Representation Learning for Heterogeneous Graphs

General idea

Company networks with pairwise relationships are generally represented as knowledge graphs or data models. For example, there can be several tables (Figure 1) for: Company description (name, number of employees, industry, address, ...), Industry description (name, materials used, ...), Shareholder relationship (company1, company2, relation type). Building relational counterpart to be used into deep learning architectures is crucial if we want systems that can learn, reason, and generalize from this knowledge graph data. The task could be identification of illicit companies, Companies clustering, Outlier detection in Company network. This might have several applications in real life, such as Fraud prevention, Money Laundering prevention, risk management.

The objective of this work is to transform a Company Knowledge graph into a heterogeneous graph (graph in which we have 2 or more types of nodes) with attributes and process the obtained graph. In other words, we want to find a meaningful way to 1) design our heterogeneous graph given the knowledge graph; 2) find a good representation of nodes (of all types) given the nodes’ attributes and edges; such that we get better performance and results on downstream tasks compared with other graph representation learning methods.

Tasks

Companies Graph Data are Graph data, meaning that any Graph representation learning method could be applied to it. The key point here is to learn the specificities of the heterogeneous graph that we can leverage to get a better final representation. The quality of the final representation will be assessed on downstream tasks such as node classification or link prediction.

Possible steps to be developed

1. Related work on Graph Representation Learning, Company Graph Mining, Binary and Multi-label node classification, Heterogeneous graph Mining

2. Detailed analysis of Knowledge Graphs (Classes, Relationships, attribute types), Heterogeneous network (node types, weights construction), ...

3. Model the desired Graph: Nodes, Edges, Features, Labels (for classification)

4. Study possible constraint linked to the graph nature (e.g. relation type) to be used during the representation learning process (it could also be used to define a weight function between two nodes)

![Figure 1: Data Model example](image)
5. Implement state-of-the-art and recent Graph Representation Learning methods, apply them on our Company data.

6. Define 2 or 3 downstream tasks (e.g. binary node classification, multi-labeled node classification, unbalanced node classification) and evaluate the baselines and our method on all downstream tasks

**Contact Person**

**Steve Gounoue**: steve.gounoue@cs.uni-bonn.de

- Please, send an email along with your transcript, and a link to GitHub (if available).

**Prerequisites:**

- Theoretical and practical knowledge of machine learning
- Good programming skills
- Basic Knowledge of SQL, Knowledge graph, Git

**Tools**

- Implementation using Python programming language
- Common libraries: torch, pandas, networkx, sklearn, numpy

**Some Material**

- Heterogeneous Propagation Graph Representation Learning for Fake News Detection [7]
- Cross-language Citation Recommendation via Hierarchical Representation Learning on Heterogeneous Graph [3]
- Gated Heterogeneous Graph Representation Learning for Shop Search in E-commerce [5]
- Semi-supervised graph embedding for multi-label graph node classification [2]
- Graphsmote: Imbalanced node classification on graphs with graph neural networks [8]

**Possible Datasets**

- Wikipedia knowledge graph datasets
- Wiki datasets (considered as separate datasets)
- OGB datasets

**Possible Baselines**

- R-HGNN [6], WSI-HGNN [1], PGB [4]

**References**


