

## CS5228-KDDM, 2024/25-2, Coursework 3

### Introduction

- This coursework comprises two parts. Part 1 involves Python programming (1 question) for graph mining, and Part 2 contains four MCQs.
- Total CA marks of this coursework is 12.
- Details of marks/parts are below.
- A Canvas quiz will be open for your coursework submission.
- For Python programming parts, I urge you to complete a Jupyter notebook and submit it. Cw3-template.ipynb is the template for your answer. You have to run your codes and get sure that answers are available in the notebook before submission.
- Regarding MCQs, there is one and only one correct answer for each question. So select the best option. There is no penalty for wrong answers.
- The deadline for this coursework is **27/4/2025**. Please be aware that no delayed submission is possible.
- Good luck, my friends.

### CW3, Part 1: Graph Mining using Python (4 marks)

#### Task: Graph Mining with Python, **Analyzing Social Network Graphs**

---

**Notice:** The goal of Part 1 is to develop a Python program that would be able to provide answers to all questions mentioned in the **3-Procedure** section below.

#### 1- Dataset

We'll use a **Facebook** dataset from the File: **facebook\_combined.csv**

- Description: An edge list of a portion of Facebook users and their connections. First column is Facebook user and the 2<sup>nd</sup> column shows one of his/her Facebook friends.
  - Also, **names.csv** file contains the Facebook user IDs and their coded names, just in case.
  - Facebook user IDs are supposed to be between 0 and 4031.
- 

#### 2- Learning Objectives

- Practice basic graph processing using Python and its libraries.

- Perform community detection and centrality analysis.
  - Interpret the structure and insights of a real-world social network.
- 

### **3- Procedure**

#### **1. Load the Dataset**

- Read the **facebook\_combined.csv** file and create a graph using `networkx.read_edgelist()`.
- What does a node show?
- Is the graph directed or undirected?
- If we assume that the graph is undirected, will it change the methods seriously?
- Based on the adjacent matrix, how can we make it undirected?
- **Attention:** For this coursework, you don't need to change the data.

#### **2. Basic Graph Properties**

- Show Number of nodes and edges
- Can you estimate the Graph density? Is it dense or moderate or sparse?
- Is the graph connected?
- What is the Number of connected components?

#### **3. Node-Level Analysis**

- Compute the degree of each node
- Find and show the top 5 nodes with the highest degree centrality
- Find and show the node with the highest betweenness centrality
- Show: graphically if you can, print the node ID otherwise.

#### **4. Community Detection**

- Use the Girvan-Newman algorithm (`networkx.community.girvan_newman`)
- Plot the communities using matplotlib or another visualization tool

#### **5. Shortest Paths**

- Develop a function where the user enter the IDs of two nodes, then find and show the shortest path between them
- Calculate the average shortest path length for the largest connected component
- How to find the largest connected component?

#### **6. Visualization**

- Plot the graph (or the largest component) with node size proportional to degree

#### **4- Deliverables**

- A Jupyter Notebook (.ipynb) with code, answers, and plots. Use cw3\_template.ipynb as your template.
- Explanations/comments for each major step are necessary. Add them properly to the notebook.
- All the bullets of all 6 questions mentioned above should be addressed and answered in your program.

### **CW3, Part 2: MCQs (4 Questions, 2 marks each, 8 marks total)**

2. In a large undirected graph, you apply a community detection algorithm and discover several tightly connected subgraphs. Later, you observe that these communities overlap significantly. Which of the following best explains this phenomenon, and what should be considered next?

- a. The communities were incorrectly merged; the algorithm likely suffers from low modularity.
- b. The graph is likely a bipartite graph, which community detection algorithms cannot handle well.
- c. The community detection algorithm used does not support overlapping communities, and a different algorithm like Clique Percolation should be considered.
- d. The degree distribution is uniform, which artificially creates overlapping communities.

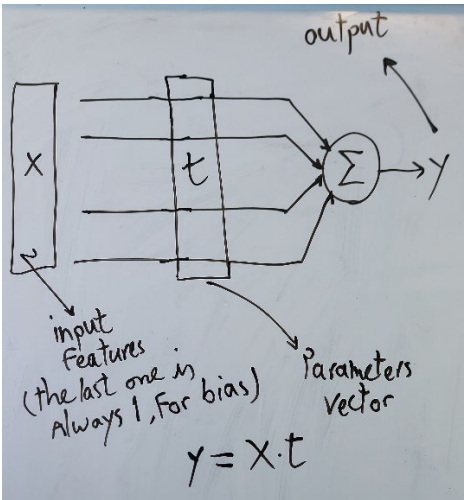
3. In a user-based collaborative filtering system, suppose two users have never rated any items in common. Which of the following consequences is most likely during the recommendation process?

- a. They will be considered highly similar due to the sparse nature of the user-item matrix.
- b. The similarity score between them will be zero, and their preferences will not influence each other's recommendations.
- c. The system will randomly assign a similarity value to ensure exploration in cold start situations.
- d. The algorithm will recommend items based on user demographic similarity instead.

4. In a high-velocity data stream environment with concept drift, you choose a sliding window approach combined with a Hoeffding Tree classifier. After several hours, model accuracy drops significantly. What is the *most plausible* cause, and what is the *most appropriate* mitigation?

- a. The sliding window is too small, failing to capture long-term trends; increasing the window size will fully recover the model’s accuracy.
- b. The Hoeffding Tree is non-incremental and doesn't adapt to concept drift; replacing it with a standard decision tree will help.
- c. Concept drift has altered the data distribution beyond the adaptive capacity of the window; augmenting with drift detection mechanisms like DDM or ADWIN is necessary.
- d. The feature space has become sparse due to windowing, so switching to clustering methods (e.g., k-means) will ensure higher accuracy.

5. In a linear regression task, we have 3 data samples, 3 numerical features per data sample, and 1 output. The input features matrix is **X**, and the output vector is **Y**; both are shown below. We will use a standard gradient descent algorithm to train the linear regressor model. The initial randomly selected parameters vector **t** is also depicted below, along with the classifier block diagram. The loss or error function used is  $\mathcal{L} = \frac{1}{2}(\hat{y} - y)^2$  and the learning rate is  $\eta = 0.1$  . What will be the loss and **t** after 2 training epochs and modification of the parameters vector twice? **t**<sup>0</sup> is the initial parameters vector, **L**<sup>1</sup> is the loss used in the 2<sup>nd</sup> training epoch, and **t**<sup>2</sup> is the parameters vector at the end of the 2<sup>nd</sup> training epoch. The last column of **X**, all 1, is to handle the bias. **y** is the target output, while **ŷ** is the predicted one.

|                                                                                     |                                                       |
|-------------------------------------------------------------------------------------|-------------------------------------------------------|
| $X = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$ | $Y = \begin{bmatrix} 0.9 \\ 0.1 \\ 0.1 \end{bmatrix}$ |
|  | $t^0 = [-2 \quad 2.1 \quad 1.3 \quad 1.5]$            |

- a.  $L^1 = 9.66$ ,  $t^2 = [-2.11, 1.7, 0.9, 1.01]$
- b.  $L^1 = 14.5$ ,  $t^2 = [-2.07, 1.87, 1.07, 1.23]$
- c.  $L^1 = 9.66$ ,  $t^2 = [0, 1, 0, 1]$
- d.  $L^1 = 14.5$ ,  $t^2 = [-3.13, 2.25, -0.81, -1.45]$

\*\*\*\*\*