DATS369 Machine Learning with Graphs Homework 1
Due: Mar 8, 2025
100 Points

Instructions

- **Collaboration policy:** Homework must be done individually, except where otherwise noted in the assignments. "Individually" means each student must hand in their own answers, and you must write and use your own code in the programming parts of the assignment. It is acceptable for you to collaborate in figuring out answers and to help each other solve the problems, and you must list the names of students you discussed this with. We will assume that, as participants in an undergraduate course, you will be taking the responsibility to make sure you personally understand the solution to any work arising from such collaboration.
- **Online submission:** You must submit your solutions online on the course Brightspace site. You need to submit (1) a PDF that contains the solutions to all questions (2) `x.py` or `x.ipynb` files for the programming questions. We recommend that you type the solution (e.g., using LaTeX or Word), but we will accept scanned/pictured solutions as well (clarity matters).
- **Generative AI Policy:** You are free to use any generative AI, but you are required to document the usage: which AI do you use, and what's the query to the AI. You are responsible for checking the correctness.
- **Computational Resource:** You are encouraged to use "Google Collab" (https://colab.research.google.com/) and Greene HPC during this class, which is free and easy to install running environment.
- **Late Policy:** Due to the spring festival, we give extra time for this homework. **No late submission is allowed.**

1 NetworkX Tutorial [5 points]

Free lunch: NetworkX (https://networkx.org/documentation/stable/) is one of the most frequently used Python packages to create, manipulate, and mine graphs. Let's see how it works by executing the cells provided in the "HW1.ipynb". You will receive full credits for successful execution. Enjoy!!!

2 Graph Basics [10 points]

Welcome to the graph field! We will load a classic graph in network science, the *Karate Club Network*. Karate Club Network is a graph which describes a social network of 34 members of a karate club and documents links between members who interacted outside the club.

Complete the missing code in the provided "HW1.ipynb" file and answer the following questions.

1. *(5 points)* What is the average degree of the karate club network?
2. *(5 points)* What is the (raw) closeness centrality for the karate club network node 5?

3 Node Classification based on Image Attributes [30 points]

We are given the **Amazon Movies** dataset, where each node represents a product sold on Amazon, and an edge between two products indicates that they have been co-purchased by certain customers.

In the provided Google Drive folder: `https://drive.google.com/drive/folders/1LQIkAbzyF4uaIydSoNY_ylXh1CCduKsn?usp=sharing`, you will find two key files: *movie_images.tar.gz* and *Movies.pt*.

movie_images.tar.gz contains the raw images, which can be extracted using the command: *tar -xzvf movie_images.tar.gz*. *Movies.pt* is a graph dictionary that consists of five keys: "adj", "label", "train", "val", and "test". "adj" and "label" represent the adjacency matrix and ground-truth labels, respectively. "train", "val", and "test" denote the training, validation, and test sets, which are used for model training and evaluation in this assignment. The objective of this task is to train an image classifier using the training and validation sets, and then evaluate its performance by reporting the accuracy on the test set.

1. *(5 pints)* Train feedforward neural networks (FNNs) for image classification. There's no limitation on the number of hidden layers, as long as the performance is good. *Hint*: transforming image into a 1D vector.

2. *(10 points)* Train convolutional neural networks (CNN) for image classification. There's no limitation on the number of hidden layers, as long as the performance is good.

3. *(15 points)* Train LSTMs for image classification. There's no limitation on the number of hidden layers, as long as the performance is good. *Hint*: transforming 2D image into a sequence of patches following `https://arxiv.org/pdf/2010.11929`

4 Network Embedding [55 points]

In this assignment, we are going to implement **Node2Vec**, one of the most popular network embedding methods in the literature. We will use gensim.Word2Vec (`https://radimrehurek.com/gensim/models/word2vec.html`) model to implement Node2Vec and test the performance on the **Amazon Movies** above. Please note that we only use the connections between products (i.e., graph structure–"adj" for Node2Vec training) for node representation learning.

1. *(30 points)* Complete the missing code in the *Node2Vec Implementation* part.

2. *(15 points)* Train the Node2Vec model to get the hidden representations of all nodes. Then, evaluate the performance of Node2Vec by building a *FNN* model for product classification. Again, there is no limitation on the number of hidden layers, as long as the performance is good.

3. *(10 points)* Hyper-parameter tuning. DeepWalk is a special case of Node2Vec by setting $p = q = 1$. How do p and q affect Node2Vec? Explore more configurations of them and show your results using figure.