Heterogeneous Graph Attention Network

By Wang et al.

EGOR DMITRIEV, Utrecht Univeristy, The Netherlands

1 GOALS

- We first propose a novel heterogeneous graph neural network based on the hierarchical attention, including node-level and semantic-level attentions

2 PRELIMINARIES

- ...

3 CHALLENGES

- Real-world graph usually comes with multi-types of nodes and edges
- Different node types may have different attributes

4 PREVIOUS WORK / CITATIONS

- metapath2vec: randomwalk -> skipgram (context based) (negative sampling)
- hin2vec: max likelihood based on path count/probability approximation (but employs a negative sampling like approach)
  - Uses multiple prediction training tasks which learn the latent vectors of nodes and meta-paths simultaneously
- PME projects different types of node into the same relation space and conducts heterogeneous link prediction.
- **This Work:**
  - We introduce node-level attention can learn the importance of meta-path based neighbors for each node in a heterogeneous graph and aggregate the representation of these meaningful neighbors to form a node embedding.
  - To address the challenge of meta-path selection and semantic fusion in a heterogeneous graph, we propose a novel semantic-level attention to automatically learn the importance of different meta-paths and fuse them for the specific task.

5 DEFINITIONS

- **Semantic-level attention**: aims to learn the importance of each meta-path and assign proper weights to them.
- **Node-level attention**: aims to learn the importance of meta-path based neighbors and assign different attention values to them
- **Heterogeneous network**: $H = \{V, E, A, R, \phi, \psi\}$
  - $v_i \in V$: vertices, $e_{ij} \in E$: edges
  - $\phi(v_i)$: Node type, $\psi(e_{ij})$: Link type
  - A_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

Notation	Explanation
Φ	Meta-path
h	Initial node feature
M_{ϕ}	Type-specific transformation matrix
h'	Projected node feature
e_{ij}^{Φ}	Importance of meta-path based node pair (i, j)
a_{Φ}	Node-level attention vector for meta-path Φ
α_{ij}^{Φ}	Weight of meta-path based node pair (i, j)
$\mathcal{N}^{\Phi}$	Meta-path based neighbors
Z_{Φ}	Semantic-specific node embedding
q	Semantic-level attention vector
w_{Φ}	Importance of meta-path Φ
β_{Φ}	Weight of meta-path Φ
Z	The final embedding

6 OUTLINE / STRUCTURE

- Node-level attention: node-level attention can learn the importance of meta-path based neighbors for each node in a heterogeneous graph and aggregate the representation of these meaningful neighbors to form a node embedding.
- Project node features into a common space: $h_i' = \mathrm{M}_{\phi_i} \cdot h_i$

6.1 Node-level Embeddings

- Importance of meta-path based node pair (**node-level attention**): $e_{ij}^{\Phi} = att_{node}(h_i', h_j'; \Phi)$
  - att_{node} is MLP and is shared
  - e_{ij}^{Φ} is asymmetric; node level attention can preserve asymmetry
- Inject structural information via **masked attention** (by calculating neighbor weights):
  - $\alpha_{ij}^{\Phi} = \mathrm{softmax}_j\left(e_{ij}^{\Phi}\right) = \frac{\exp\left(\sigma\left(\mathrm{a}_{\Phi}^{\mathrm{T}} \cdot \left[\mathrm{h}_i' \| \mathrm{h}_j'\right]\right)\right)}{\sum_{k \in \mathcal{N}_i^{\Phi}} \exp\left(\sigma\left(\mathrm{a}_{\Phi}^{\mathrm{T}} \cdot \left[\mathrm{h}_i' \| \mathrm{h}_k'\right]\right)\right)}$
    - σ: activation
    - $||$: concatenation operation
    - Referred to as Weight Coefficient of Meta-path node pair
- Meta-path based embedding: aggregated based on neighbors and their weight coefs:
  - $\mathrm{z}_i^{\Phi} = \sigma\left(\sum_{j \in \mathcal{N}_i^{\Phi}} \alpha_{ij}^{\Phi} \cdot \mathrm{h}_j'\right)$

Close up of meta-path based embedding calculation

- Extend node-level attention to multi-head attention:
  - Since heterogeneous graph present the property of scale free, the variance of graph data is quite high
  - Process becomes more stable
  - Repeat node-level attention for K times
  - $\mathrm{z}_i^{\Phi} = ||_{k=1}^{K} \sigma\left(\sum_{j \in N_i^{\Phi}} \alpha_{ij}^{\Phi} \cdot \mathrm{h}_j'\right)$

