Heterogeneous Graph

Nasy

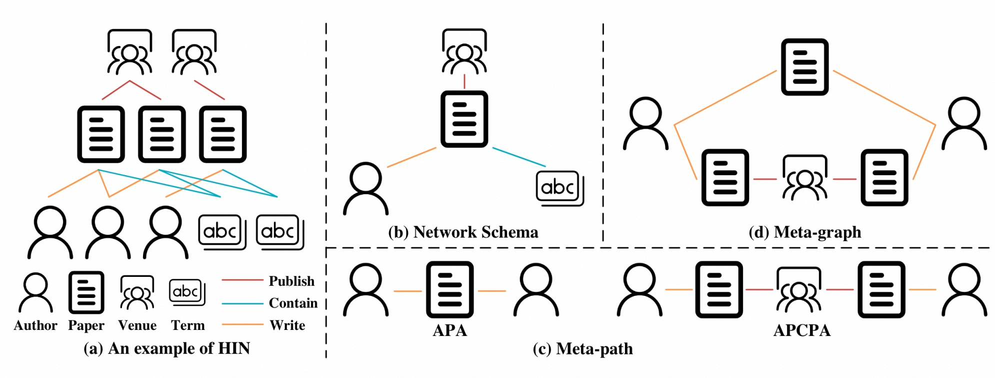
Jul 31, 2022

Outline

1. Introduction
2. RGCN
3. HetGNN
4. HAN
5. HGT
6. References

Heterogeneous Graph

Heterogeneous graph (HG), also known as heterogeneous information network (HIN).



(a) An example of HIN

(b) Network Schema

(c) Meta-path

(d) Meta-graph

Figure: Figure excerpted from[2]. Illustration of The heterogeneous graph. (a) An academic network including four types of node (i.e., Author, Paper, Venue, Term) and three types of link (i.e., Publish, Contain, Write). (b) Network schema of the academic network. (c) Two meta-paths used in the academic network (i.e., Author-Paper-Author (APA) and Author-Paper-Conference/Venue-Paper-Author (APCPA)). (d) A meta-graph used in the academic network.

[1] Xiao Wang and others, “A Survey on Heterogeneous Graph Embedding: Methods, Techniques, Applications and Sources” [2022] IEEE Transactions on Big Data 1.

[2] Xiao Wang and others, “A Survey on Heterogeneous Graph Embedding: Methods, Techniques, Applications and Sources” [2022] IEEE Transactions on Big Data 1.

Heterogeneous graph

Heterogeneous graph

A heterogeneous graph can represent as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where each node $v \in \mathcal{V}$, and each edge $e \in \mathcal{E}$ contain their own type $\tau(v)$ and $\phi(e)$. τ_v and ϕ_e are two mapping functions, where $\tau_v : V \to \mathcal{A}$ denotes node types and $\phi_e : \mathcal{E} \to \mathcal{R}$ represents edge types. The network schema is a graph defined over node types $\mathcal{A}$ and edge types $\mathcal{R}$ following the relations.

Meta-path

A meta-path m is based on a network schema $\mathcal{S}$, which is denoted as $m = A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} ... \xrightarrow{R_{l+1}} A_{l+1}$ (i.e., $A_1 A_2 ... A_{l+1}$), where node type $A_l \in \mathcal{A}$ and link type $R_l \in \mathcal{R}$. Different meta-paths describe the semantic relations from different views.

Meta-graph

A meta-graph, $\mathcal{T} = (V_{\mathcal{T}}, E_{\mathcal{T}})$ can be seen as a directed acyclic graph (DAG) composed of multiple meta-path with common nodes, where $V_{\mathcal{T}}$ is the set of nodes, and $E_{\mathcal{T}}$ is the set of edges.

RGCN

Modeling Relational Data with Graph Convolutional Networks

Michael Schlichtkrull and others, “Modeling Relational Data with Graph Convolutional Networks” (Aldo Gangemi and others eds, Lecture Notes in Computer Science, Springer International Publishing 2018)

Introduction

R-GCN is one of the first attempts using graph convolutional networks on Heterogeneous graph. It solved the use of GCN to deal with the influence of different edge relationships on nodes in the graph structure, which is also a point ignored in GCN, without considering the relationship between nodes.

Model

R-GCN is implemnted based on a simple differentiable message-passing framework:

$$h_i^{(l+1)} = \sigma(\sum_{m \in \mathcal{M}_i} g_m(h_i^l, h_j^l))$$

- $h_i^l \in \mathbb{R}^{d^{(l)}}$ and $h_j^l \in \mathbb{R}^{d^{(l)}}$ are the hidden states of node v_i and v_j in the l-th layer
- $\mathcal{M}_i$ represents the set of incoming messages for node v_i
- $g_m(h_i^l, h_j^l) = Wh_j$ is typically chosen to be a (message-specific) neural network-like function or simply a linear transformation with the weight matrixs W.

