CS 579 HW3 Solution

Question 1:

(a)

In order to choose a network model that is best suited for a given real-world social media dataset to allocate resource for community building, it is best to first understand the data present.

Exploratory Data Analysis:
To understand the data, we will analyze it and focus on extracting meaningful insights related to network models such as the number of nodes, average path length, degree distribution, clustering coefficient, network density, and more. These metrics gives us an idea of how the network may look like and help us in deciding a suitable network model.

Model Selection:

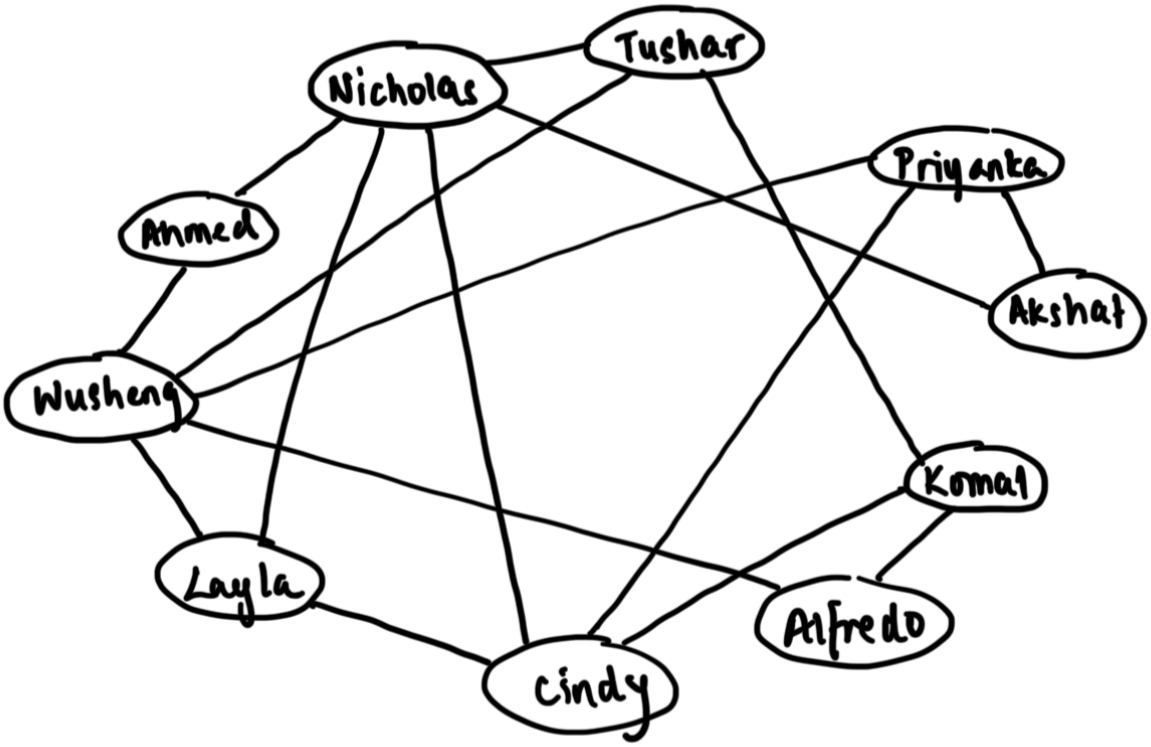
- Random Graph Model: The assumption behind the random graph model is that connections in real-world networks are formed at random. Although unrealistic, random graphs can model average path lengths in real-world networks properly but underestimate the clustering coefficient. In reality, there are no random connections.
- Small-World Model: In real-world interactions, many individuals have a limited and often at least, a fixed number of connections. This model thrives in networks with high clustering coefficients and short average path lengths, often observed in social media with strong local communities.
- Preferential Attachment Model: This model is appropriate for networks where new nodes tend to connect to already highly connected nodes. In other words, a rich-get richer phenomenon is observed here. This model also exhibits realistic average path lengths that are smaller than the average path lengths in random graphs, but it generates a small clustering coefficient, which contradicts high clustering coefficients observed in real-world networks.

Based on the understanding of these models, and the fact that we are using the data to allocate resources for community building, it may be that the Small-World Model will be best suited to be chosen as the model to simulate the original network. However, the model should be selected by comparing the metrics results of the simulated network with the metrics extracted from the actual data set given. Choose the model that closely aligns with the data.

(b)

After selecting the appropriate model, we can use it allocate resources for community building. It can be used to simulate network evolution and the diffusion of information within it. With various ways of allocating resources, we can simulate how information spreads in each scenario to find the best outcome. The model can be used to identify the densely connected clusters that can be representing communities and their characteristics. Influential nodes can also be determined as they are strongly connected with many nodes. Resources can be strategically allocated based on the information we gather.

