Matched Answers

Matched Answers

Q1. Short answer/objective question.

(a) Consider the following incidence matrix of a simple undirected graph. Convert this into an adjacency matrix representation.

Student's Answer:

```
Given Matrix:
| 0 1 0 |
| 1 0 1 |
| 0 1 0 |
| 0 0 1 |

here each row represents a node (vertex) : A, B, C, D
each column represents an edge: (e1, e2, e3)

e1 connects nodes A, B
e2 connects nodes B, C
e3 connects nodes B, D

Adj matrix (nxn) is
  A B C D
A 0 1 0 0
B 1 0 1 1
C 0 1 0 0
D 0 1 0 0

from incidence matrix we place 1's for each pair
symmetrically in the adj matrix to reflect the undirected
connections.
```

(b) Which network model assumes that edges are formed between pairs of nodes with a uniform probability, independent of other edges?

Student's Answer:

Erdős{Rényi (Random network model)
* This model assumes that edges are formed
between pairs of nodes with uniform probability and
each edge is added independently of others.
* Other models involve preferential attachment,
local rewiring or fixed degree sequences.

(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:

Student's Answer:

Nash Equilibrium
* It represents a stable state where all players
are making the best decision they can, given the decisions
of others.

(d) The tendency for individuals in a social network to associate and bond with similar others is defined as:

Student's Answer:

Assortative Mixing
* Refers to the tendency of individuals in a social
network to associate and bond with others who have
similarities to themselves like age, professions, beliefs,
hobbies, etc.

(e) Why might betweenness centrality be a more relevant measure than degree centrality for identifying critical nodes in a network transmitting information that must follow specific paths?

Student's Answer:

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

```
* Betweenness centrality measures the extent to which
a node lies on the shortest paths between other nodes which
makes it highly relevant in networks where information
flows alongside specific paths as nodes with high betweenness
can control communication between different parts of the
network.
```

Q2. A novel influenza strain (following an SIR - Susceptible, Infected, Recovered - model) is spreading in a city. You have access to a network graph representing close social contacts (nodes=people, edges=contacts). Resources are limited, allowing you to preemptively vaccinate (move directly to the 'Recovered' state) only 5% of the population. Describe a strategy using at least two distinct network analysis concepts that can be applied together to identify the individuals to vaccinate to most effectively minimize the total number of infections. Justify why your chosen concepts are appropriate and how they would be applied together.

Student's Answer:

```
Strategy to vaccinate 5% of the population:

To minimize the spread of infection in an SIR
model using only 5% vaccination coverage we can
combine 1) Modularity optimization
         2) Betweenness centrality
to select most impactful individuals for vaccination.

Betweenness Centrality:
* Measures how often a node lies on the
shortest paths between other nodes in the network.
* A node with high betweenness centrality has
more control over the flow of information/spread of
disease as it acts as a bridge between different
parts of the network.
* By vaccinating individuals with high
betweenness centrality we can block the flow of infection
between various groups or communities within the
network.

Why choose Betweenness Centrality:
```

* Vaccinating high-betweenness individuals can
interrupt the transmission path most effectively.
* These individuals are likely to connect clusters of
individuals who would otherwise have limited contact
so vaccinating them can prevent the disease from
spreading across larger population of the network.

Modularity optimization:
* Involves identifying clusters of nodes that are
more densely connected with each other than with the
rest of the network.
* By identifying communities, we can target
the vaccination of individuals in critical communities that
may have higher likelihood of experiencing rapid outbreaks
due to denser connectivity.

Why choose Modularity optimization:
* In networks with community structure,
vaccinating individuals in densely connected communities can
be especially effective in limiting outbreaks.
* If a community is more tightly connected an
infection can spread quickly within it so preemptively
vaccinating key individuals within those communities
can prevent large scale spread.

Combining the two concepts:
i) Identify the communities within the network
using a Modularity optimization algorithm.
ii) Within each community calculate betweenness
centrality to identify individuals who serve
as key bridges between communities.
iii) Prioritize vaccinating the individuals who have
high betweenness centrality and are situated
in communities with high interconnectivity which
ensures that resources are allocated effectively
to block transmission both within & between
communities.

Justification:
This combined approach addresses both the
inter-community transmission and intra-community

```
transmission ensuring the vaccination strategy is
both effective & efficient given the limited resources.
```

Q3. You are tasked with improving the 'suggested collaborators' feature on a platform for academic researchers. Explain how you could combine link prediction algorithms with node embedding techniques (e.g., Node2Vec trained on paper citation/co-authorship networks) to generate recommendations.

