NLP - Graph Embedding

A Brief Introduction into GE Models utilizing Random Walk

Supervisor: M.Sc. Matthias Assenmacher

Noah Hurmer

tbd

Abstract

TODO

Contents

List of Tables

List of Figures

List of Abbreviations

Term	Abbreviation
DL	Deep Learning
GE	Graph Embedding
RW	Random Walk

1 Introduction

1.1 What is a graph

Graphs are a type of data structure that consist of so called nodes (or vertices) and edges. Often times, the nodes describe entities and the edges the pairwise relation to one another. Therefore, a graph G is usually noted as $G = (V, E)$. Here, V is the set of nodes and E the set of edges.
In essence, this can be thought of as a network of entities. One of the simpler realizations to imagine is perhaps a social media platform. Where a person (or their profile) is represented as a node and an edge indicates that two people are befriended on the platform. As apparent in the above mentioned example, graphs can have edges that embody abstract concepts, that do not intuitively fit into any conventional coordinate system, as they cannot be mapped into an euclidean space. That being said, one can also imagine more conventional observed data of a large feature space as a graph, where the edges are weighted and represent a similarity between objects.

1.2 Graph types

The type of entity that a node represents can be in any form of data, be that words or text, images

1.3 Applications

- uses in social networks
- visualization
- Network compression
- Network Partitioning
- Node Classification
- Link Prediction
- fake news detection

perhaps omit some examples as to not overcrowd or overcomplicate this passage.

1.4 Graph types

- What is a Graph?
- Nodes and edges
- Nodes are the entities
- Edges are the relations between nodes
- nodes can consist of all sort of Data, images, text, etc even mixed
- nodes can have additional attributes

1.4.1 Graph modes

- homogeneous, heterogeneous
- graphs can be directional, weighted, semantik, knowledge based

1.5 Motivation

Two of the main issues with data in the form of a graph are its inherent structure and the limited applicable mathematics available to deal that, and the computational challenge associated with any type of storage or calculation performed on it.
The above described composition of a graph is usually stored in a so called adjacency matrix, with the dimensions of $N \times N$, where N is the number of nodes in the graph. The edges are then captured with a binary indicator (or a value for a weighted edge), whether or not two nodes are connected via an edge. Not only does this become a problem if the relationship the edge represents becomes more complex, but the sheer size of such matrices can quickly become a problem both in form of (dynamic) storage space but also computational expense.
Therefore we aim to compress the information of a graph down to lower dimensions and into a form that lets us better apply analysis tools. Usually a vector space is selected as the embedding dimension.

1.6 Embedding types

There are different forms of graph embedding which each go along with specific tasks they are used for.

- entire Graphs can be embedded in order to compare different graphs to each other. This can for example be useful for biologist to compare proteins or predict their functional labels. Here, a complex protein can be a graph which will then be embedded to a single vector.
- It is also possible to embed subgraphs or groups of a graph separately.
- edge embedding.
- The most common way of graph embedding is to embed the nodes of a graph.

2 Problem and Dataset

introduce the Problem and the associated Dataset on which the algorithms below should be run on.

3 Deep Learning and random Walk

Short Paragraph explaining how one Approach to (node)Graph embedding utilizes Deep Learning methods. List the other approaches not using DL with a brief explanation, and especially mention how the computational advantage of the Random Walk approach can be preferable to MF.

3.1 Random Walk

Introduce the random Walk concept as another subsection of Deep Learning tools for Graph embedding. What is the Aim of the Random Walks?

3.2 SkipGram

Briefly explain the NLM SkipGram and how its paired with Random Walks. Also introduce the problem of calculating the associated softmax equation.

3.2.1 DeepWalk as an example for Hierarchical softmax

3.2.2 node2vec as an example for negative sampling

3.3 (LSTM)

Optional Section considered as an excursion to another Random Walk method using a different Model and heterogeneous Graphs as Inputs. Perhaps using HSNL as the algorithm example.

4 Performance Evaluation

Discuss how to evaluate Graph Embedding methods and algorithms and then evaluate the above used ones.

5 References

Alicia Frame, PhD. 2019. "Graph Embeddings." 2019. https://www.youtube.com/watch?v=oQPCxwmBiWo&ab_channel=Neo4j.

Cai, HongYun, Vincent W. Zheng, and Kevin Chen-Chuan Chang. 2018. "A Comprehensive Survey of Graph Embedding: Problems, Techniques, and Applications." *IEEE Transactions on Knowledge and Data Engineering* 30 (9): 1616–37. https://doi.org/10.1109/TKDE.2018.2807452.

Godec, Primož. 2018. "Graph Embeddings — the Summary." https://towardsdatascience.com/graph-embeddings-the-summary-cc6075aba007.

Goyal, Palash, and Emilio Ferrara. 2018. "Graph Embedding Techniques, Applications, and Performance: A Survey." *Knowledge-Based Systems* 151: 78–94. https://doi.org/https://doi.org/10.1016/j.knosys.2018.03.022.

Grover, Aditya, and Jure Leskovec. 2016. "Node2vec: Scalable Feature Learning for Networks." *CoRR* abs/1607.00653. http://arxiv.org/abs/1607.00653.

Hong, Shanon. 2020. "An Introduction to Graph Neural Network(GNN) for Analysing Structured Data." https://towardsdatascience.com/an-introduction-to-graph-neural-network-gnn-for-analysing-structured-data-afce79f4cfdc.

Perozzi, Bryan, Rami Al-Rfou, and Steven Skiena. 2014. "DeepWalk: Online Learning of Social Representations." *CoRR* abs/1403.6652. http://arxiv.org/abs/1403.6652.

Pilehvar, Mohammad Taher, and Jose Camacho-Collados. 2020. "Embeddings in Natural Language Processing: Theory and Advances in Vector Representations of Meaning." *Synthesis Lectures on Human Language Technologies* 13 (4): 1–175. https:

//doi.org/10.2200/S01057ED1V01Y202009HLT047.