Degree Project in Computer Science and Engineering

Second cycle, 30 credits

Lightweight Memory Networks for Link Prediction

Generalization of the LiMNet Architecture

TITOUAN MAZIER

Lightweight Memory Networks for Link Prediction

Generalization of the LiMNet Architecture

TITOUAN MAZIER

Master's Program, Computer Science
Date: April 15, 2025

Supervisors: Šarūnas Girdzijauskas and Lodovico Giaretta
Examiner: Viktoria Fodor
School of Electrical Engineering and Computer Science
Host company: RISE, Research Institute of Sweden
Swedish title: Lightweight Memory Networks för Länkprediktion
Swedish subtitle: Generalization av LiMNet Arkitektur

Abstract

Lorem ipsum dolor.

Keywords

Graph Representation Learning, Temporal Interaction Networks, Link Prediction, Data Mining, Machine Learning, Recommendations

Sammanfattning

Lorem ipsum dolor.

Nyckelord

Graph Representation Learning, Temporal Interaction Networks, Link Prediction, Data Mining, Machine Learning, Recommendations

Acknowledgments

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do.

Lorem.

Lorem ipsum dolor.

Stockholm, April 2025
Titouan Mazier

Contents

Acronyms and Abbreviations

Chapter 1
Introduction

Lorem ipsum dolor sit amet.

1.1. Problem

Lorem ipsum.

1.2. Purpose

epoch Lorem ipsum dolor sit amet.

1.3. Goals

Lorem.

1.4. Structure of the Thesis

Lorem ipsum dolor sit amet.

Chapter 2
Background

This chapter provide the background for the project. In Section 2.1 we provide an overview of link prediction and the classical solutions for the problem, in Section 2.2 we further develop the concept of graph embeddings and some common methods to create them. Then in Section 2.3 we present in details our model of interest and why we believe that it is a relevant addition to the task of link prediction.

2.1. Link Prediction

The rapid expansion of digital content has resulted in an overwhelming abundance of information, posing significant challenges for users seeking relevant and meaningful material. A consequence to this information overload is the growing importance of search engines and recommendation systems. Among the many techniques that have emerged to tackle this challenges, one of the most used is content personalization. Instead of filtering the information in the same way for everyone, the systems will use the user's context: their search history, demographics, pasts interactions with the system, etc. to filter the information to display. For example, a system can identify the user's location through localization or search history and filter out any local information that does not match her current localization.

Content personalization can commonly be represented as a link-prediction problem in a user-item graph. In such a graph, each user and each item is associated to a node. An item can be any kind of information the user is interested in, such as web pages, music tracks, items in an e-commerce catalog, ... For each interaction a user will have with the system, it will register as an edge in the graph, the goal of the personalization system is to find which item is the most relevant for a given user, which is the same as predicting which interaction should be added next in the graph.

A classical approach to that problem is to measure how close each item is to the user in the graph. Research in graph theory has provided us with a range of different ways to compute closeness between two nodes, such as

measuring the shortest path connecting them, how many neighbors they share or how exclusive their common neighbors are.

The main issue with this approach is that it can only leverage structural data from the underlying graph while most of the applications provide rich context features for the users and the items. Features like such are typical inputs for Machine Learning systems, that inspired Lichtenwalter et al. to approach link prediction as a supervised problem[1]. Given a user and an item, the task is now to evaluate how likely it is that an edge will form between them in the future, based on the knowledge of the user features, the item features and a range of similarity measures computed on the past interactions graph.

This typical machine learning setup leads to switch from a straightforward prediction setting to a feature engineering approach when it comes to graph-based data. Instead of looking for the desired property in the structure of the graph, this approach will try to summarize the structure into rich representation compatible with machine learning algorithm. The goal is not to proxy the desired property, but to create a rich representation of the graph data that can be used by a machine learning algorithm.

2.2. Graph Representation Learning

- The task of learning embeddings from graphs is called GRL and have applications outside link-prediction: node-level, edge-level, graph-level (molecule graphs, ask gpt).
- Two approaches have been used so far: random walk based and GNNs. Random walk are used to create node sequences that are then fed to sequence based models of the same kind as Language models. GNNs are NN using the structure of the graphs to pass information through nodes.

(- Practical examples of GNNs in for link prediction: GraphSAGE)

- Recently, researchers have been trying to exploit even more information with the addition of dynamic graphs. Until there, the temporality of the interactions did not have impact on the recommendations. e.g. listening session on spotify.
- There are 2 way of representing dynamic graphs: DTDG and CTDG. We are interested in the latter since it better represent the reality.
- CTDG can be seen as a stream of interactions instead of a stable graph structure. Thus, graphs concepts such as neighborhood become blurred.

- Approaches to cope with this added dificulty include using a window of past interactions to feed into a ML system (e.g. cross-attention with DeePRed).
- A common solution since deepcoevolve is to use cross-RNN. The system keeps a memory of each node and will have these memories “interact” to update each other whenever an interaction is observed.
- This approach has the benefit of conserving the causality of interactions into the embeddings.

2.3. LiMNet

- LiMNet is such a solution that aims at being as lightweight and simple as possible, using only one RNN cell to compute the embeddings. This has the double benefit of making it very cheap to run but also very flexible with node insertion and deletion being trivial operations.
- Description of LiMNet Architecture
- LiMNet has proven effective on the task of botnet and fraud detection but was not initially designed to tackle link prediction. Which is what this work aims to do.

Chapter 3
Method

References

[1] R. N. Lichtenwalter, J. T. Lussier, and N. V. Chawla, "New perspectives and methods in link prediction," in *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, in KDD '10. Washington, DC, USA: Association for Computing Machinery, 2010, pp. 243–252. doi: 10.1145/1835804.1835837.

TRITA – EECS-EX 2024:0000
Stockholm, Sweden 2025
www.kth.se