Deep Learning on Graphs
UE22AM343BB3
Assignment 1

Dr. Maison's Pharmacy.

10th February 2025 to 4th March 2025

1 Background

Dr. Maison [1] is an excellent Doctor specialising in identifying and solving diagnostic cases no one can solve!
However, Dr. Maison's most trusted friend, confidant and most importantly pharmacist Joseph Walton[2] is away for a conference (or a holiday we don't know for sure) to the grasslands of Kenya, making him unreachable.
It so happens, that he has a bunch of cases lined up, where the conventional prescribed medicine is not available in the city! Confident in his diagnosis, Dr. Maison looks for backup pharmacists for suggesting alternate drugs to help him get the patients all better!
Dr. Maison needs YOUR help to figure out the best courses of action for each of his cases!

2 Your Task

You are provided with two tasks based on graph concepts, which are as follows:

- ***Task 1:*** Suggest *alternate drugs* based for *each disease case* on the dataset. (Banana)
- ***Task 2:*** Suggest *alternate drugs* **based on patient's specific history** based on the dataset. (Orange)

3 How to do your tasks?

- For ***Task 1:*** The expected approach is to leverage knowledge graph embeddings to determine the best probable *links* between the **candidate drugs** and **target disease**.
- For ***Task 2:*** The expected approach here is to leverage a **GCN** and explain why in your work. You can use **Pytorch** family of tools.

4 Dataset

For this task you are going to leverage the dataset Hetionet. You can find the dedicated resources about the dataset at there website, linked here.

For your ease, the datasets can be conveniently loaded from **PyKeen**, through `Pykeen.datasets()`.

[1] He is not called House or Homes for the sake of originality

[2] Again, not called Wilson or Watson because ;)

5 Task 1: Alternate Drugs Beyond Direct Connections

5.1 Problem Description

- **Goal:** Suggest candidate drugs for a given disease that are **not** directly connected to it in the graph.
- **Motivation:** Direct connections might capture obvious, already known associations. By excluding these, you aim to discover drugs that are indirectly related to the disease, thereby opening up possibilities for drug repurposing.

5.2 Suggested Approaches

- **Knowledge Graph Embeddings (KGE):** Learn embeddings for all entities using models such as TransE, DistMult, or ComplEx. Then, compute similarity scores between the disease and compounds. Finally, filter out compounds that are directly connected to the disease in the graph.
- **Hybrid Models:** Use KGE embeddings as initial features for a Graph Neural Network (e.g., GCN or GraphSAGE) and design a link prediction model that explicitly excludes direct connections.

5.3 Test Case JSON Fields

- `"type": "alternate_drug_global"`
  Indicates that this test case requires recommending alternate drugs while excluding those with a direct relationship to the disease.
- `"disease_id": <integer>`
  The unique identifier (ID) of a disease node in the graph.
- `"criteria": {"exclude_direct": true}`
  Specifies that any compound directly connected to the disease in the graph should be excluded.
- `"eval_metric": "NDCG@3"`
  The evaluation metric is Normalized Discounted Cumulative Gain at rank (NDCG@3), which measures the quality of ranked recommendations.

6 Test Case Type 2: Alternate Drugs With Side Effect Constraints

6.1 Problem Description

- **Goal:** Suggest candidate drugs for a given disease while considering potential side effects.

- **Motivation:** When repurposing drugs, it is important not only to consider efficacy but also safety.

6.2 Suggested Approaches

- **Graph Neural Networks (GCN/GraphSAGE):** These models can leverage the structural and attribute information of the graph, incorporating side effect information into the node features.
- **Filtering Post-Processing:** Use a pre-trained embedding model (or your own GNN) to generate compound scores, then filter out candidates whose side effect risk exceeds a specified threshold.

6.3 Test Case JSON Fields

- `"type": "alternate_drug_narrowed"`
  Indicates that this test case requires drug recommendations while considering side effect constraints.
- `"disease_id": <integer>`
  The unique identifier (ID) of a disease node in the graph.
- `"criteria": {"avoid_side_effects": true, "side_effect_threshold": <float>}`
  Specifies that drugs with a side effect risk exceeding the given threshold should be excluded.
- `"eval_metric": "Hits@3"`
  The evaluation metric is Hits@3, which measures whether at least one of the top 3 recommended drugs meets the desired criteria.

7 Final Notes

By following this guide and understanding each component of the test case JSON, you will be well-prepared to design and implement your own drug recommendation models.

8 Deliverables

- Code Implementation (Jupyter Notebook or Python script).
- **Model Performance Report (2-3 pages)**: Describe your approach, methodology, results, and limitations.

9 Deliverables

- Code Implementation (Jupyter Notebook or Python script).
- **Model Performance Report (2-3 pages)**: Describe your approach, methodology, results, and limitations.

10 Submission details

You will be required to submit your work on this google form.