Question 2:



(a) k-cliques

A k-clique is a maximal subgraph where the shortest path between any two nodes is always less than or equal to k.

1-cliques: {Nicholas, Cindy, Layla}

2-cliques:

{Nicholas, Tushar, Priyanka, Komal, Cindy, Layla, Wusheng}
{Tushar, Priyanka, Komal, Alfredo, Cindy, Layla, Wusheng}
{Nicholas, Tushar, Priyanka, Akshat, Cindy, Layla, Wusheng, Ahmed}

3-cliques:

{Nicholas, Tushar, Priyanka, Akshat, Cindy, Layla, Wusheng, Ahmed, Alfredo, Komal}

(b) k-plexes

For a set of vertices V, the structure is called a k-plex if we have $d_v \geq |V| - k, \forall v \in V$, where d_v is the degree of v in the induced subgraph (i.e., the number of nodes from the set V that are connected to v).

A clique of size k is a 1-plex. We want any k-plex to start with a maximal clique.

The maximal clique for the friendship graph is {Nicholas, Cindy, Layla}

For example, let us take the clique {Layla, Cindy, Nicholas} and add the node {Tushar}.
|V| = 4

For k = 1, d >= 4 – 1 = 3
For k = 2, d >= 4 – 2 = 2
For k = 3, d >= 4 – 3 = 1

For the above subgraph, the degree of each node Nicholas is 3, for Layla and Cindy it is 2, and for Tushar it is 1. This subgraph is a 3-plex.

Another example, let us take the subgraph {Layla, Cindy, Nicholas, Tushar} and add the node {Wusheng}.
|V| = 5

For the above subgraph, the degree for nodes Nicholas, Tushar, Cindy, Layla, and Wusheng are 3, 2, 2, 2, and 2 respectively. If each node should have ties with 2 nodes, then k = |V| - dv = 5 – 2 = 3. This subgraph is a 3-plex.

k = 1:

{Nicholas, Layla, Cindy}

k = 3
{Nicholas, Layla, Cindy, Ahmed}
{Nicholas, Layla, Cindy, Wusheng}
{Nicholas, Layla, Cindy, Tushar}
{Nicholas, Layla, Cindy, Priyanka}

{Nicholas, Layla, Cindy, Akshat}
{Nicholas, Layla, Cindy, Komal}
{Nicholas, Layla, Cindy, Ahmed, Wusheng}
{Nicholas, Layla, Cindy, Wusheng, Tushar}
{Nicholas, Layla, Cindy, Wusheng, Priyanka}
{Nicholas, Layla, Cindy, Tushar, Komal}
{Nicholas, Layla, Cindy, Komal, Alfredo}

k = 4

{Nicholas, Layla, Cindy, Wusheng, Alfredo}
{Nicholas, Layla, Cindy, Komal, Alfredo}
{Nicholas, Layla, Cindy, Ahmed, Wusheng, Tushar}
{Nicholas, Layla, Cindy, Ahmed, Wusheng, Priyanka}
{Nicholas, Layla, Cindy, Wusheng, Priyanka, Tushar}
{Nicholas, Layla, Cindy, Wusheng, Priyanka, Akshat}
{Nicholas, Layla, Cindy, Wusheng, Alfredo, Komal}
{Nicholas, Layla, Cindy, Tushar, Komal, Wusheng}

k = 5

{Nicholas, Layla, Cindy, Ahmed, Wusheng, Alfredo}
{Nicholas, Layla, Cindy, Wusheng, Priyanka, Alfredo}
{Nicholas, Layla, Cindy, Tushar, Komal, Alfredo}
{Nicholas, Layla, Cindy, Ahmed, Wusheng, Tushar, Komal}
{Nicholas, Layla, Cindy, Ahmed, Wusheng, Tushar, Priyanka}
{Nicholas, Layla, Cindy, Ahmed, Wusheng, Priyanka, Akshat}
{Nicholas, Layla, Cindy, Ahmed, Wusheng, Priyanka, Komal}
{Nicholas, Layla, Cindy, Ahmed, Wusheng, Alfredo, Komal}
{Nicholas, Layla, Cindy, Wusheng, Priyanka, Tushar, Komal}
{Nicholas, Layla, Cindy, Wusheng, Priyanka, Tushar, Akshat}
{Nicholas, Layla, Cindy, Wusheng, Priyanka, Alfredo, Akshat}
{Nicholas, Layla, Cindy, Wusheng, Priyanka, Alfredo, Komal}
{Nicholas, Layla, Cindy, Wusheng, Komal, Alfredo, Tushar}

