Heterogeneous Network Representation Learning: A Unified Framework with Survey and Benchmark

By Yang et al.

EGOR DMITRIEV, Utrecht Univeristy, The Netherlands

1 GOALS

- we aim to provide a unified framework to **deeply summarize and evaluate existing research** on heterogeneous network embedding (HNE)
  - we first formulate a unified yet flexible mathematical paradigm of HNE algorithms
  - we propose a generic objective function of *network smoothness*, and reformulate all existing models into this uniform paradigm while highlighting their individual novel contributions
- we provide a generic paradigm for the **systematic categorization and analysis over the merits** of various existing HNE algorithms
- we **create four benchmark datasets** with various properties regarding scale, structure, attribute/label availability, and etc. from different sources, towards handy and fair evaluations of HNE algorithms
- we carefully refactor and amend the implementations and **create friendly interfaces for 13 popular HNE algorithms**

2 PRELIMINARIES

- HNE: Heterogenous network embedding
- **Hadamard product**: element-wise product

3 CHALLENGES

- real-world objects and interactions are often multi-modal and multi-typed

4 PREVIOUS WORK / CITATIONS

-
- **This Work:** ...

5 DEFINITIONS

- **Heterogeneous network**: $H = \{V, E, X, R, \phi, \psi\}$
  - $v_i \in V$: vertices, $e_{ij} \in E$: edges
  - $\phi(v_i)$: Node type, $\psi(e_{ij})$: Link type
  - X_i^o: Node attribute, U_{ij}^o: Link attribute
- **Meta-Path**: Path $o_1 \rightarrow^{l_1} o_2 \rightarrow^{l_2} ...o_m \rightarrow^{l_{m+1}} o_{m+1}$
  - Where o and l are node/link types
  - Carries semantics (composed relation)
  - Allows computing **multi-modal proximity**
- **Network embedding**: $\Phi : V \rightarrow \mathbb{R}^{|V| \times d}$
- **Heterogenous network embedding**: $\{\Phi_k : V \rightarrow \mathbb{R}^{|V_k| \times d}\}_{k=1}^{K}$
  - where K is number of node types
- **Smoothness Objective**: $\mathcal{J} = \sum_{u,v \in V} w_{uv} d\left(\boldsymbol{e}_u, \boldsymbol{e}_v\right) + \mathcal{J}_R$
  - Where $\boldsymbol{e}_u, \boldsymbol{e}_v$ are node embeddings

  - w_{vw}: proximity weight
  - $d(\cdot,\cdot)$: distance function

6 OUTLINE / STRUCTURE

6.1 Taxonomy

6.1.1 Proximity-Preserving Methods.

- Goal of network embedding is to capture network topological information
- Preserving different types of proximity among nodes
- Approaches:
  - **Random Walk Approaches** (DeepWalk [29])
    * metapath2vec: randomwalk -> skipgram (context based) (negative sampling)
    * hin2vec: max likelihood based on path count/probability approximation (but employs a negative sampling like approach)
  - **First/Second-order Proximity** (LINE [30])
    * PTE: Split into multiple bipartite networks. Per network maximize the co-occurrence objective (skipgram)
      · Instead of co-occurrence counts, edge weight is used (based on different type edge counts)
    * HEER:
      · each edge type has embedding μ_l
      · each edge has embedding g_{uw}
      · assume:
      $$\boldsymbol{\mu}_l^T \boldsymbol{g}_{uv} = \boldsymbol{e}_u^T \boldsymbol{A}_l \boldsymbol{e}_v$$
      · Again max likelihood across bi networks
  - considered as **shallow network embedding**, due to their essential single-layer decomposition

6.1.2 Message Passing Methods.

- aim to learn node embeddings e_u based on x_u by aggregating the information from u's neighbors.
- Considered as **deep network embedding** due to multiple layers of learnable projection functions
- **In unsupervised setting**: objective is link prediction
- Meta path based neighborhood: $\mathcal{N}_{\mathcal{M}}(u) = \{v|v \text{ connexts with } u \text{ via meta-path } \mathcal{M}\})$
- $a_{uv}^{\mathcal{M}}$: Learned weight of of neighbors
- $\beta_{\mathcal{M}}$: Meta path weight

6.1.3 Relation-Learning Methods.

- Knowledge Graphs are a special case of heterogeneous networks
- Explicitly model the **relation types** of edges via **parametric algebraic operators**
- Focus on the designs of triplet based scoring functions
  - learn a scoring function $s_l(u, v)$ which evaluates an arbitrary triplet (where l is relation type)
- Usually margin based ranking loss is used + regularizer
  - Which has very similar form to negative sampling loss (!)
- Works:
  - **TransE**: assume $e_u + e_l \approx e_v$ when relation l holds (translation of embedding)
    * Optimizes by maximizing margin between related and unrelated pairs

- Distmult: exploits similarity based scoring (usually $e_u^T A_l e_v$) (aka the alignment score with some diagonal matrix inbetween)
- ComplEx: utilizes complex valued representations which allows capturing asymmetric relations

7 EVALUATIONS

- Tested: Classification and Link prediction
- Proximity-preserving algorithms: **often perform well** on both tasks under the unsupervised unattributed HNE setting
- Message-passing methods: perform poorly except for HGT, especially on node classification. But are known to excel due to their integration of node attributes, link structures, and training labels (which are not available).
- Relation learning methods: perform well on link predictions (when there are a lot of link types)

8 CODE

- https://github.com/yangji9181/HNE)

9 RESOURCES

-

10 NOTE (TO SELF)

- Read more about transformers and check out [85]