Model – Forward

Single R-GCN layer

For an entity (node v_i)

$$h_i^{(l+1)} = \sigma\left(\sum_{r\in\mathcal{R}}\sum_{j\in\mathcal{N}_i^r}\frac{1}{c_{i,r}}W_r^{(l)}h_j^{(l)} + W_0^{(l)}h_i^{(l)}\right)$$

where $\mathcal{N}_i^r$ denotes the set of neighbor indices of node i under relation $r \in \mathcal{R}$, c is the problem-specific normalization contsant, and σ is the activation function.

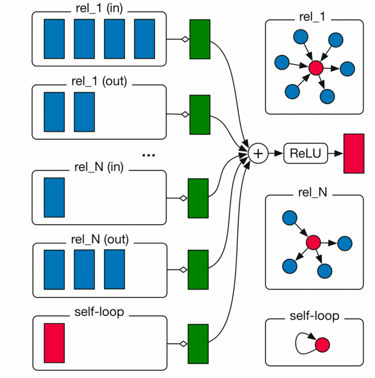


Figure: Figure excerpted from RGCN (). rel_1 (in) and rel_1 (out) denote the incoming and outgoing activations from the nodes connected by relation 1, respectively. Similar processes are implemented for other relations.

Optimizetion for large graph (TODO: need fig)

Why?

Overfitting on rare relations

How to solve?

- basis-decomposition: $W_r^{(l)} = \sum_{b=1}^{B} a_{rb}^{(l)} V_b^{(l)}$
- block-diagonal decomposition: $W_r^{(l)} = \bigoplus_{b=1}^{B} Q_{br}^{(l)}$

where Q_{br} is the diagonal matrix: diag($Q_{1r}^{(l)}, ..., Q_{Br}^{(l)}$) with $Q_{br}^{(l)} \in \mathbb{R}^{(d^{(l+1)}/B) \times (d^{(l)}/B)}$.

HetGNN

Heterogeneous Graph Neural Network

Zhang and others[3]

[3] Chuxu Zhang and others, “Heterogeneous Graph Neural Network” (KDD ’19, Association for Computing Machinery 25 July 2019) ⟨https://doi.org/10.1145/3292500.3330961⟩.

Introduction I

Problem

Designing a model F to learn embeddings $\mathcal{E} \in \mathbb{R}^{|\mathcal{V}| \times d} (d \ll |\mathcal{V}|)$ that is able to encode both heterogeneous structural closeness and unstructured contents.

Challenges

- Nodes in HG may not connect to all types of the neighbors. For exapmle, in *author-paper-venue* network in Figure 1, the *author-venue* do not connect directly, while they may still express strong correlations.
- A node in HG may include multiple unstructured heterogeneous contents, e.g., text, image, and attributes.
- Different types of the neighbors contribute differently to the target node embeddings in HG. For example, the *author* and *paper* should have different contribution to *venue* embeddings.

Model

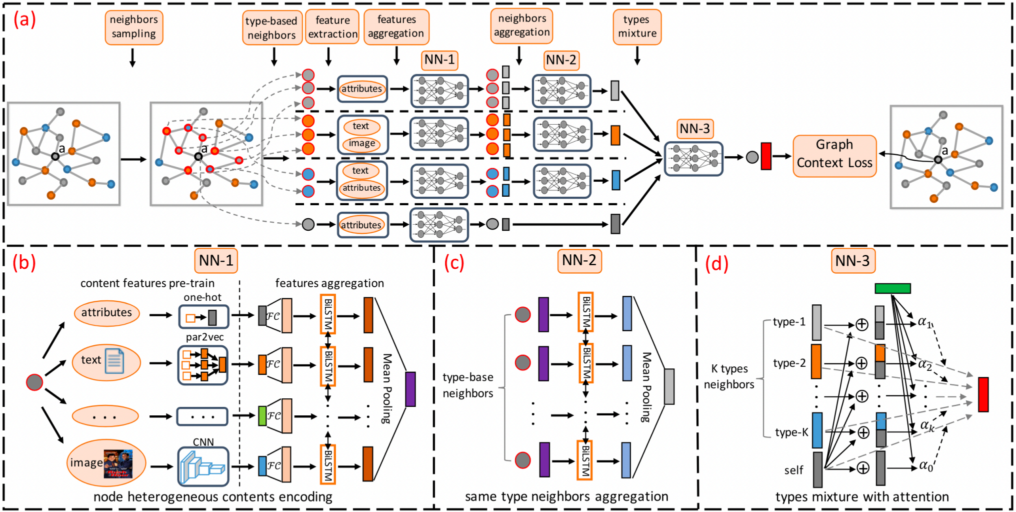


Figure: Figure excerpted from HetGNN.[5] (a) The overall architecture of HetGNN: it first samples fix sized heterogeneous neighbors for each node (node a in this case), next encodes each node content embedding via NN-1, then aggregates content embeddings of the sampled heterogeneous neighbors through NN-2 and NN-3, finally optimizes the model via a graph context loss; (b) NN-1: node heterogeneous contents encoder; (c) NN-2: type-based neighbors aggregator; (d) NN-3: heterogeneous types combination.