k = 6

{Nicholas, Layla, Cindy, Ahmed, Wusheng, Tushar, Alfredo}
{Nicholas, Layla, Cindy, Ahmed, Wusheng, Priyanka, Alfredo}
{Nicholas, Layla, Cindy, Tushar, Alfredo, Komal, Priyanka}
{Nicholas, Layla, Cindy, Tushar, Akshat, Komal, Priyanka}
{Nicholas, Layla, Cindy, Ahmed, Wusheng, Tushar, Komal, Priyanka}
{Nicholas, Layla, Cindy, Ahmed, Wusheng, Tushar, Komal, Alfredo}
{Nicholas, Layla, Cindy, Ahmed, Wusheng, Tushar, Akshat, Priyanka}

k = 7

{Nicholas, Layla, Cindy, Ahmed, Wusheng, Tushar, Akshat, Komal}
{Nicholas, Layla, Cindy, Ahmed, Wusheng, Tushar, Alfredo, Priyanka}
{Nicholas, Layla, Cindy, Ahmed, Wusheng, Tushar, Akshat, Alfredo}
{Nicholas, Layla, Cindy, Ahmed, Wusheng, Tushar, Akshat, Priyanka, Komal}
{Nicholas, Layla, Cindy, Ahmed, Wusheng, Tushar, Alfredo, Priyanka, Komal}

k = 8

{Nicholas, Layla, Cindy, Ahmed, Wusheng, Tushar, Alfredo, Priyanka, Komal, Alfredo}

(c)

K-cliques:

A clique is a subset of nodes in a graph where every pair of nodes is connected by an edge, that is all the nodes have a direct connection to one another. A clique is a maximal complete subgraph, meaning it cannot be extended by adding more vertices while still maintaining the property that every pair of vertices is connected. Cliques are often used to represent tightly knit groups or communities in networks.

K-plexes:

A k-plex is a subset of nodes in a graph where each node has at least k-1 neighbors within the subgraph. This is an alternative way of relaxing the strong assumptions of the maximal complete subgraph seen in cliques, which allows nodes to have connections to not all nodes as long as the minimum connectivity criteria is met. K-plexes can represent communities with varying degrees of interconnectedness unlike cliques.

Question 3:

Girvan-Newman Algorithm:
1. Calculate edge betweenness for all edges in the graph.
2. Remove the edge with the highest betweenness.
3. Recalculate betweenness for all edges affected by the edge removal.
4. Repeat until all edges are removed.
Edge-Betweenness Values:

	Wusheng	Ahmed	Nicholas	Tushar	Priyanka	Akshat	Komal	Alfredo	Cindy	Layla
Wusheng	0	5.0333	0	4.7833	7	0	0	6.85	0	4.5333
Ahmed		0	5.0333	0	0	0	0	0	0	0
Nicholas			0	5.3166	0	5.6666	0	0	4.7833	3.5333
Tushar				0	0	0	4.0333	0	0	0
Priyanka					0	4.3333	0	0	4.3333	0
Akshat						0	0	0	0	0
Komal							0	3.65	6.1166	0
Alfredo								0	0	0
Cindy									0	3
Layla										0

For example, finding the edge betweenness for edge (Cindy, Layla):

Shortest paths from node Wusheng to node Cindy:
- Wusheng->Priyanka->Cindy
- Wusheng->Layla->Cindy

Shortest paths from node Priyanka to node Layla:
- Priyanka->Wusheng->Layla
- Priyanka->Cindy->Layla

Shortest paths from node Layla to node Cindy:
- Layla->Cindy

Shortest paths from node Layla to node Komal:
- Layla->Cindy->Komal

Edge Betweenness = (1/2) + (1/2) + 1 + 1 = 3

Now, removing the edge with the highest betweenness which is (Wusheng, Priyanka)

	Wusheng	Ahmed	Nicholas	Tushar	Priyanka	Akshat	Komal	Alfredo	Cindy	Layla
Wusheng	0	4.5333	0	4.2833	0	0	0	5.35	0	6.0333
Ahmed		0	6.5333	0	0	0	0	0	0	0
Nicholas			0	6.6499	0	8.3333	0	0	5.7833	4.0333
Tushar				0	0	0	4.5333	0	0	0
Priyanka					0	3.3333	0	0	7.6666	0
Akshat						0	0	0	0	0
Komal							0	5.15	7.7833	0
Alfredo								0	0	0
Cindy									0	5
Layla										0

