approach be suboptional for tasks like Node classification in Networks with high hetrophily (where connected Nodes tend to be dissimilar?

Answer: Because aggregating features from dissimilar Neighbours can blur the Node's own identity representive features, Making classification harder.

Reason: Heterophilic graph lead to poor GCN Performance due to Noisy Aggrigation.

Qn(2) Solution

Problem Statement:

Vaccinate 5% of the population to minimize infections in a social contact network using Network Analysis Concept.

SIR Model:

S → Susceptible, I → Infected

R → Recovered (includes Vaccinated Individuals)

Goal: Select Critical Nodes to vaccinate (preemptively Move to R)

Strategy: Use two Network Measures.

1) Degree centrality: Nodes with the highest Number of direct connections

Why?: They can infect Many Neighbors Quickly.

(2) Betweenness Centrality

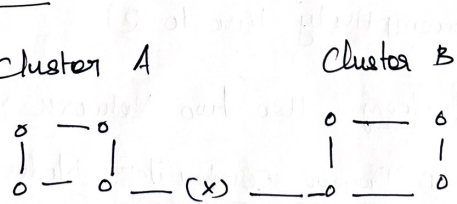
Measures how often a Node lies on the shortest paths between other Nodes

Why?: Nodes acting as bridges between clusters can spread infection across the Network.

Combined Approach:

Rank all Nodes by a Combined Score (e.g. weighted sum of degree and betweenness) & then select the top 5%

Diagram:



(x) high betweenness.

Vaccinating (x) cuts off Inter-Cluster transmissions.

(3) Suggested Collaborators: link prediction + Embedding.

Task: Use Node2Vec and link prediction to reccommandaal academic collaborators

Solution

① Node Embedding. (Node2Vec).

This converts Nodes into Vectors representations based on Network Structures.

Similar Nodes (eg. co-Authors) are closer in vector spaces.

② Link predictions

To predict the link we can use Similarity Metrices (like Cosine similarity) between embeddings.

Predict New links (collaborations) between Similar researchers.

Cosine Similarity formula:

Cosine Similarity = $(A.B)/(|A| * |B|)$.

(3) Homophily: Researchers in Similar fields tend to collaborate More (birds of a feather)

To encourage Cross-Disciplinary Collaboration:

⇒ Use Metadata (e.g) research field to diversity recommendations.

⇒ Reward link Suggestions between Structurally similar but topically diverse Nodes.

Table Example:

Researcher A	Field	Similarity Score	Cross-Discipline
Dr. X	AI	0.92	No
Dr. Y	Biology	0.85	Yes (Promote)

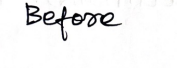
Q4: Community Detection: Girvan - Newman Vs. Louvain.

(a) Girvan - Newman Algorithm

It removes edges with highest betweenness centrality. As edges are removed, graph breaks into communities.

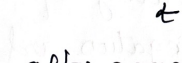
Ex:

Before



∴ B-D has edge betweenness

So after removing B-D:



⇒ Two Communities.

(b) Edge between centrality:

Number of shortest paths between Nodes that passes through the edge.

Iteratively remove highest-scoring edges.

(c) Limitations

i) Expensive computation ($O(n^3)$),

ii) Not suitable, scalable for large graphs.

(d) Louvain Method:

It optimizes Modularity (Quality of community partitions). There are Two phases:

i) Local Modularity optimization,

2) Node Aggregation and Iteration,

for Much faster and scalable.

Qn: 5) Page Rank Algorithm:

The intuition behind the page Rank algorithm:

It is Important to link the important pages by other Important Pages.

It Simulates a random web surfer.

Equation :

$$PR(i) = (1-d)/N + d * \Sigma [PR(j) / \text{out\_degree}(j)]$$

(b) Damping Factor :

Damping factor (d) (typically 0.85) represents the probability of following a link.

(1-d) represents random jumps to any Node (It prevent getting any Stuck) it helps in Maintaining Convergence and reflecting realistic ~~sort~~ Surfing Behaviour

(c) Dangling Node Problems :

The Dangling Nodes have No outgoing edges (No links to follow)

Problem their page rank can leak out of the System, affecting convergence.

Solution: Redistribute their PageRank uniformly across all Nodes in each iteration

This keeps the transition Matrix Stochastic (rows sum to 1), ensuring stable and correct result.

Page Rank Formula:

For a Node i, the page Rank. PR(i) is calculated as:

$$PR(i) = \frac{1-d}{N} + d \sum_{j \in In(i)} \frac{PR(j)}{outdegree(j)}$$

Where

d - damping factor (usually 0.85)

N -> Total Number of Nodes.

In (i) = set of Nodes linking to Node (i).

Outdegree (j) = Number of links going out from Node j.

