Project Proposal Statement

Course: CSE 474 / CSE 5074 Social Network Analysis
Project Title: Temporal Link Prediction Using Graph Neural Networks
Library: PyTorch Geometric

Abstract

This project focuses on tackling the task of **Temporal Link Prediction**, an essential problem in temporal graphs where the objective is to forecast the existence of an edge between two nodes within a specified time window. Unlike traditional link prediction, temporal link prediction incorporates the temporal dynamics of graphs, making it applicable to real-world scenarios such as social network event forecasting, demand prediction in e-commerce, and user engagement in recommender systems.

In this project, students will build a unified machine-learning model that generalizes across two different graph datasets:

- **Dataset A:** A dynamic event graph with entities as nodes and events as edges.
- **Dataset B:** A user-item interaction graph where users and items are nodes and interactions are edges.

Students will use **PyTorch Geometric Temporal** to implement and optimize models capable of predicting whether an edge of a given type will form between two nodes within a future time interval.

Project Objectives

- Implement a graph neural network-based model for **Temporal Link Prediction** using PyTorch Geometric Temporal.
- Handle large-scale temporal graph data efficiently.
- Predict the probability of edge formation between given nodes within a specific time range.
- Develop a solution that generalizes well for both dynamic event graphs (Dataset A) and user-item interaction graphs (Dataset B).

Datasets Description

Dataset A

- **edges_train_A.csv:** Temporal edges with source node, destination node, edge type, and timestamp.

- **node_features.csv:** Node features with categorical attributes (-1 indicates missing values).
- **edge_type_features.csv:** Edge type features with anonymized categorical attributes.

Dataset B

- **edges_train_B.csv:** Temporal edges with source node, destination node, interaction type, timestamp, and edge features.

Test Data

- Two files: **input_A.csv** and **input_B.csv** containing test queries specifying source node, destination node, edge type, start time, and end time.

Expected Output:
Two files **output_A.csv** and **output_B.csv**, each containing the predicted probability of edge formation for the test queries.

Submission Requirements

- **Code:** Complete Python implementation of the model and preprocessing scripts.
- **Report:** Detailed documentation explaining the methodology, architecture, experimental setup, results, and insights.

Evaluation Metrics

- **Performance will be assessed using T-Scores calculated from the AUC (Area Under the Curve) values for both datasets.**
- **The final ranking score will be computed as:**

$$Score = \frac{TScore_A + TScore_B}{2}$$

where

$$TScore = \frac{AUC - mean(AUC)}{std(AUC)} \times 0.1 + 0.5$$

Methodology

1. **Data Preprocessing:** Handle missing node features in Dataset A and edge feature sparsity in Dataset B.
2. **Model Architecture:** Design a GNN model that incorporates temporal graph representations.
3. **Training:** Train the model on both datasets with shared architecture but adaptable hyperparameters.
4. **Evaluation:** Validate the model on test data to predict edge probabilities.
5. **Optimization:** Fine-tune hyperparameters to achieve balanced performance across both datasets.

Resources

- **Datasets:**

  https://github.com/dglai/WSDM2022-Challenge

  https://www.dgl.ai/WSDM2022-Challenge/

- **PyTorch Geometric Temporal:**

  https://github.com/benedekrozemberczki/pytorch_geometric_temporal

- **Dynamic GNN Implementations:**

  https://github.com/manuel-dileo/dynamic-gnn

This project will provide students with practical experience in temporal graph neural networks, preparing them to solve complex graph-based problems in dynamic environments.