Now, removing the edge with the highest betweenness which is (Nicholas, Akshat)

	Wusheng	Ahmed	Nicholas	Tushar	Priyanka	Akshat	Komal	Alfredo	Cindy	Layla
Wusheng	0	4.0333	0	3.7833	0	0	0	4.85	0	6.5333
Ahmed		0	6.0333	0	0	0	0	0	0	0
Nicholas			0	4.9833	0	0	0	0	8.45	2.5333
Tushar				0	0	0	4.7	0	0	0
Priyanka					0	9	0	0	16	0
Akshat						0	0	0	0	0
Komal							0	5.65	9.45	0

Alfredo								0	0	0
Cindy									0	7
Layla										0

Now, removing the edge with the highest betweenness which is (Cindy, Priyanka)

	Wusheng	Ahmed	Nicholas	Tushar	Priyanka	Akshat	Komal	Alfredo	Cindy	Layla
Wusheng	0	4.0333	0	3.7833	0	0	0	4.85	0	4.5333
Ahmed		0	4.0333	0	0	0	0	0	0	0
Nicholas			0	3.9833	0	0	0	0	3.45	2.5333
Tushar				0	0	0	3.7	0	0	0
Priyanka					0	1	0	0	0	0
Akshat						0	0	0	0	0
Komal							0	3.65	4.45	0
Alfredo								0	0	0
Cindy									0	3
Layla										0

Now, removing the edge with the highest betweenness which is (Wusheng, Alfredo)

	Wusheng	Ahmed	Nicholas	Tushar	Priyanka	Akshat	Komal	Alfredo	Cindy	Layla
Wusheng	0	2.9999	0	5	0	0	0	0	0	3.3333
Ahmed		0	4.6666	0	0	0	0	0	0	0
Nicholas			0	4.5	0	0	0	0	4.1666	2.3333
Tushar				0	0	0	6.8333	0	0	0
Priyanka					0	1	0	0	0	0
Akshat						0	0	0	0	0
Komal							0	7	6.1666	0
Alfredo								0	0	0
Cindy									0	4
Layla										0

Now, removing the edge with the highest betweenness which is (Komal, Alfredo)

	Wusheng	Ahmed	Nicholas	Tushar	Priyanka	Akshat	Komal	Alfredo	Cindy	Layla
Wusheng	0	2.6666	0	3.6666	0	0	0	0	0	3.3333
Ahmed		0	4	0	0	0	0	0	0	0
Nicholas			0	3.6666	0	0	0	0	3.3333	2.3333
Tushar				0	0	0	3.6666	0	0	0
Priyanka					0	1	0	0	0	0
Akshat						0	0	0	0	0
Komal							0	0	3.3333	0
Alfredo								0	0	0
Cindy									0	3
Layla										0

Now, removing the edge with the highest betweenness which is (Ahmed, Nicholas)

	Wusheng	Ahmed	Nicholas	Tushar	Priyanka	Akshat	Komal	Alfredo	Cindy	Layla

Wusheng	0	6	0	5.5	0	0	0	0	0	5.5
Ahmed		0	0	0	0	0	0	0	0	0
Nicholas			0	3.5	0	0	0	0	2	2.5
Tushar				0	0	0	4	0	0	0
Priyanka					0	1	0	0	0	0
Akshat						0	0	0	0	0
Komal							0	0	3	0
Alfredo								0	0	0
Cindy									0	4
Layla										0

Now, removing the edge with the highest betweenness which is (Wusheng, Ahmed)

	Wusheng	Ahmed	Nicholas	Tushar	Priyanka	Akshat	Komal	Alfredo	Cindy	Layla
Wusheng	0	0	0	3	0	0	0	0	0	3
Ahmed		0	0	0	0	0	0	0	0	0
Nicholas			0	3	0	0	0	0	2	2
Tushar				0	0	0	3	0	0	0
Priyanka					0	1	0	0	0	0
Akshat						0	0	0	0	0
Komal							0	0	3	0
Alfredo								0	0	0
Cindy									0	3
Layla										0