Discuss the role of homophily (researchers collaborating within similar fields) in this context and suggest one potential way to promote cross-disciplinary collaborations using your proposed system.

Student's Answer:

```
To enhance the 'suggested collaborators' feature
we can combine link prediction algorithms with
node embedding techniques trained on academic
networks.

i) Node Embedding with "Node2Vec":
* Node2Vec learns low-dimensional representation
of nodes by simulating biased random walks on the graph.
* These embeddings capture both structural roles
and community affiliations.
* Train Node2Vec on co-authorship networks
and citation networks.

ii) Link Prediction:
* Use the learned embeddings to predict the
likelihood of future collaboration:
-> Combine embeddings of two nodes via dot
product or neural nets.
-> Rank potential collaborators by predicted
scores.
* This allows suggesting pairs of researchers likely
to collaborate in the future based on both network
proximity & embedding similarities.

-> Role of Homophily:
* Homophily is the tendency of nodes
```

to connect with similar others.
-> In research this means scholars often
collaborate within the same field or institution.
* Node2Vec naturally captures homophily as
nodes with similar neighbours are placed close in
embedding space.
* This improves recommendations accuracy,
ensuring relevance of suggestions.

-> Promoting Cross-Disciplinary collaboration:
* Introduce a controlled diversity factor
-> Penalize overly similar embeddings
-> Prioritize researchers with complementary
expertise by using content features.

Eg:
* Use metadata to find non-overlapping yet
related fields.
* Boost recommendations where embeddings are
moderately similar but come from different
disciplines.

Q4. Describe the core idea behind the Girvan-Newman algorithm for community detection.

Student's Answer:

Girvan-Newman Algorithm for Community Detection:
* Identifies communities by iteratively removing
edges that are central to the network.
* The idea is that communities within a network
are separated by edges with high betweenness centrality.
* By systematically removing edges with high
betweenness centrality the network progressively splits
into smaller communities.
* This process continues until the network is
divided into distinct communities, with the assumption
that these communities are internally more connected and
externally sparse.

Q5. Explain how it uses edge betweenness centrality iteratively.

Student's Answer:

```
* Calculates the betweenness centrality of all edges.
* Removes the edge with the highest betweenness.
* Recalculates the betweenness centrality of
the remaining edges after each removal.
* Repeats this process until the network splits
into distinct communities.
```

Q6. What is a major computational limitation of this algorithm?

Student's Answer:

```
A major computational limitation of this
algorithm is its high time complexity.
* It requires to recalculate edge betweenness
centrality after each edge removal.
O(n^3)   n -> nodes
         m -> edges
```

Q7. Briefly explain how the Louvain method provides a more scalable alternative for optimizing modularity.

Student's Answer:

```
* The Louvain method is a more scalable approach
for community detection specifically designed to optimize
modularity.

Two phases:
i) Local Modularity Optimization:
* Each node is assigned to its own community
* For each node the algorithm evaluates whether
moving the node to a neighbouring community would
increase the overall modularity of the network and
move the node if it does.

ii) Community Aggregation:
```

```
* The communities formed in the 1st phase are
aggregated into super nodes, and the algorithm
repeats the modularity optimization process on this
new group of communities.
* This process is iterated, with each iteration
resulting in a finer level of community structure.

Advantages:
* Provides a direct measure of community quality.
* Highly efficient in scaling to large networks
as the size of network is reduced in each step.
* Low time complexity.
```

Q8. Explain the intuition behind the PageRank algorithm for determining node importance.

Student's Answer:

```
Page rank algorithm:
* Based on the idea that the importance
of a node is determined by the importance of the
nodes that link to it.
[in this case nodes are webpages]
* The intuition is that a webpage is more
important if it is linked by other important pages.
* Page rank treats the network as a directed
graph where nodes represent webpages and edges
represent hyperlinks between them.
-> The algorithm assigns a score to each node
based on two main principles,
* Incoming links
* Link quality
* Page rank is computed iteratively with each
node's rank being based on the ranks of the nodes
linking to it.
```

Q9. Describe the role of the 'damping factor' (d) for random surfer based PageRank algorithm.

