Student Answer Sheet Analysis

Automated Processing

July 19, 2025

Questions and Answers

For each question below, show the question text followed by the student's answer.

Question 1(a)

Question: Consider the following incidence matrix of a simple undirected graph. Convert this into an adjacency matrix representation. [2]

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

Answer: Given the incidence Matrix:

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

Consider rows as nodes - 4 nodes. Consider columns as edges - 3 edges. Find adjacency matrix by following how mapping Edge 1 connects node A B Edge 2 connects node B C Edge 3 connects node B D

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

Question 1(b)

Question: Which network model assumes that edges are formed between pairs of nodes with a uniform probability, independent of other edges? [2] A. Barabási-Albert Model B. Erdős–Rényi (Random Network) Model C. Watts-Strogatz (Small-World) Model D. Configuration Model

Answer: B. Erdős–Rényi (Random Network Model) It creates edges between pairs of nodes is formed independently with equal probability

Question 1(c)

Question: 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: [2] A. Zero-Sum Game B. Dominant Strategy C. Nash Equilibrium D. Mixed Strategy

Answer: C. Nash Equilibrium It states where no player can benefit by unilaterally changing their strategy, assuming all other players are holding their strategies constant.

Question 1(d)

Question: The tendency for individuals in a social network to associate and bond with similar others is defined as: [2] A. Structural Equivalence B. Assortative Mixing C. Regular Equivalence D. Network Density

Answer: B. Assortative Mixing The tendency of nodes in a network to connect with other nodes that have similar characteristics.

Question 1(e)

Question: 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? [2] A. Because it measures the total number of connections a node has. B. Because it prioritizes nodes with high clustering coefficients. C. Because it is easier to calculate for large graphs. D. Because it quantifies how often a node lies on the shortest paths between other nodes.

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

Question 1(f)

Question: A key finding about scale-free networks (like those generated by the Barabási-Albert model) is their robustness to random node failures but vulnerability to targeted attacks on hubs. What underlying property best explains this? [2] A. Low average path length. B. A uniform degree distribution. C. The presence of many nodes with very high degrees (hubs) that maintain connectivity. D. A high clustering coefficient across all nodes.

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

Question 1(g)

Question: In community detection, optimizing for high modularity aims to find partitions where: [2] A. The number of intra-community edges is significantly higher than expected in a random network with the same degree sequence. B. The number of inter-community edges is maximized. C. The number of communities is maximized. D. The size of all communities is perfectly balanced.

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

Question 1(h)

Question: Consider two nodes, X and Y. The neighbors of X are A, B, C, D. The neighbors of Y are C, D, E. What is the Jaccard Coefficient for link prediction between X and Y? [2] A. $\frac{2}{7}$ B. $\frac{2}{5}$ C. $\frac{2}{4}$ D. $\frac{4}{3}$

Answer: X: A, B, C, D Y: C, D, E XUY = 5 X Intersection Y = 2 Jaccard Coefficient = 2/5

B: 2/5

Question 1(i)

Question: In the context of information cascade models, how does the activation mechanism differ fundamentally between the Independent Cascade Model (ICM) and the Linear Threshold Model (LTM)? [2] A. ICM uses edge probabilities independently; LTM uses a weighted sum of active neighbors compared to a node threshold. B. LTM uses edge probabilities independently; ICM uses a weighted sum of active neighbors compared to a node threshold. C. Both models rely solely on the number of active neighbors, ignoring edge weights or probabilities. D. ICM activates nodes based on global network properties, while LTM uses only local information.

Answer: A. ICM uses edge probabilities independently; LTM uses a weighted sum of active neighbors compared to a node threshold.

Question 1(j)

Question: A standard Graph Convolutional Network (GCN) aggregates information from a node's immediate neighbors. Why might this standard message-passing approach be suboptimal for tasks like node classification in networks with high heterophily (where connected nodes tend to be dissimilar)? [2] A. Because GCNs can only be applied to undirected graphs. B. Because aggregating features from dissimilar neighbors can blur the node's own representative features, making classification harder. C. Because GCNs require nodes to have features, which is not always possible. D. Because GCNs are computationally too expensive for heterophilic graphs.

Answer: B. Because aggregating features from dissimilar neighbors can blur the node's own representative features, making classification harder.

Question 2

Question: 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

Answer:

Vaccination strategy using network analysis:

a. Betweenness centrality - It helps to identify nodes that act as bridges between different communities. These nodes often lie on the shortest paths between many pairs of nodes and control the flow of the infection between different parts of the network.

b. Degree centrality - It helps to identify high degree nodes that have many connections and higher potential to spread the virus.

Question 4(a)

Question: Describe the core idea behind the Girvan-Newman algorithm for community detection. [3]

Answer: The core idea behind Girvan Newman algorithm is to identify communities in a network by iteratively removing edges with high betweenness centrality until the network breaks down into smaller, well defined groups.

Question 4(b)

Question: Explain how it uses edge betweenness centrality iteratively. [2]

Answer: Betweenness centrality measures how often a node lies on the shortest path between other nodes in a network.

It helps to calculate betweenness centrality for all edges in the network. Removing the edge with highest betweenness centrality

Question 4(c)

Question: What is a major computational limitation of this algorithm? [2]

Answer: Major computational limitation of Girvan-Newman algorithm is its high computational complexity. For a network n nodes m edges will calculate with O(m$^2n)time.Thiscalculationmustbe$

Overall, this makes the algorithm impractical for large networks with thousands/millions of nodes edges.