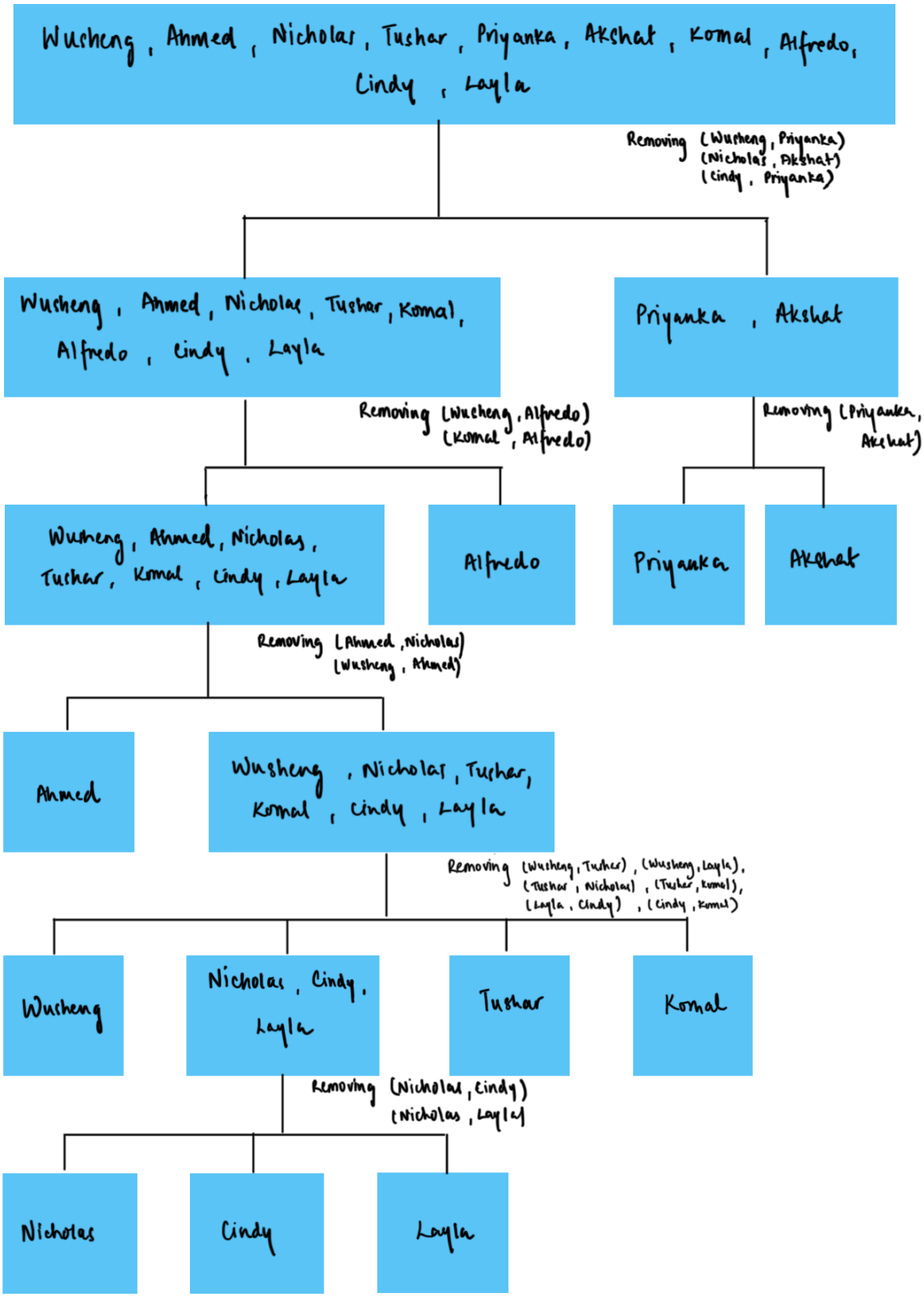
Now, removing the edges with the highest betweenness which are (Wusheng, Tushar), (Wusheng, Layla), (Tushar, Nicholas), (Tushar, Komal), (Layla, Cindy), and (Cindy, Komal)

	Wusheng	Ahmed	Nicholas	Tushar	Priyanka	Akshat	Komal	Alfredo	Cindy	Layla
Wusheng	0	0	0	0	0	0	0	0	0	0
Ahmed		0	0	0	0	0	0	0	0	0
Nicholas			0	0	0	0	0	0	2	2
Tushar				0	0	0	0	0	0	0
Priyanka					0	1	0	0	0	0
Akshat						0	0	0	0	0
Komal							0	0	0	0
Alfredo								0	0	0
Cindy									0	0
Layla										0

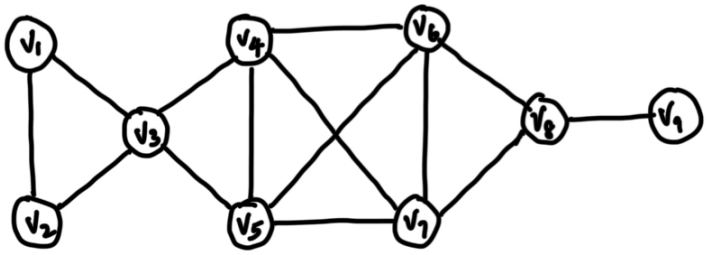
Now, removing the edges with the highest betweenness which are (Layla, Nicholas), and (Cindy, Nicholas)

