1 (a) Incidence to adjacency Matrix Conversion:

The incidence matrix has 4 rows (nodes, 1,2,3,4) and 3 kolumns (edges e1, e2, e3)

Edge e1 connects nodes 1 and 2.

Edge e2 connects nodes 2 and 3

Edge e3 connects nodes 2 and 4.

The corresponding adjacency matrix A for this simple undirected graph:

$$
\begin{array}{c|cccc}
 & 1 & 2 & 3 & 4 \\
1 & [0 & 1 & 0 & 0] \\
2 & [1 & 0 & 1 & 1] \\
3 & [0 & 1 & 0 & 0] \\
4 & [0 & 1 & 0 & 0]
\end{array}
$$

1 (b) Network Model

Answer (B). Erdős-Renyl (Random Network Model)

The Erdős-Renyl model (Specifically $G(n,p)$) defines a graph where each possible edge between n nodes is included independently with a uniform probability p.

1 (c). Answer (C) Nash Equillibrium

A Nash Equillibrium is a state in a game where no ~~[illegible]~~ player can benefit by changing their strategy unilaterally. assuming all other players keep their strategies unchanged.

1 (d) Answer (B) Assortive Mixing

Assortive mixing, often referred to as homophily, is the principle that needs in a network tend to connect to other nodes that are similar to themselves in some characteristic (example age, interests)

~~(2)(e)~~ 1 (e) Answer

(D) Because it quantifies how often a node lies on the shortest paths between other nodes.

Degree centrality measures direct connections while betweenness centrality measure a node's importance as an intermediary in the network. For information flowing along specific paths, nodes with high betweenness act as crucial bridges or bottlenecks, making betweenness more relevant than just the number of connections

1 (f) Answer

(C) Scale-Free Network Robustness/vulnerability.

Scale free networks are characterized by hubs. Random failures are unlikely to hit these rate hubs, preserving overall connectivity [redacted] However targeted attacks removing these hubs quickly fragment the network. (Vulnerability)

1 (g) ~~Answer~~ Answer.

(A). The number of intra-community edges is significantly higher than expected in a random network with the same degree sequence.

① (g) Justification:

Modularity measures the strength of division of a network into community. High modularity indicates dense connections within community and sparse connections between them, compared to a random baseline

① (h) answer

Neighbour $(X) = N(X) = \{A, B, C, D\}$

Neighbour$(Y) = N(Y) = \{C, D, E\}$

Intersection $|(N(X)) \cap (N(Y))| = |\{C, D\}| = 2$

Union $|N(X) \cup N(Y)| = |N(X) \cap N(Y) / N(X) \cup N(Y)| = 2/5$

Answer is (B) 2/5

① (i) Answer

(A) ICM uses edge probabilities independently; LTM uses a weighted sum of active neighbours compared to a node threshold.

Justification In ICM, an active node tries to activate its neighbours based on probabilities associated with the edges independent of other neighbours. In LTM, a node activates only if the total weight of influence from its already active neighbours surpasses its individual threshold.

~~1(f)~~ 1 (j) Answer

(B). Because aggregating features from disimilar neighbours can blur the node's own ~~representative~~ ~~features~~, making classification harder. Standard GCNs work by ~~averag~~ averaging of summing features from neighbours. In heterophilic networks, where neighbours are often disimilar. this aggregation mixes unlike features, potentially obscurring the central node's characteristics and making tasks like node classification less acurate

4 (a) Girvan-Newman algorithm

The Girvan-Newman algorithm identifies communities by iteratively removing edges with the highest edge betweeness centrality. This process progressively breaks the network down into smaller discontinued components, which represent the detected communities

4 (b) The algorithm calculates the betweeness centrality for all edges in the current network. It then removes the edge (or edges, in case of ties) with the highest betweeness score. After removal, it recalculates the betweeness centrality for all remaining edges and repeats the removal process.

4 (c) A major computational limitation is its time complexity. Calculating edge betweeness centrality for all edges is computationally expensive and the algorithm needs to repeat this calculation after ~~each~~ each edge removal, making it very slow for ~~big~~ large networks.

