Problem Statement - I

Graph Neural Network for Predicting Academic Collaborations and Citations

1. Introduction

Academic research is fundamentally driven by collaboration. Researchers routinely form partnerships to combine expertise, share resources, and drive innovation. Simultaneously, the flow of citations creates a dynamic network of influence and impact. Recognizing and predicting these evolving patterns not only provides strategic insights for funding agencies and research institutions but also helps individual scholars understand emerging trends and potential areas for interdisciplinary work.

Graph Neural Networks (GNNs) have recently emerged as powerful tools for modeling complex relationships in structured data. By leveraging a GNN, one can learn meaningful low-dimensional representations (embeddings) of entities and predict future interactions in a network. In this assignment, your task is to build a system that predicts whether two academic authors, who currently do not have a collaboration link, are likely to collaborate (or form a significant citation relationship) in the future.

---

2. Problem Objectives

2.1 Primary Objective

Develop and evaluate a Graph Neural Network (GNN) model that predicts the likelihood of future collaborations or significant citation interactions between academic authors.

2.2 Specific Objectives

- **Data Curation & Graph Construction:**
    - Build a dataset comprising 100–300 academic authors along with their associated research metadata.
    - Extract vital features for each author such as:
        - **Display Name:** The primary name of the author.
        - **Affiliated Institution:** Institution or university where the author is (or has been) associated.
        - **Publication Count:** The total number of publications (works) the author contributed to.
        - **Citation Count:** The cumulative number of citations received by the author.
    - Establish edges in the graph:
        - **Co-authorship Edges:** Connect two authors if they have co-authored at least one paper.
        - **(Optional) Citation Edges:** Optionally include additional edge types for significant citation relationships.
    - Ensure the dataset is stored in a structured and accessible format (e.g., CSV or JSON).
- **GNN Model Development:**

    - Design a GNN model (e.g., using Graph Convolutional Networks or GraphSAGE) that processes the graph and learns node embeddings representing each author.
    - Implement a link prediction module capable of assessing unconnected node pairs (authors) and predicting the probability of them establishing a new collaboration (or citation relationship) in the future.
    - Incorporate necessary training strategies, such as negative sampling for non-existent edges, and use an appropriate loss function (e.g., binary cross-entropy).
- **Model Evaluation & Analysis:**
    - Evaluate the model using well-established metrics for link prediction, such as accuracy, Area Under the Receiver Operating Characteristic Curve (AUC), and F1-score.
    - Visualize the learned node embeddings using dimensionality reduction techniques (such as t-SNE or PCA) to explore clustering and network structure.
    - Analyze which features (e.g., high publication counts, institutional overlap) significantly contribute to collaboration likelihood.

---

3. Data Sources and Curation Guidelines

3.1 Data Sources

To build your dataset, you are required to use publicly available academic data sources. Recommended sources include:

- **OpenAlex API:**
  Provides comprehensive, up-to-date information on academic publications and authors.
- **DBLP:**
  A long-standing bibliographic database for Computer Science which offers structured publication data.
- **Microsoft Academic Graph (MAG):**
  Now largely superseded by OpenAlex, but its legacy datasets are also acceptable if they meet the data requirements.

Note: While the problem statement emphasizes using real scholarly data, you are encouraged to choose the data source(s) that best meet your needs.

3.2 Data Curation Requirements

- **Domain and Temporal Filtering:**
    - Focus exclusively on Computer Science publications.
    - Only include papers written in English.
    - Limit the dataset to publications dated from January 1, 2020, to December 31, 2024.
    - Use appropriate filters (e.g., concept IDs, date filters) to enforce these requirements.
- **Metadata Extraction:**
    - For each paper, extract the list of authors, and record each author’s unique ID and display name.
    - Aggregate metrics for each author across all included papers:

        - **Works Count:** Increment every time an author appears in a paper.
        - **Citation Count:** Sum the citations of the papers an author appears on.
    - Enrich each author's record with affiliation details by querying for additional metadata if available.
- **Graph Construction:**
    - **Nodes:** Create a node for each unique author and attach their features.
    - **Edges:**
        - For every publication, generate all unique pairs of authors (each pair represents a co-authorship relationship).
        - Store edges such that if two authors have collaborated on multiple papers, the edge is recorded only once.
- **Subsetting:**
  Given the potential volume of authors, you must design a strategy to reduce the final graph to a manageable size (100–300 nodes). For example, you might select the top 200 authors by publication count and filter the edges accordingly.
- **Data Export:**
  Save the final nodes and edges in CSV or JSON format as separate files (e.g., `openalex_authors_nodes.csv` and `openalex_coauthorship_edges.csv`).

---

4. Model Implementation Requirements

4.1 GNN Architecture

- **Framework:**
  Implement your model using a graph deep learning library such as PyTorch Geometric, DGL, or a comparable framework.
- **Architecture Options:**
  Your solution should include a GNN with 2–3 layers. Examples include:
    - Graph Convolutional Networks (GCN)
    - GraphSAGE
- **Link Prediction Module:**
  The final part of the model should take two node embeddings and compute a score representing the likelihood that a new edge (collaboration or citation) will be formed. This could be achieved using simple similarity measures (e.g., dot product) or by concatenating embeddings followed by a Multilayer Perceptron (MLP).

4.2 Training Procedures

- **Edge Sampling:**
  Use negative sampling to generate training examples of non-collaborating author pairs.
- **Loss Function:**
  Adopt a binary cross-entropy loss for the link prediction task.
- **Hyperparameter Tuning:**
  Experiment with various hyperparameters (learning rate, number of layers, hidden units, dropout rate) to optimize model performance.

- **Evaluation:**
  Evaluate the model on a validation/test set of node pairs (edges) using metrics such as accuracy, AUC, and F1-score.

4.3 Analysis and Visualization

- Visualize the node embeddings using t-SNE or PCA to explore how authors cluster based on their features.
- Provide an analysis of which features (publication count, citation count, institutional similarity) appear to have the most influence on the prediction of future collaborations.

---

5. Deliverables

1. **Dataset Files:**
   - **Nodes File:**
     A file (CSV/JSON) detailing author information (author ID, display name, institution, publication count, citation count).
   - **Edges File:**
     A file (CSV/JSON) listing co-authorship relationships between authors (optionally including both author IDs and names).
2. **Model Code:**
   - Fully documented source code that covers data preprocessing, graph construction, the GNN model, training, and evaluation.
   - Clear commit history reflecting iterative progress and incremental improvements.
3. **Assignment Report/README:**
   - A comprehensive report detailing:
     - Data curation and graph construction methods.
     - GNN architecture design and training methodology.
     - Evaluation results along with visualizations of the network and embeddings.
     - Discussion of encountered challenges, insights into model performance, and directions for future work.
4. **Visualizations:**
   - Graph visualizations (showing the co-authorship network structure) and plots of the learned embeddings.

---

6. Submission Guidelines

- **Documentation:**
  Ensure all code is well-commented, and provide a README or similar document that explains your design decisions, methodology, and instructions on how to run your code.
- **Data Integrity:**
  The dataset should accurately reflect the filtered criteria (Computer Science, English publications from

2020–2024) and be stored in a standard format for further analysis.

- **Evaluation:**
  Clearly present performance metrics and visualization outputs, accompanied by analytical insights regarding the factors influencing collaboration.

---

7. Final Notes

While the problem statement provides comprehensive requirements and clear deliverables, you are encouraged to explore creative approaches in data preprocessing, graph construction, model design, and feature engineering. Your ability to innovate, properly document your process, and clearly explain your results will be critical to the success of the assignment.