	Wusheng	Ahmed	Nicholas	Tushar	Priyanka	Akshat	Komal	Alfredo	Cindy	Layla
Wusheng	0	0	0	0	0	0	0	0	0	0
Ahmed		0	0	0	0	0	0	0	0	0
Nicholas			0	0	0	0	0	0	0	0
Tushar				0	0	0	0	0	0	0
Priyanka					0	1	0	0	0	0
Akshat						0	0	0	0	0
Komal							0	0	0	0
Alfredo								0	0	0
Cindy									0	0
Layla										0

Removing the last edge (Priyanka, Akshat)


Wusheng, Ahmed, Nicholas, Tushar, Priyanka, Akshat, Komal, Alfredo, Cindy, Layla
Removing (Wusheng, Priyanka) (Nicholas, Akshat) (Cindy, Priyanka)
Wusheng, Ahmed, Nicholas, Tushar, Komal, Alfredo, Cindy, Layla
Priyanka, Akshat
Removing (Wusheng, Alfredo) (Komal, Alfredo)
Removing (Priyanka, Akshat)
Wusheng, Ahmed, Nicholas, Tushar, Komal, Cindy, Layla
Alfredo
Priyanka
Akshat
Removing (Ahmed, Nicholas) (Wusheng, Ahmed)
Ahmed
Wusheng, Nicholas, Tushar, Komal, Cindy, Layla
Removing (Wusheng, Tushar), (Wusheng, Layla), (Tushar, Nicholas), (Tushar, Komal), (Layla, Cindy), (Cindy, Komal)
Wusheng
Nicholas, Cindy, Layla
Tushar
Komal
Removing (Nicholas, Cindy) (Nicholas, Layla)
Nicholas
Cindy
Layla

Question 4: Text Ch 6 question 7



$$\sigma_{Jaccard}(v_i, v_j) = \frac{|N(v_i) \cap N(v_j)|}{|N(v_i) \cup N(v_j)|}$$

$$\sigma_{cosine}(v_i, v_j) = \frac{|N(v_i) \cap N(v_j)|}{\sqrt{|N(v_i)| \, |N(v_j)|}}$$

(a) Excluding the nodes:

$N(v_4) = \{v_3, v_5, v_6, v_7\}$

$N(v_8) = \{v_6, v_7, v_9\}$

$$\sigma_{Jaccard} = \frac{|N(v_4) \cap N(v_8)|}{|N(v_4) \cup N(v_8)|} = \frac{2}{5} = 0.40$$

$$\sigma_{cosine} = \frac{|N(v_4) \cap N(v_8)|}{\sqrt{|N(v_4)| \, |N(v_8)|}} = \frac{2}{\sqrt{4 \times 3}} = \frac{2}{\sqrt{12}} = 0.5773 = 0.58$$

(b) Including the nodes:

$N(v_4) = \{v_3, v_4, v_5, v_6, v_7\}$

$N(v_8) = \{v_6, v_7, v_8, v_9\}$

$$\sigma_{Jaccard} = \frac{|N(v_4) \cap N(v_8)|}{|N(v_4) \cup N(v_8)|} = \frac{2}{7} = 0.2857 = 0.29$$

$$\sigma_{cosine} = \frac{|N(v_4) \cap N(v_8)|}{\sqrt{|N(v_4)| \, |N(v_8)|}} = \frac{2}{\sqrt{5 \times 4}} = \frac{2}{\sqrt{20}} = 0.4472 = 0.45$$

