DATS369 Machine Learning with Graphs Homework 2
Due: Mar 29, 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.**

Background

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 a product classifier using the training and validation sets, and then evaluate its performance by reporting the accuracy on the test set.

1 Node Classification based on GNNs [65 points]

Based on the provided **Amazon Movies** dataset, we will train a set of Graph Neural Network (GNN) models, including GCN, GAT, and GraphSAGE, for a node classification task.

1. *(5 points)* **Node Feature Preparation**. To train GNN models, we need a node feature matrix as input. In this assignment, we explore two approaches to generate node features. 1) **Pre-trained Vision-Language Model (e.g., CLIP)**: Use the image encoder from a pre-trained vision-language model such as CLIP (https://huggingface.co/docs/transformers/en/model_doc/clip) to extract image embeddings. 2) **CNN-Based Image Classifier** (from HW1): Utilize the penultimate layer output of the CNN-based image classifier trained in HW1 as image embeddings.

2. *(20 points)* **Implementing GCN for Product Classification**. Implement a Graph Convolutional Network (GCN) model that takes the node feature matrix (from question 1) and the graph structure as input. Train the model and report the test accuracy for both types of node features. Hint: Use the validation accuracy as an early stopping criterion to determine the best model.

3. *(20 points)* **Implementing GAT for Product Classification**. Implement a Graph Attention Network (GAT) model using the same setup as GCN. Train the model and report the test accuracy for both types of node features.

4. *(20 points)* **Implementing GraphSAGE for Product Classification**. Implement a GraphSAGE model using the same setup as GCN and GAT. Train the model and report the test accuracy for both types of node features.

2 Unsupervised Graph Representation Learning with GNNs [35 points]

Similar to node2vec, in this section, we aim to explore the effectiveness of Graph Neural Network (GNN) methods for unsupervised graph representation learning. Specifically, we will use the observed edges in the input graph as training signals for link prediction. For this task, we treat link prediction as a binary classification problem, where: **Positive samples** are the observed links in the input graph. **Negative samples** are randomly sampled non-existent edges (e.g., for each observed link, sample one negative edge). **Evaluation metric**: We use accuracy to assess the model's performance. Hint: To define a stopping criterion, randomly split a small set of edges as a validation set and use the remaining edges for training.

1. *(5 points)* **Link Prediction Setup**. Randomly split the observed edges in the input graph into 80% training and 20% validation sets.

2. *(10 points)* **Train a GCN Encoder for Link Prediction**. Train a GCN-based encoder using the training and validation edges from question 1. Report: 1) The training accuracy and best validation accuracy. 2) A learning curve plot, where X-axis represents the training iterations and Y-axis represents the training loss.

3. *(10 points)* **Train a GAT Encoder for Link Prediction**. Train a GAT-based encoder using the same setup as GCN. Report 1) The training accuracy and best validation accuracy. 2) A learning curve plot (same as in question 2).

4. *(10 points)* **GCN vs. GAT Comparison**. Evaluate the effectiveness of the GCN and GAT encoders using downstream classification (similar to how we evaluate node2vec), i.e., train a logistic regression model on the node representations generated by GCN and GAT in a supervised fashion. Report the best test accuracy for both GCN and GAT encoders.

3 Bonus Question [10 points]

Use the best techniques you've learned in class to improve the classification performance of GCN/GAT/GraphSAGE. **Only the top 3 individuals with the highest accuracy will receive the bonus points!**