Are Meta-Paths Necessary? Revisiting Heterogeneous Graph Embeddings

元路径是否必要之重识异构图嵌入

Rana Hussein
eXascale Infolab, University of Fribourg, Switzerland

Dingqi Yang*
eXascale Infolab, University of Fribourg, Switzerland
{firstname.lastname}@unifr.ch

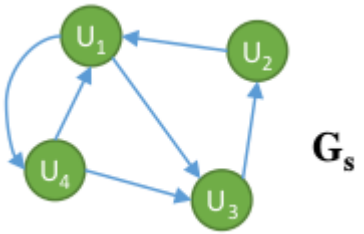
Philippe Cudré-Mauroux
eXascale Infolab, University of Fribourg, Switzerland

2018 CIKM

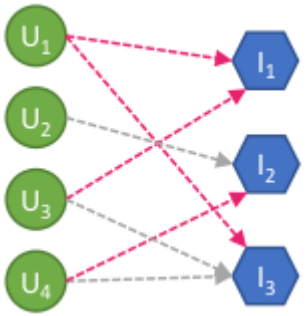
王润生
August 2, 2019

Heterogeneous Graph

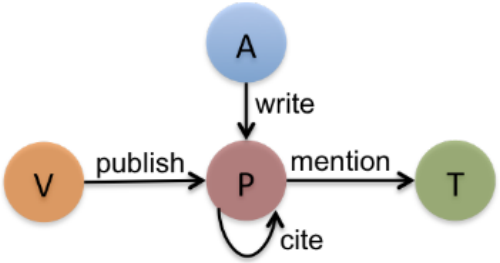
Definition 3.1. ***(Heterogeneous Graph)*** A heterogeneous graph is denoted as $G = (V, E)$, where V and E refer to the set of nodes and edges (either directed or undirected), respectively. Each node $v \in V$ is mapped to a specific data domain using a mapping function $\phi(\cdot)$, i.e., $\phi(v) = q$, where $q \in Q$ and Q denotes the set of domains in graph G. For a heterogeneous graph G, we always have $|Q| > 1$ as it contains more than one node domains.



$\mathbf{G_s}$



$\mathbf{G_r}$



(a)

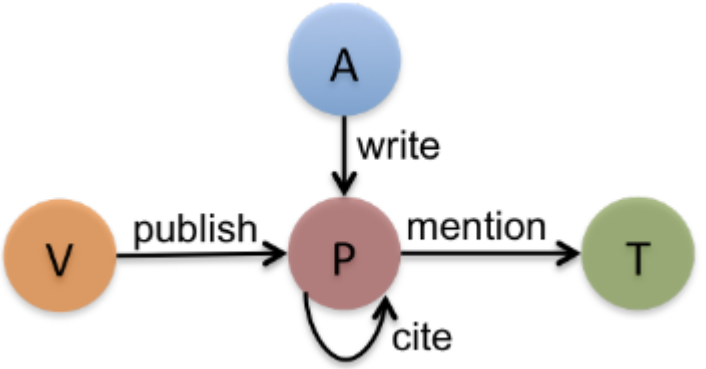
Definition 1. **Heterogeneous Information Network**: In a heterogeneous information network $\mathbf{H} = (V, E, T)$, each node v and each link e is associated with a mapping function $\phi(v) : V \rightarrow T_V$ and $\phi(e) : E \rightarrow T_E$, respectively. T_V and T_E denote the sets of object and relation types, where $|T_V| + |T_E| > 2$.

Meta-Paths

- Definition: Fixed sequences of node types

  固定了节点类型的序列

- Example
  - “A-P-A” represents a co-authorship relationship on a paper between two authors
  - “A-P-V-P-A” represents papers published by two authors in the same venue



Graph Embeddings

- network embeddings or network representation learning
  - The key idea is to represent nodes in a graph using a continuous low-dimensional vector space (i.e., node embeddings),
  - while preserving key structural properties

    (e.g., node proximity) of the graph.
- In the current literature, heterogeneous graph embedding techniques mostly rely on meta-path-guided random walks to sample node pairs for learning node embeddings.

Shortcoming of Meta-Paths

- Different meta-paths express different semantic meaning. How to select meta-paths from a given heterogeneous graph remains unclear.
- Even more critically, the existing literature has shown that the choice of meta-paths highly affect the quality of the learnt node embeddings
- It‘s dataset-specific meta-path based on prior knowledge from domain experts

Path	Schema	Description
P_1	$U \xrightarrow{p} I \xleftarrow{p} U$	Users who have consumed the same item are similar with each other
P_2	$U \xrightarrow{t} U \xrightarrow{t} U$	A user may trust her/his friends' friends
P_3	$U \xleftarrow{t} U \xrightarrow{t} U$	Users who are trusted by the same user are similar with each other
P_4	$U \xrightarrow{t} U \xleftarrow{t} U$	Users who share the same friends are similar with each other
P_5	$U \xrightarrow{t} U \xrightarrow{p} I \xleftarrow{p} U$	A user may have similar taste with someone who has similar taste with his/her trustee
P_6	$U \xrightarrow{t} U \xrightarrow{t} U \xrightarrow{p} I \xleftarrow{p} U$	Help find users of the similar tastes that are distant from each other on the social network