Question 5: Text Ch 6 question 9

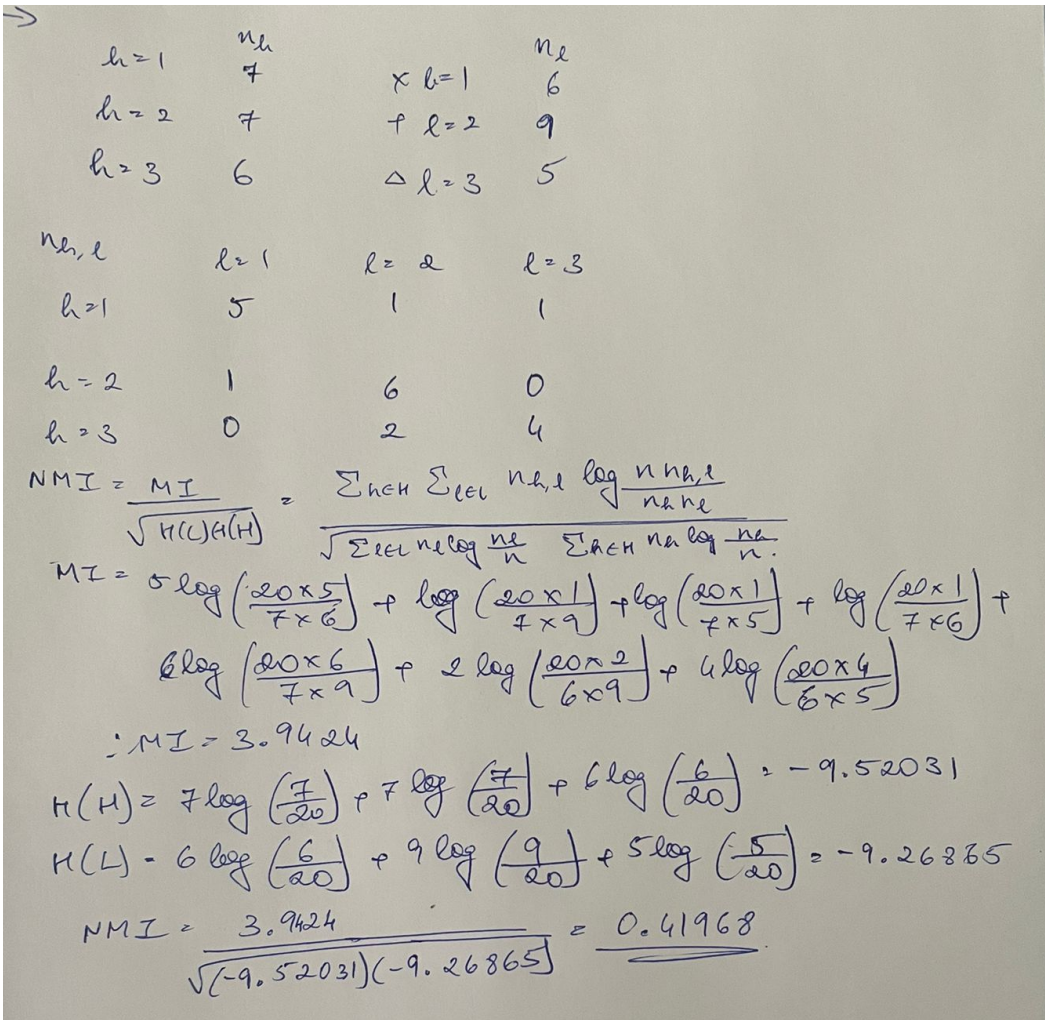
Normalized mutual information (NMI) is used to evaluate community detection results when the actual communities (labels) are known beforehand.

• What are the maximum and minimum values for the NMI? Provide details.
The maximum and minimum values for the NMI are 1 and 0, respectively. Therefore, NMI has values in range [0, 1]. An NMI value close to one indicates high similarity between communities found and labels. A value close to zero indicates a long distance between them.
• Explain how NMI works (describe the intuition behind it)
Normalized Mutual information (NMI) is calculated by normalizing mutual information (MI) which originates in information theory and describes the amount of information that two random variables share. In other words, by knowing one of the variables, NMI measures the amount of uncertainty reduced regarding the other variable. Therefore, it is a good measure for determining the quality of clustering. If there are two independent clusterings, then the NMI is zero and knowing one would not help in knowing more information about the other. Since it's normalized we can compare the clusterings that have different number of clusters.

Question 6: Text Ch 6 question 10

Compute NMI for Figure 6.15.

Figure 6.15: Community Evaluation Example. Circles represent communities, and items inside the circles represent members. Each item is represented using a symbol, +, ×, or △, that denotes the item's true label.



Question 7: Text Ch 6 question 11

Why is high precision not enough? Provide an example to show that both precision and recall are important.

Precision is more focused in the positive class than in the negative class, it actually measures the probability of correct detection of positive values. Thus, it helps in finding the proportion of positive identifications that are actually correct. Recall is the true positive rate and represents the proportion of actual positives that are identified correctly.

$$Precision = \frac{TP}{TP+FP},\ Recall = \frac{TP}{TP+FN}$$

Thus, Precision can be seen as a measure of quality, and recall as a measure of quantity. If data is imbalanced, the accuracy score can be highly misleading and that's when precision and recall are useful as in case of spam email classification where the number of spam emails are minority. However, high precision is not enough because precision does not consider false negatives, i.e., it does not account for the cases when we miss our target

event. A higher precision would indicate that the model makes fewer false positive predictions and hence, it is more likely to be correct whenever it predicts a positive outcome. However, some spam emails might be left undetected and not marked as spam getting into the inbox. In this case, recall will be useful. Thus, both precision and recall are important but there is a trade-off between them.