Question 4(d)

Question: Briefly explain how the Louvain method provides a more scalable alternative for optimizing modularity. [3]

Answer: Louvain method provides a more scalable alternative for optimizing modularity through.

Local optimization - It starts by assigning each node to its own community. Then iteratively moves individual nodes to communities that result in the largest increase in modularity.

Hierarchical aggregation - It creates a new network where nodes are the communities found in the previous step.

Iteration on reduced network - It repeats the process on the smaller network allowing it to detect Multilevel community structure.

Question 5(a)

Question: Explain the intuition behind the PageRank algorithm for determining node importance. [3]

Answer: Intuition behind PageRank is based on the idea that the popularity of a webpage is determined not only by the number of incoming links but also by the kind of incoming links. Situations from highly ranked pages contribute more than lower ranked webpages.

Question 5(b)

Question: Describe the role of the 'damping factor' (d) for random surfer based PageRank algorithm. [3]

Answer: Damping factor(d) represents the probability that the random surfer will follow an outgoing link rather than randomly teleporting to another page.

Question 5(c)

Question: What problem arises from 'dangling nodes' (nodes with no outgoing links), and how is this typically handled in the PageRank calculation to ensure convergence? [4]

Answer: Problems with dangling nodes: Dangling nodes create a leak in pagerank calculation. It causes all pagerank values to approach zero.

Question 6(a)

Question: Identify all Pure Strategy Nash Equilibria in this game. Briefly explain why they are equilibria. [3]

Answer: Given payoff matrix

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

pure strategy (U, A) - player 1 gets 3, player 2 gets 2 (U, B) - player 1 gets 0, player 2 gets 1 (L, A) - player 1 gets 2, player 2 gets 0 (L, B) - player 1 gets 2, player 2 gets 3

Question 6(b)

Question: 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. [4]

Answer: For expected payoffs for A

E[A] = p x 2 + (1-p) x 0 = 2p Expected payoffs for B E[B] = p x 1 + (1-p) x 3 = p+3-3p = 3-2p

Question 6(c)

Question: What will be the expected outcome if $p = 0.7$? [2]

Answer: Expected outcome of p = 0.7 E[A] = 2p = 2 x 0.7 = 1.4 E[B] = 3-2p = 3 - 2 x 0.7 = 1.6

Question 7

Question: Consider the following simple directed graph where edges point towards node B: A $\rightarrow$ B, C $\rightarrow$ B, D $\rightarrow$ 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. [10]

The initial (layer 0) feature vectors for the nodes are $\mathbf{h}_A^{(0)}$, $\mathbf{h}_B^{(0)}$, $\mathbf{h}_C^{(0)}$, and $\mathbf{h}_D^{(0)}$. (Note: Specific vectors are provided at the end of the question).

The GNN layer performs the following steps to update node B's features:

1. Aggregate neighbor features: Calculate the average of the initial feature vectors of B's neighbors (A, C, D). Let this aggregated vector be $\mathbf{h}_{\mathcal{N}(B)}^{(0)}$, where $\mathcal{N}(B) = \{A, C, D\}$.

2. Transform: Apply a linear transformation using the weight matrix W to the aggregated neighbor vector.

3. Activate: Apply the ReLU (Rectified Linear Unit) activation function, where $\text{ReLU}(x) = \max(0, x)$ element-wise.

The formula for this update is:

$$\mathbf{h}_B^{(1)} = \sigma\left(W \cdot \left(\frac{1}{|\mathcal{N}(B)|} \sum_{u \in \mathcal{N}(B)} \mathbf{h}_u^{(0)}\right)\right)$$

Where σ is the ReLU activation and the weight matrix is given by:

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

Calculate the updated feature vector $\mathbf{h}_B^{(1)}$ for node B. Show your steps for aggregation, transformation, and activation clearly.

Consider initial feature vectors are $\mathbf{h}_A^{(0)} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$, $\mathbf{h}_C^{(0)} = \begin{pmatrix} 0 \\ 3 \end{pmatrix}$, $\mathbf{h}_D^{(0)} = \begin{pmatrix} 2 \\ 2 \end{pmatrix}$.

Answer: Neighbours of B = N(B) = A, C, D

Given feature vectors

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

Find average of initial feature vectors $h_{N(B)}^{(0)} = \frac{1}{3}(h_A^{(0)} + h_C^{(0)} + h_D^{(0)}) = \frac{1}{3}(\begin{bmatrix} 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 0 \\ 3 \end{bmatrix} + \begin{bmatrix} 2 \\ 2 \end{bmatrix})$ $= \frac{1}{3}\begin{bmatrix} 3 \\ 6 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$

Given weight Matrix $W = \begin{bmatrix} 0.5 & 0 \\ 0.1 & 0.2 \end{bmatrix}$ $W \cdot h_{N(B)}^{(0)} = \begin{bmatrix} 0.5 & 0 \\ 0.1 & 0.2 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 0.5 * 1 + 0 * 2 \\ 0.1 * 1 + 0.2 * 2 \end{bmatrix}$ $= \begin{bmatrix} 0.5 + 0 \\ 0.1 + 0.4 \end{bmatrix} = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}$

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