4 (d) Ans. Louvain method:

This is a greedy method, hierachical algorithm, that optimizes modularity much faster, than Girvan-Newman, It iteratively performs two steps,

First it locally optimizes modularity by moving nodes between communities

Second: It aggregates nodes within the same community into 'super nodes' to build a new smaller network. These steps are repeated leading to a fast convergence and making it highly scalable for large networks because it avoids the expensive global recalculations needed in Girvan-Newman algorithm.

2 Answer

I can derive a strategy based on the learnings from classes, we need to first think about the ~~state~~ strategy.

Strategy:

Calculating degree of centrality and betweenness for each individual in the social contact network. Then we need to select top 5% of individuals for vaccination by prioritizing those who rank highest in the either degree of centrality or betweenness centrality:

Degree of centrality: Target individuals with the direct contacts in the highest number. These people are likely highly infected and there are chances like these people will affect others more rapidly. So identifying and vaccinating them and putting them in isolation can stop superspreading.

2

2 continue Betweenness Centrality: It basically target people who acts like a bridge between person to person, and group to group in the community of network. So by capturing these and vaccinating them will ideally prevents silent wide speading of the influenza, which will help reducing the virus spread from cluster to cluster.

3 Answer

To improve collaborator suggestions, first we need to generate vector embedding for each researcher using node2vec trained on the paper citation and co-authorship network. These embeddings captures researchers' positions and relationships within the network, such that similar researchers (based on citation/collaboration patterns) have similar vectors.

Next we need apply link prediction algorithms to these embeddings. This involves calculating a score for potential collaborations (links) between pairs of researchers based on their respective embeddings. Common methods include computing cosine similarity between embedding vectors or using the embeddings as input features to a leaned function (example, supervised classifier trained on known past collaborators) to predict the probability of a future link, Higher the scores indicates a higher likelihood of collaboration.

3 Ans Continued:

To promote cross disciplinary collaborations, the recommendations output can be adjusted calculating initial link prediction scores, re-rank potential collaborators by introducing a diversity criterion. For instance, slightly penalizing the score of collaborators from the researchers' own field or [redacted] boost the score for those from different fields, using metadata (example: publication keywords, departmental affiliation) to determine disciplinary alignment.

In this way we can establish cross disciplinary collaborations.

5 (a) Answer:

PageRank ranks nodes based on the idea that important nodes are likely to be linked to by other important nodes. It simulates a random surfer navigating the networks. The more likely the surfer is to land on a page (node), the higher its rank. A node's importance score depends on both the number of incoming links and the importance of the nodes providing those links. Links from important nodes transfer more "rank value" than links from less important ones.

This is how page ranking works in backend, search systems like google, yahoo, youtube search works on the similar ideologies along with other metrics also.

5 (b) The damping factor (d) represents the probability that the random surfer will follow an outgoing link from the current node. The complimentory probability (1-d), is the chance that the surfer will stop following links, and instead 'teleport' to a random node anywhere in the network. This prevents the surfer from getting trapped in cycles of non-zero page rank scored edges.

5 (C) Dangling Nodes

Dangling Nodes creates a few problems like the surfer will stuck at any node and from there, he will might not be able to find another forward node. in other words he will stuck there because of having no idea about which node is linked from there. This leads page rank leaking out of the network model, as the probability mass arriving at these nodes, are not distributed further

To handle this & Convergence ensuring, the algorithm typically treats dangling nodes as if they link to every node in the network with equal probability. This means that when the random surfer reaches the dangling nodes, in the next step, they teleport to any node in the graph uniformly at random, effectively redistributing the pageRank that would otherwise be lost.

(7) Computing updated feature vector for node B. vector $h_B^{(1)}$

(i) Aggregate neighbour features:

The neighbours of node 'B' are $N(B) = \{A, C, D\}$. The number of neighbours is $|N(B)| = 3$.