Question 8: Text Ch 6 question 12

Discuss situations where purity does not make sense.

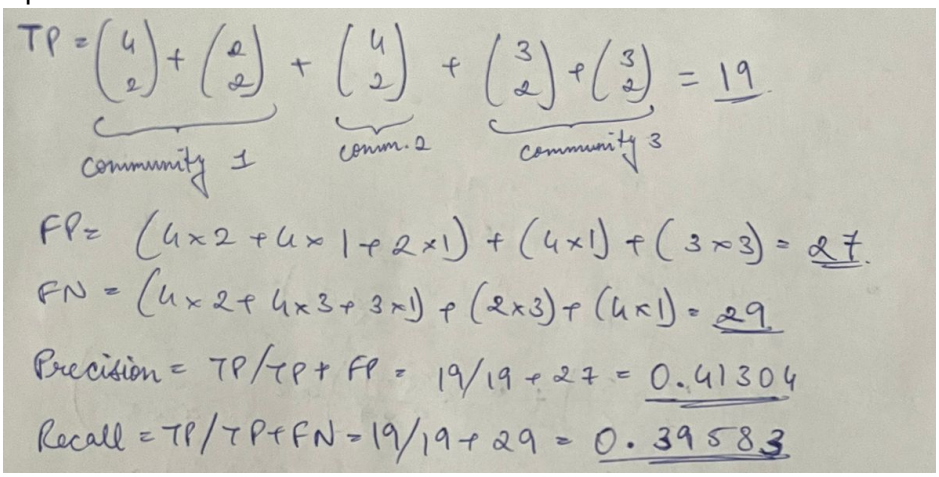
In purity, we assume that the majority of a community represents the community. Hence, we use the label of the majority of the community against the label of each member of the community to evaluate the algorithm. Purity is defined as the fraction of instances that have labels equal to their community's majority label. Purity can be easily manipulated to generate high values; consider when nodes represent singleton communities (of size 1) or when we have very large pure communities (ground truth = majority label). In both cases, purity does not make sense because it generates high values.

Question 9: Text Ch 6 question 13

Compute the following for Figure 6.17:

Figure 6.17: Community Evaluation Example.

• precision and recall



- F-measure

$$\text{F-measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = 0.40425$$

- NMI

NMI :

Found communities	n_h
$h=1$	7
$h=2$	5
$h=3$	6

Actual communities	n_l
× $l=1$	5
+ $l=2$	8
△ $l=3$	5

$n_{h,l}$	$l=1$	$l=2$	$l=3$
$h=1$	4	1	2
$h=2$	1	4	0
$h=3$	0	3	3

$$NMI = \frac{MI}{\sqrt{H(L)\,H(H)}} = \frac{\sum_{h\in H}\sum_{l\in L} n_{h,l} \log\left(\frac{n\, n_{h,l}}{n_h \cdot n_l}\right)}{\sqrt{\left(\sum_{l\in L} n_l \log\frac{n_l}{n}\right)\left(\sum_{h\in H} n_h \log\frac{n_h}{n}\right)}}$$

$$MI = 4\log\left(\frac{18\times 4}{7\times 5}\right) + \log\left(\frac{18\times 1}{7\times 8}\right) + 2\log\left(\frac{18\times 2}{7\times 5}\right) + \log\left(\frac{18\times 1}{5\times 5}\right) + 4\log\left(\frac{18\times 4}{5\times 8}\right) + 3\log\left(\frac{18\times 3}{6\times 8}\right) + 3\log\left(\frac{18\times 3}{6\times 5}\right)$$

$$\therefore MI = 2.5823$$

$$H(L) = 5\log\left(\frac{5}{18}\right) + 8\log\left(\frac{8}{18}\right) + 5\log\left(\frac{5}{18}\right) = -8.3804$$

$$H(H) = 7\log\left(\frac{7}{18}\right) + 5\log\left(\frac{5}{18}\right) + 6\log\left(\frac{6}{18}\right) = -8.5154$$

$$NMI = \frac{2.5823}{\sqrt{(-8.3804)(-8.5154)}} = 0.3056$$

- purity

$$\text{Purity} = \frac{1}{N}\sum_{i=1}^{k}\max_{j}|C_i \cap L_j| = \frac{1}{18}(4+4+3) = 0.6111$$