Social Network Analysis

Abhijeet Kumar
G24AI1005

1)

a) Given graph represents
4 vertices with ~~5~~ 3 edges
Edge 1 connects 1 and 2
Edge 2 connects 2 and 3
Edge 3 connects 2 and 4
The resulting adjacency matrix is

$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

b) B) Erdos Renyi (Random Network Model)

c) C) Nash Equilibrium

d) B) Assortative Mixing

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

f)

C) The presence of many nodes with very high degrees (hubs) that maintain connectivity

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

h) B) 2/5

i) A) ICM uses edge probabilities independently LTM uses a weighted sum of active neighbors compared to a node threshold.

j) B) Because aggregating features from dissimilar neighbours can blur the nodes own representative features, making classification harder.

2) In this senario, I would prefer to use Betweeness Centrality and K-core decomposition

Strategy

- By performing Betweeness Centrality Analysis, on the Social network, key individuals who frequently lie on the shortest paths between others, serving as bridges between different communities are identified. They act as critical transmission points between otherwise disconnected groups
- K-core decomposition could be used to identify where the social network graph is densest. Indiduals in higher k-core represent those who are highly interconnected with many others. These are those who are situated in network regions with high transmission potential.
- Combined Approach - Priovitize vaccination of those indiduals with wigh ranking in both measures. Allocate little more than half to high betweeness nodes to cut transmission from one group to another, and little under half (remaining) to the indiduals with high k-core vank and not already vaccinated due to high betweeness

Justification

Betweeness Centrality targets critical bridge nodes that connect distinct communities, effectively restricting the outbreak within infected groups, stopping ~~transmit~~ transmission to other groups in the network.

K core decomposition identifies densely connected cluster where the disease can spread the fastest. The highest k-cores represent super spreader environments where creating ideal conditions for explosive transmission.

Combined Approach - A combined approach of both restricts the disease for quickly ~~to~~ transmitting ~~for~~ from one group to another group of closely connected individuals. At the same time it ensures that those who are at the maximum risk, and can quicky spread the disease when infected are also vaccinated, restricting the ~~ma~~ overall number of infection, rate of increase of infection and reduces risk of infection to other individuals who can ~~spread~~ spread it to many people.

3) This is how I would combine link prediction algorithms with node embedding techniques to do the asked.

- Data Ingestion & Network Construction
  - Build a multi layer network incorporating citation Networks (who cites whom), Research topic summary (based on paper content) and Institutional Affiliations
- Node Embedding Generation
  - Apply Node2Vec to learn low dimentional vector representations of researchers that capture structural roles in the collaboration network, Proximity in the citation network and set parameters to balance between local network exploration and structural equivalence
- Link Prediction Framework
  - Use the generated embeddings as features for supervised link prediction models and train models on historical collaboration data to predict future collaborators.
- Time aware modelling
  - Incorporate temporal dynamics by weighing recent collaborations / citations more heavily.

Role of Homophily

In the embedding space, homophilly creates dense clusters of researchers within disciplines

Standard link prediction will mostly recommend within-in field collaborators that have higher probability

Promoting cross-disciplinary collaborations

I would implement a "bridging score" system that

- Uses community detection to identify disciplinary networks in the network
- Calculate the embedding space distance between different research communities
- Identifies researchers with moderate similarity across disciplinary bounderis
- Scope potential collaborators based on-
  - Complimentary expertise
  - Track record of successful boundary-spanning (Previous cross disciplinary work)
  - Balance Embedding similarity (similar enough to collaborate- different enough to bring different perspectives)
- Implement a recommendation diversity

parameter that users can adjust to control the balance between

- high probability - within field collaborators
- lower probability - cross disciplinary collaborations

4)

a) Core idea of Girvan Newman Algorithm

The core idea behind the Girvan Newman algorithm is to identify communities by progressively removing edges that are most likely to connect different communities rather than nodes within the same community. It works on the principle that inter community edges generally have higher betweeness centrality than intra community edges, as they serve as bridges between densely connected groups. By removing these high-betweeness edges, the network naturally separates into its community structure.

b) Girvan-Newman algorithm uses edge betweeness centrality iteratively through the following process

i) Calculate betweeness centrality of all edges
ii) Remove edge with highest betweeness centrality
iii) Recalculate betweeness of all remaining edges (this step is crucial as the removal of an edge changes the shortest paths)

iv) Repeat steps 2-3 until the network is fully decomposed

This iterative approach creates a hierarchical community structure that can be represented as a dendogram, with communities merging at different levels of edge removal

c) The major computational limit of this algorithm is its high time complexity. For a network with n nodes and m edges, Calculating the edge betweeness centrality has time complexity of $O(mn)$, and this calculation must be repeated after each edge removal, giving it complexity of $O(m^2n)$ for sparse networks and $O(n^3)$ for dense networks.

