Graph Neural Networks driven Recommender Systems

Vertiefungsprojekt I (HS22), Master of Eng. in Data Science

Student: Roman Loop

Academic Supervisor: Shao Jü Woo

Project duration: Start: 10. October 2021 and End: 15. Jan. 2022

Project description

Recommender systems are the secret ingredient behind personalized online experiences and powerful decision-support tools in retail, entertainment, healthcare, finance, and other industries. Recommender systems work by understanding the preferences, previous decisions, and other characteristics of many people. For example, recommenders can predict the types of movies an individual will enjoy based on the movies they've previously watched.

The three key objects managed by recommender systems are users, items and user-item interactions. These objects are tightly connected with each other and influence each other via various relations. For this very reason, recommender systems can be most naturally modelled by means of graphs through which the complex and heterogeneous nature of the available amount of information and data can be captured. It is therefore not surprising that in recent years the integration of graphs into recommender systems has attracted considerable attention from researchers and practitioners.

In a graph, the nodes correspond to entities (users and items), and edges correspond to relations between entities. Entities and their attributes, can be mapped into a graph to understand the mutual relations between them

As a graph learning technique **Graph Neural Networks** (GNN) will be applied. GNNs have recently become very popular due to their ability to learn complex systems of relations or interactions arising in a broad spectrum of problems. They have proven to be among the best performing architectures for a variety of graph learning tasks. The key idea in GCNs is to learn how to iteratively aggregate feature information from local graph neighborhoods using neural networks. This aggregation step allows each node to learn a more general node representation from its local neighborhood.

The areas covered by the project are Graph Neural Network techniques and graph databases.

Main Steps in Project

1. Study basic theory on graph neural networks

Graph Neural Networks (GNN)

- (303) Understanding Graph Neural Networks | Part 1/3 - Introduction - YouTube
- (303) Understanding Graph Neural Networks | Part 2/3 - GNNs and it's Variants - YouTube
- (303) Understanding Graph Neural Networks | Part 3/3 - Pytorch Geometric and Molecule Data using RDKit - YouTube
- (303) How to use edge features in Graph Neural Networks (and PyTorch Geometric) - YouTube

Transformer

The Transformer architecture has become a dominant choice in natural language processing and has been applied in GNN's.

- A nice tutorial explaining Transformers which was proposed in the paper *Attention is All You Need*: https://jalammar.github.io/illustrated-transformer/
- Paper attention is all you need…

2. A literature review of papers dealing explicitly with recommender systems

The following publications / Youtube Videos should be studied as part of the literature review process in this VP as they will provide a sound theoretical basis for the project:

- Wu et al. (2022): Graph Neural Networks in Recommender Systems: A Survey
- Building a Real-time Recommendation Engine With Neo4j: https://www.youtube.com/watch?v=wbI5JwIFYEM
- Integrate Neo4j with PyTorch Geometric to create recommendations: https://towardsdatascience.com/integrate-neo4j-with-pytorch-geometric-to-create-recommendations-21b0b7bc9aa
- How Uber uses Graph Neural Networks to recommend you food: https://www.youtube.com/watch?v=9O9osybNvyY

Remark: https://github.com/wusw14/GNN-in-RS provides a long list of papers related with GNN based recommender systems.

3. Building the graph neural network-based recommender model

The goal is to use a prediction model that outputs a score that describes probability that a startup will be successful. The features that we will use are inspired by the papers studied in the literature review process performed in the previous step.

3.1 GNN Models for computing the embeddings

As a first step we need to generate the input for the prediction model which obtained by aggregating & extracting semantic information simultaneously from the complex knowledge graph. This process is also referred to as **graph embedding**. This step will be performed using graph neural networks (GNNs) which belong to a category of neural networks that operate naturally on data structured as graphs. GNN's are currently a very active area of research.

The three graph embedding models that will be studied in this project are described in three very recent publications:

1. Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W.L., and Leskovec, J. (2018): Graph convolutional neural networks for web-scale recommender systems. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 974–983.
2. Donghan Yu et al. (2021): Knowledge Embedding Based Graph Convolutional Network (Paper and code available at https://github.com/PlusRoss/KE-GCN)
3. Rampasek et al. (2022): GraphGPS: General Powerful Scalable Graph Transformers (Paper and code available at https://github.com/rampasek/GraphGPS)
   - A blog article discussing the paper: GraphGPS: Navigating Graph Transformers | by Michael Galkin | Towards Data Science

3.2 Link prediction model

Phase 1: Once the GNN has transformed the initial node features to node embeddings, the next step is to build a model that predicts the interaction between users and items. The goal in link prediction is to learn, from an observed set of edges, a model that predicts the weight of unseen edges. In the case of a movie recommender system, predicting this weight is equivalent to predicting what **rating** the user would give to an item.

- **GNN software library**
  The software library available for graph neural network development: PyTorch Geometric.

- **Graph database**
  We will use Neo4j as our graph database. Free access has been obtained based on an Neo4j's Neo4j Educator Program. See: Neo4j - Quick Guide (tutorialspoint.com)

- **Dataset**
  On https://www.kaggle.com/ many datasets are available for recommender system development. Among them only few can be found in which also user attributes (demographics) are provided. In practical recommender systems, user profiles are available, hence it is important to use datasets that contain personal information on users. One of these datasets is the **MovieLens 100K** dataset. This data was collected

through the MovieLens web site (movielens.umn.edu) from 1997 through 1998 and contains the following user attributes: user id, age, gender, occupation, and zip code. This dataset can be downloaded from: Movielens Rating Dataset | Kaggle

Phase 2 (tentative): If time permits in this project, one could extend the model by considering the fact that users might change their perception of items over time. In this case, the use of temporal contextual information should improve link prediction results in recommendation tasks. Two papers should be consulted for the second phase of building the prediction model:

1. Experiments performed on the widely used MovieLens-100k movie recommendation datasets suggest that the use of dynamic graphs lead to better link prediction results than those observed for static baseline models [Link Prediction in Dynamic Graphs for Recommendation, Fadel et al., 2018].
2. A more recent paper in which dynamics graphs are studied is [Parameter-free Dynamic Graph Embedding for Link Prediction, Liu et al., 2022]. The developed prediction model is shown to outperform state-of-the-art methods in accuracy on two link prediction tasks.