HAN

Heterogeneous Graph Attention Networks

Wang and others[6]

[6] Xiao Wang and others, “Heterogeneous Graph Attention Network” (WWW ’19, Association for Computing Machinery 13 May 2019) ⟨https://doi.org/10.1145/3308558.3313562⟩.

Introduction

- The first to propose learning both node-level attention and semantic-level attention in HG.
- The node-level attention is utilized to learn the importance between a node and its neighbors.
- The semantic-level attention is responsible for learning the importance between meta-path.

Model

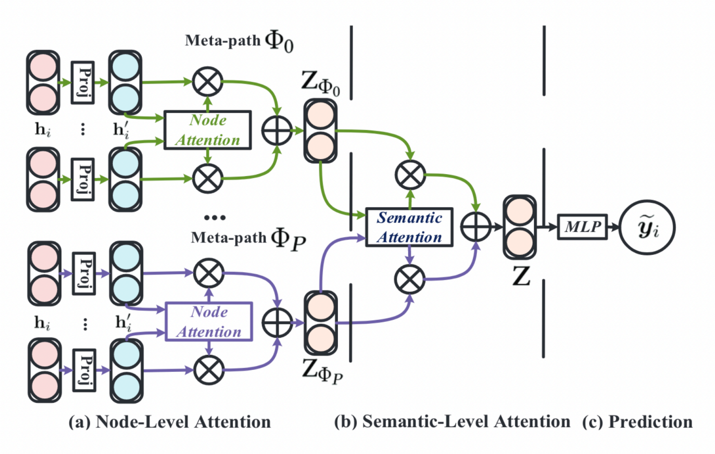


Figure: Figure excerpted from HAN.[8] The overall framework of the proposed HAN. (a) All types of nodes are projected into a unified feature space and the weight of meta-path based node pair can be learned via node-level attention. (b) Joint learning the weight of each meta-path and fuse the semantic-specific node embedding via semantic-level attention. (c) Calculate the loss and endto-end optimization for the proposed HAN.

Model

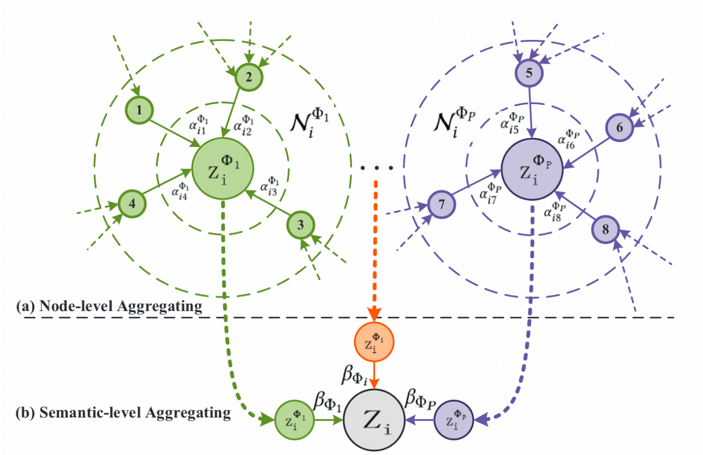


Figure: Figure excerpted from HAN.[10] Illustration of the aggregating process in both node-level and semantic-level.

[9] Wang and others (n 7).

[10] ibid.

HGT

Heterogeneous Graph Transformer

11

[11] Ziniu Hu and others, "Heterogeneous Graph Transformer" in *Proceedings of The Web Conference 2020* (Comment: Published on WWW 2020, Association for Computing Machinery 20 April 2020) ⟨https://doi.org/10.1145/3366423.3380027⟩.

Introduction

Heterogeneous Graph Transformer (HGT) proposes to learn the meta-relations for heterogeneous graph by parameterizing weight matrices for heterogeneous mutual attention and message passing.

Limitations when utilizing traditional methods on HG

- The construction of meta-path needs domain knowledge;
- The features are simply either shared for different types of nodes/edges or keep distinct for non-sharing weights for nodes/edges alone;
- The scalability is limited when modeling Web-scale (large) heterogeneous graph

Workflow

- Heterogeneous Mutual Attention, which is used to learn from the embeddings of source nodes and the target node, and output the edge-related attention matrices;
- Heterogeneous Message Passing, which is utilized to output the message of edges; (3)
- Target-Specific Aggregation, which is responsible for aggregating the neighbors' information.