d) Louvain method provides a more scalable alternative for optimizing modularity through

- Local Optimization - It starts by assigning each node to its own community and then iteratively moves nodes to neighboring communities if the move increases modularity, focussing on local changes rather than global edge properties

- Hierarchical Aggregation - After the first phase, it creates a new network where communities become nodes, and the process repeats at this higher level, creating a hierarchical structure.
- Greedy Approach - By using a greedy optimization approach, it achieves significant computation efficiency with a typical time complexity of $O(n \log n)$ making it more suitable for very large networks.
- Non deterministic Results - While faster, the algorithm may produce slightly different results in different runs due to its heuristic nature, though these variations usually have similar modularity scores

5)

a) The core intution behind the PageRank algorithm is that a node's importance in a network is determined by both the quantity and quality of the links pointing to it where
- A node recieves importance from other nodes that link to it
- The importance passed from a node is devided equally among all its out going links
- More important nodes pass more importance to other neighbours.

This creates a recursive definition where a node's PageRank depends on the PageRank of nodes

pointing to it, modelling the idea that important pages are linked to by other important pages. The algorithm simulates a "random surfer" who follows links randomly and occassionally jumps to random pages, with the PageRank value representing the probability that this surfer will land on the given page

b) The damping factor d, serves several crucial role in the PageRank algorithm
- It represents the probabily that a user will continue following the links rather than jumping to a new page
- It ensures that the iterative calculation will converge to a stable solution by making the system of equations contractive
- It efficiently makes the graph fully connected by allowing jumps between any nodes, addressing issues with disconnected components
- Small values of d makes the algorithm less sensitive to the link structure and more evenly distribute rank across all nodes.

c) Dangling Nodes create issues in PageRank calculation because of following
- They act as "rank sinks" that accumulate PageRank but dont distribute it back to the network
- They violate the stochastic ~~natu~~ nature of the transition matrix needed for convergence
- Without handling, they can cause Rank to leak out of the system in each iteration.

Solution
- For dangling nodes, the algorithm redistributes their Page Rank uniformly across all nodes in the network
- Modifying the graph structure to add links from dangling nodes to all other nodes

6)

a) Pure strategy Nash Equilibrium is a pair of strategies (one for each player) where neither player can unilaterally deviate to a different strategy and improve their payoff.
(U,A): Player 1 prefers 3 over 2 (if Player 2 plays A) and Player 2 prefers 2 over 1 (if Player 1 plays U)

b) Expected payoff for Player 2
Strategy A - $E_2(A) = p \times 2 + (1-p) \times 0 = 2p$
Strategy B - $E_2(B) = p \times 2 + (1-p) \times 3 = 3 - 2p$

c) If $p = 0.7$

$E_2(A) = 2 \times 0.7 = 1.4$

$E_2(B) = 3 - 2 \times 0.7 = 1.6$

7) Aggregate neighbor features

$$\text{Aggregate} = \frac{1}{|N(B)|} \sum_{u \in N(B)} h_u^0$$

$$= \frac{1}{3}\left(h_A^{(0)} + h_C^{(0)} + h_D^{(0)}\right)$$

Since number of neighbors of B = 3 so $N(B) = 3$

$$= \frac{1}{3}\begin{pmatrix}1\\1\end{pmatrix} + \begin{pmatrix}0\\3\end{pmatrix} + \begin{pmatrix}2\\2\end{pmatrix}$$

$$= \frac{1}{3}\begin{pmatrix}3\\6\end{pmatrix} = \begin{pmatrix}1\\2\end{pmatrix}$$

Now applying linear transformation using the weight matrix W to the aggregated neighbor vector

Transform = W . Aggregate.

$$= \begin{pmatrix}0.5 & 0\\0.1 & 0.2\end{pmatrix}\begin{pmatrix}1\\2\end{pmatrix}$$

$$\therefore \text{Transform} = \begin{pmatrix}0.5\times1 + 0\times2\\0.1\times1 + 0.2\times2\end{pmatrix}$$

$$= \begin{pmatrix}.5+0\\.1+.4\end{pmatrix}$$

$$= \begin{pmatrix}.5\\.5\end{pmatrix}$$

Hence transformed vector is $\begin{pmatrix}.5\\.5\end{pmatrix}$

Applying ReLU

$$h_B^{(1)} = \sigma(\text{Transform})$$

$$= \sigma\begin{pmatrix} .5 \\ .5 \end{pmatrix}$$

$$= \begin{matrix} \max 0, .5 \\ \max 0, .5 \end{matrix} = \begin{matrix} .5 \\ .5 \end{matrix}$$

Hence updated feature vector for Node B $h_B^{(1)} = \begin{matrix} .5 \\ .5 \end{matrix}$