Ex:

A → B
B → C
C → A

} Each Node likes to one other Node.

Let Assuming the Damping factor
$d = 0.85$

Initial pageRank of each Node: PR = 1/3

Iterating 1 calculated:

(i) PR(A) A has one in coming link from C

PR(C) = 1/3 Out degree(C) = 1.

$$PR(A) = \frac{1-0.85}{3} + 0.85 \cdot 1/3 = 0.05 + 0.283$$

$$= 0.333$$

$$PR(B) = 0.05 + 0.85 \times 1/3 = 0.333$$

$$PR(C) = 0.05 + 0.85 \times 1/3 = 0.333.$$

In this perfectly circular graph, the PageRanks remains equal at each step.

Dangling Node case:

Let's say Node C has No Outgoing links (a dangling Node):

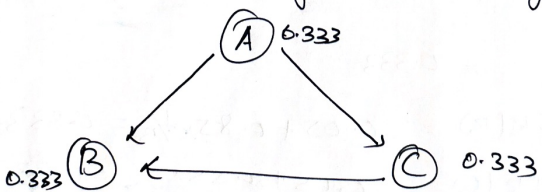
A → B
B → C
C → ϕ

} To handle dangling Node C

Its Rank Redistributed uniformly to A, B, & C.

So PR(C) = 0.3 this 0.3 is divided equily Each Node gets 0.3/3 = 0.1.

This ensures the page Rank remains Normalized & algorithm Converges.



Q.N(6)

Solution:

Game between two players are here shown in table.

Payoff Matrix:

	Strategy A	Strategy B
U (Player 1)	(3,2)	(0,1)
L (player 2)	(2,0)	(2,3)

here first value = player 1's payoff

Second value = player's payoff.

(a) Pure Strategy Nash Equilibrium

A Nash equilibrium is where NO player wants to change strategy unilaterally.

check each cell:

-> (U,A): player 2 prefers B since (3 > 2).

(U, B) player 1 prefers L (since 2 > 0)

(L, A) player 2 prefers B (since 3 > 0)

(L, B) Both players are best-responding to each other → Nash Equilibrium.

Answer: only (L, B) is pure strategy Nash equilibrium.

(b) Expected payoffs for players above.

Let player 1 play:

U with probability P.

L with probability (1-P).

Payoff of player 2:

If they choose A:

$= 2p + 0(1-p) = 2p.$

If they choose B

$1p + 3(1-p) = p + 3 - 3p = 3 - 2p.$

(c) For (P=0.7)

If P=0.7 than. A : 2*0.7 = 1.4 &

B = 3-2*0.7 = 1.6.

Here player 1 should be choose Strategy B, as it gives higher expected pay off.

Here the Table:

		Strategy A.	
Player 1		(3,2)	(0,1)
Player 1	U	(2,0)	(2,3)
	L	(2,0)	

Qn:(4) GNN Layer update for Node B.

Problem Statement: The given directed graph where Nodes A, C and D point to Node B, Here we need to calculate the updated feature

Vector $h_B^{(1)}$ using simple GNN Layer

Soln:

GNN Layer operations

⇒ Aggregate Neighbor features (Average)

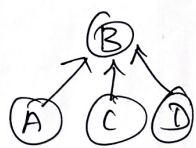
⇒ Transform using a weight Matrix W

Activate using ReLU

Given Data:

Initial features Vectors.

- $h_A^{(0)} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$
- $h_C^{(0)} = \begin{bmatrix} 0 \\ 3 \end{bmatrix}$
- $h_D^{(0)} = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$



Neighbours of A, C, D.

Weight Matrix W:

$$W = \begin{bmatrix} 0.5 & 0 \\ 0.1 & 0.2 \end{bmatrix}$$

Aggregate - Neighbor feature (Average):

$$h_N^{(0)}(B) = \frac{1}{3}\left(h_A^{(0)} + h_C^{(0)} + h_D^{(0)}\right) = \frac{1}{3}\begin{bmatrix} 1+0+2 \\ 1+3+8 \end{bmatrix}$$

$$= \frac{1}{3}\begin{bmatrix} 3 \\ 6 \end{bmatrix}$$

Linear Transformation

$$W \cdot \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 0.5\times 1 + 0.2 \\ 0.1\times 1 + 0.2\times 2 \end{bmatrix} = \begin{bmatrix} 0.5 \\ 0.1+0.4 \end{bmatrix}$$

$$= \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}$$

Apply ReLu Activation

ReLu(x) = Max(0,x) applied element wise

$$h_B^{(1)} = \text{ReLu}\left(\begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}\right) = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}$$

So the final Answer:

$$h_B^{(1)} = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}$$

This is the updated feature vector for Node B after one GNN Layer.