Student's Answer:

```
Role of damping factor (d) in the Random Surfer
model:
* Damping factor (d) in page rank algorithm models
the behaviour of a random surfer navigating the web.
* The damping factor serves to:
i) Control Random Behaviour:
The parameter 'd' represents the
probability that the random surfer will continue
following links (usually 0.85)
the term (1-d) represents the probability
that the surfer will jump to a random page, ensuring that
every node in the graph has a non-zero probability of
being reached even if it is isolated or has no links
incoming.

ii) Ensure Convergence: without the teleportation
step the algorithm could get stuck in pages with
no outgoing links or sink into cycles.
The damping factor prevents this, by allowing
surfers to jump to any page in the graph.
```

Q10. What problem arises from 'dangling nodes' (nodes with no outgoing links), and how is this typically handled in the PageRank calculation to ensure convergence?

Student's Answer:

```
Dangling nodes:
* Nodes in the graph that have no outgoing
links.
* In the context of page ranking algorithm
this presents a problem as, if a node has no
outgoing links, it can't contribute to the rank of
any other node.
* This results in a loss of rank distribution
causing the algorithm to not converge properly or
assign incorrect rank values.

Handling Dangling nodes in Page Rank:
i) Redistribute Rank: Instead of allowing the
```

```
rank to leak out of dangling nodes, their rank
is redistributed across all other nodes.

ii) Teleportation:
The damping factor already includes a
mechanism that reduces the impact of dangling
nodes by giving a rank to all nodes via teleportation.
```

Q11. Two players on a network edge play a game with the following payoff matrix. Row player is Player 1 and Column player is Player 2.

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

(a) Identify all Pure Strategy Nash Equilibria in this game. Briefly explain why they are equilibria.

Student's Answer:

```
Pure Strategy Nash Equilibria:

Payoff matrix
Strategy A   Strategy B
(3,2)        (0,1)
(2,0)        (2,3)

Player 1 (row):
* if player 2 plays A
U: 3  L: 2  -> best response = U
* if player 2 plays B
U: 0  L: 2  -> best response = L

Player 2 (column):
* if player 1 plays U
A: 2  B: 1  -> best response = A
* if player 1 plays L
A: 0  B: 3  -> best response = B

So, the Nash Equilibria are where both players
playing best responses to each other
```

```
(U, A) -> player 1 plays U, player 2 plays A
(L, B) -> player 1 plays L, player 2 plays B

Neither players can improve their payoff
by deviating unilaterally.
```

(b) Suppose Player 1 chooses 'Strategy U' with probability p and 'Strategy L' with probability 1 - p. Calculate the expected payoff for Player 2 for each of the strategy.

Student's Answer:

```
Let player 1 play
* U with probability p
* L with probability 1-p

Expected payoff of player 2:

If player 2 plays A
payoff = p.2 + (1-p).0 = 2p

If player 2 plays B
payoff = p.1 + (1-p).3
       = p + 3 - 3p
       = 3 - 2p
```

(c) What will be the expected outcome if p = 0.7?

Student's Answer:

```
If p = 0.7

Player 2's payoff if playing A
2.0.7 = 1.4

Player 2's payoff if playing B
3 - 2.0.7 = 3 - 1.4 = 1.6

 1.6 > 1.4 player 2 will choose strategy B.

Expected payoff:
player 1's:
```

```
B = 0.7.0 + 0.3.2 = 0.6
player 2's: 1.6

=> (0.6, 1.6)
```

Q12. Consider the following simple directed graph where edges point towards node B: A $\to$ B, C $\to$ B, D $\to$ B. We want to compute the updated feature vector for node B, denoted as $\mathbf{h}_B^{(1)}$, using one layer of a simple Graph Neural Network.

Student's Answer:

```
Given
N(B) = A, C   [neighbours of node B]
feature vectors
h_A^(0) = [1,1] ; h_C^(0) = [0,3] ; h_D^(0) = [2,2]
weight matrix:
W = | 0.5 0 |
    | 0.1 0.2 |

Aggregate neighbours features:
h_N(B)^(0) = 1/3 (h_A^(0) + h_C^(0) + h_D^(0))
           = 1/3 ([1,1] + [0,3] + [2,2])
           = 1/3 [3 6] = [1 2]

Transform:
W.h_NB^(0) = | 0.5 0 | [1]
             | 0.1 0.2| [2]
           = | 0.5 + 0.2 |
             | 0.1 + (0.2)2 |
           = | 0.5 |
             | 0.5 |

Activate Relu
h_B^(1) =  (| 0.5 |) = | 0.5 |
          | 0.5 |      | 0.5 |

h_B^(1) = | 0.5 |
          | 0.5 |
```