The initial feature vectors are

$$h_A^{(0)} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

$$h_C^{(0)} = \begin{pmatrix} 0 \\ 3 \end{pmatrix}$$

$$h_D^{(0)} = \begin{pmatrix} 2 \\ 2 \end{pmatrix}$$

Calculating the average of these vectors.

$$h_{N(B)}^{(0)} = \frac{1}{3}\left(h_A^{(0)} + h_{(C)}^{(0)} + h_{(D)}^{(0)}\right) = \frac{1}{3}\left(\begin{pmatrix} 1 \\ 1 \end{pmatrix} + \begin{pmatrix} 0 \\ 3 \end{pmatrix} + \begin{pmatrix} 2 \\ 2 \end{pmatrix}\right)$$

$$h_{N(B)}^{(0)} = \frac{1}{3}\begin{pmatrix} 1+0+2 \\ 1+3+2 \end{pmatrix} = \frac{1}{3}\begin{pmatrix} 3 \\ 6 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

(ii) Transform

Next, we need to apply the linear transformation using the weight matrix W:

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

$$W \times h_{N(B)}^{(0)} = \begin{pmatrix} 0.5 & 0 \\ 0.1 & 0.2 \end{pmatrix}\begin{pmatrix} 1 \\ 2 \end{pmatrix} = \begin{pmatrix} 0.5\times1 + (0\times2) \\ 0.1\times1 + (0.2\times2) \end{pmatrix} = \begin{pmatrix} 0.5 \\ 0.1+0.4 \end{pmatrix} = \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix}$$

(iii) Finally, we have to apply the ReLU activation $\sigma(x) = \max(0, x)$ by each element

⑦ Continues

$$h^{(1)}_{(B)} = \sigma\left(\begin{pmatrix}0.5\\0.5\end{pmatrix}\right) = \begin{pmatrix}\max(0, 0.5)\\ \max(0, 0.5)\end{pmatrix} = \begin{pmatrix}0.5\\0.5\end{pmatrix}$$

The updated feature vector for node B is $= h^{(1)}_{(B)} = \begin{pmatrix}0.5\\0.5\end{pmatrix}$

---

⑥

~~Star Strategy~~	Strategy A	Strategy B
Strategy U	(3,2)	(0,1)
Strategy L	(2,0)	(2,3)

(6)(a) The pure strategy Nash Equilibris (PNSE) are (U,A) and (L,B)

(U,A) is a PSNE:

~~Because~~ Because If player 1 plays U, player 2 gets a higher payoff from A than B. (1)

If player 2 plays A, player 1 gets a higher payoff from U(3) than L(2)

Neighther player has an Eacentive to unilaterally deviate.

(L,B) is a PSNE:

Because If player 1 plays L, player 2 gets a higher payoff B (3) than A(0)

If player 2 plays B, player 1 gets a higher payoff from L(2) than U(0)

Neigther player has an incentive to unilaterally deviate.

6(b)

Let E2(A) be the expected payoff for player 2 playing strategy A and E2(B) be the expected payoff for player 2 playing strategy 2.

Expected Payoff for player 2 choosing strategy A:

$E2(A) = (\text{payoff of A if P1 plays U}) \times P(\text{P1 plays U}) + (\text{Payoff of A if P1 plays L}) \times P(\text{P1 plays L})$

$E2(A) = 2 \times P + 0 \times (1-P)$

$E2(A) = 2P$ — ①

Expected payoff for player 2 choosing strategy B

$E2(B) = (\text{Payoff of B if P1 plays U}) \times P(\text{P1 plays U}) + (\text{payoff of B if P1 plays L}) \times (P(\text{P1 plays L}))$

$E2(B) = 1 \times P + 3 \times (1-P)$

$= P + 3 - 3p$

$E2(B) = 3 - 2p$ — ②

The expected of player 2 is 2p, for strategy A and 3-2p for strategy B

6(c) If p = 0.7 from 6(b) we know that $E2(A) = 2p$

$E2(B) = 3 - 2p$

If we substitute p in equation ① & ②

$E2(A) = 2(0.7) = 1.4$

$E2(B) = \cancel{2 \times (0.7)}\ 3 - (2 \times 0.7)$

$= 3 - 1.4 = 1.6$