Model

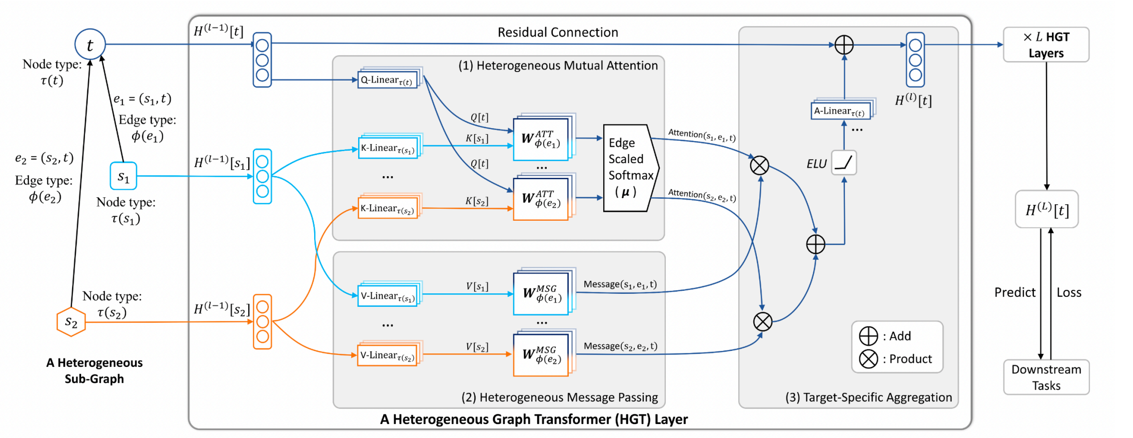


Figure: Figure excerpted from HGT.[13] The overall architecture of Heterogeneous Graph Transformer (HGT)

[12] Hu and others (n 11).

[13] ibid.

Heterogeneous Mutual Attention (TODO: in detail or ignore)

$$\begin{aligned}\textbf{Attention}_{HGT}(s,e,t) &= \underset{\forall s \in N(t)}{\text{Softmax}}\left(\mathop{\Big\|}_{i\in[1,h]} ATT\text{-}head^i(s,e,t)\right) \\ ATT\text{-}head^i(s,e,t) &= \left(K^i(s)\ W^{ATT}_{\phi(e)}\ Q^i(t)^T \right) \cdot \frac{\mu_{\langle \tau(s),\phi(e),\tau(t)\rangle}}{\sqrt{d}} \\ K^i(s) &= \text{K-Linear}^i_{\tau(s)}\left(H^{(l-1)}[s]\right) \\ Q^i(t) &= \text{Q-Linear}^i_{\tau(t)}\left(H^{(l-1)}[t]\right),\end{aligned} \tag{1}$$

Heterogeneous Message Passing (TODO: in detail or ignore)

$$\textbf{Message}_{HGT}(s, e, t) = \mathop{\Big\|}_{i \in [1,h]} MSG\text{-}head^{i}(s, e, t), \tag{2}$$

$$MSG\text{-}head^{i}(s, e, t) = \text{M-Linear}^{i}_{\tau(s)}\left(H^{(l-1)}[s]\right) W^{(MSG)}_{\tau(e)}. \tag{3}$$

Target-Specific Aggregation (TODO: in detail or ignore)

$$\widetilde{H}^{(l)}[t] = \bigoplus_{\forall s \in N(t)} \left(\mathbf{Attention}_{HGT}(s, e, t) \cdot \mathbf{Message}_{HGT}(s, e, t)\right), \tag{4}$$

$$H^{(l)}[t] = \text{A-Linear}_{\tau(t)}\left(\sigma\left(\widetilde{H}^{(l)}[t]\right)\right) + H^{(l-1)}[t]. \tag{5}$$

References I

- Hu Z and others, “Heterogeneous Graph Transformer” in *Proceedings of The Web Conference 2020* (Comment: Published on WWW 2020, Association for Computing Machinery 20 April 2020) ⟨https://doi.org/10.1145/3366423.3380027⟩.
- Schlichtkrull M and others, “Modeling Relational Data with Graph Convolutional Networks” (Gangemi A and others eds, Lecture Notes in Computer Science, Springer International Publishing 2018).
- Wang X and others, “Heterogeneous Graph Attention Network” (WWW ’19, Association for Computing Machinery 13 May 2019) ⟨https://doi.org/10.1145/3308558.3313562⟩.
- Wang X and others, “A Survey on Heterogeneous Graph Embedding: Methods, Techniques, Applications and Sources” [2022] IEEE Transactions on Big Data 1.
- Zhang C and others, “Heterogeneous Graph Neural Network” (KDD ’19, Association for Computing Machinery 25 July 2019) ⟨https://doi.org/10.1145/3292500.3330961⟩.