Revisiting Heterogeneous Graph Embeddings

- “Are meta-paths really necessary

  for heterogeneous graph embeddings?”
- Initial motivation
  - the initial motivation of using meta-paths in heterogeneous graph embeddings lies in the fact that random walks on heterogeneous graphs are biased to highly visible types (domains) of nodes

    (the node distribution from the random walk sequences are skewed towards these highly visible domains)

JUST—
Random Walk with JUmp & STay

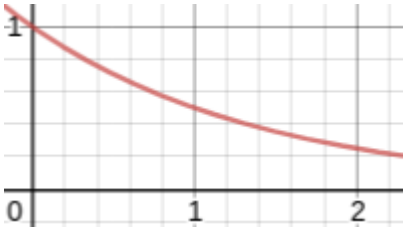
- Specifically, when performing random walks over a heterogeneous graph, we choose the next node either by jumping to one of the other data domains, or staying in the same data domain.
- The key idea of our solution is to ***probabilistically balance*** these two options, in order to avoid the above-mentioned bias.
  - 1) jump or stay;
  - 2) if jump where to jump.

JUST—
Jump or Stay

from the current node v_i,
the next node v_{i+1} is selected by one of the following options:

- **Jump** to a target domain q: uniformly sampling one node from those in a target domain q connected to v_i via heterogeneous edges. The candidate set of nodes for jumping from v_i to the target domain q is denoted as $V^q_{jump}(v_i) = \{v|(v_i, v) \in E_{he} \vee (v, v_i) \in E_{he}, \phi(v) = q\}$.
- **Stay** in the current domain: uniformly sampling one node from those connected to v_i via homogeneous edges. The corresponding candidate set of nodes is denoted as $V_{stay}(v_i) = \{v|(v_i, v) \in E_{ho} \vee (v, v_i) \in E_{ho}\}$.

$$Pr_{stay}(v_i) = \begin{cases} 0, \text{ if } V_{stay}(v_i) = \emptyset \\ 1, \text{ if } \{V^q_{jump}(v_i)|q \in Q, q \neq \phi(v_i)\} = \emptyset \\ \alpha^l, \text{ otherwise} \end{cases}$$

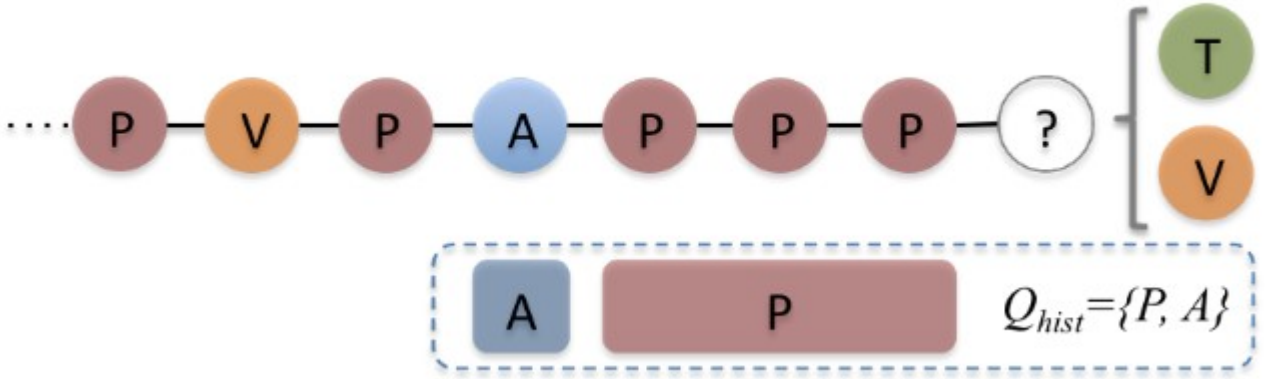


where $\alpha \in [0, 1]$ is an initial stay probability
l refers to the number of nodes consecutively visited in the same domain

JUST—
Where to Jump

4.1.2 Where to Jump? In case of a jump, given the current node v_i, a target domain q needs to be selected before performing the jump.

To achieve this goal, we define a fixed-length queue Q_{hist} of size m to memorize up-to-m previously visited domains



Algorithm 1 Truncated Random Walk with Jump & Stay

Require: A heterogeneous graph $G = (V, E)$, initial stay probability α, number of memorized domains m, number of random walks per node r, maximum walk length L_{max}

Initialize an empty set of walks $\mathcal{W} = \emptyset$
for $i = 1$ to r **do**
    for each $v \in V$ **do**
        Initialize a random walk by adding v;
        Initialize Q_{hist} by adding $\phi(v)$;
        while $|W| < L_{max}$ **do**
            Pick a Jump or Stay decision
            if Stay **then**
                Continue W by staying;
            else if Jump **then**
                Sample a target domain q
                Continue W by jumping to domain q;
                Update Q_{hist} by keeping only last m domains;
            end if
        end while
        Add W to $\mathcal{W}$
    end for
end for
return The set of random walks $\mathcal{W}$

The End

Thank You