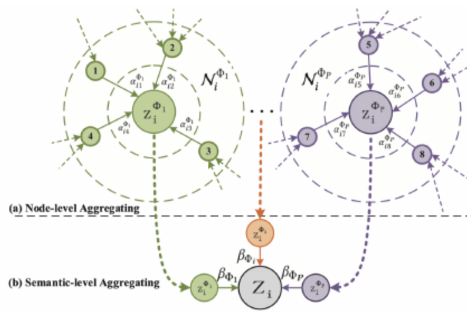

Fig. 1. Figure 3 Explanation of aggregating process in both node-level and semantic-level.

6.2 Semantic-level attention

- Calculate weight of each meta-path node pair: $(\beta_{\Phi_1}, \ldots, \beta_{\Phi_P}) = att_{\text{sem}}\ (Z_{\Phi_1}, \ldots, Z_{\Phi_P})$
- att_{sem}: DNN performing semantic level attention
  - Calculate importance of a meta path (by averaging the pair weights)
    * $w_{\Phi_p} = \frac{1}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} \mathrm{q}^{\mathrm{T}} \cdot \tanh\left(\mathrm{W} \cdot \mathrm{z}_i^{\Phi_p} + \mathrm{b}\right)$
  - Weight of meta-path is obtained by normalizing the importance of all meta-paths
    * $\beta_{\Phi} = \frac{\exp\left(w_{\Phi_p}\right)}{\sum_{p=1}^{P} \exp\left(w_p\right)}$
    * can be interpreted as the contribution of the meta-path Φ_p for specific task
- Final embedding:
  - $Z = \sum_{p=1}^{P} \beta_{\Phi_p} \cdot Z_{\Phi_p}$
- Loss function: minimizing Cross-Entropy over all labeled nodes

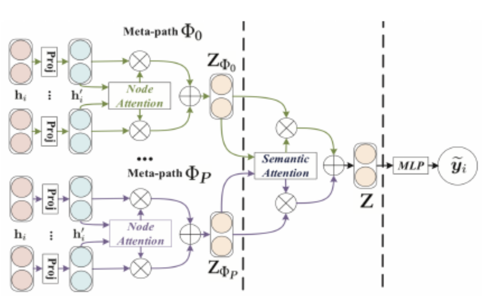

Fig. 2. Screenshot_20211102_210011

Algorithm 1: The overall process of HAN.

Input : The heterogeneous graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$,
The node feature $\{\mathbf{h}_i, \forall i \in \mathcal{V}\}$,
The meta-path set $\{\Phi_0, \Phi_1, \ldots, \Phi_P\}$.
The number of attention head K,

Output : The final embedding $\mathbf{Z}$,
The node-level attention weight α ,
The semantic-level attention weight β .

for $\Phi_i \in \{\Phi_0, \Phi_1, \ldots, \Phi_P\}$ **do**
  for $k = 1...K$ **do**
    Type-specific transformation $\mathbf{h}'_i \leftarrow \mathbf{M}_{\phi_i} \cdot \mathbf{h}_i$;
    for $i \in \mathcal{V}$ **do**
      Find the meta-path based neighbors $\mathcal{N}_i^{\Phi}$;
      for $j \in \mathcal{N}_i^{\Phi}$ **do**
        Calculate the weight coefficient α_{ij}^{Φ} ;
      end
      Calculate the semantic-specific node embedding
      $\mathbf{z}_i^{\Phi} \leftarrow \sigma\left(\sum_{j \in \mathcal{N}_i^{\Phi}} \alpha_{ij}^{\Phi} \cdot \mathbf{h}'_j\right)$;
    end
    Concatenate the learned embeddings from all attention head $\mathbf{z}_i^{\Phi} \leftarrow \Big\Vert_{k=1}^{K} \sigma\left(\sum_{j \in \mathcal{N}_i^{\Phi}} \alpha_{ij}^{\Phi} \cdot \mathbf{h}'_j\right)$;
  end
  Calculate the weight of meta-path β_{Φ_i} ;
  Fuse the semantic-specific embedding
  $\mathbf{Z} \leftarrow \sum_{i=1}^{P} \beta_{\Phi_i} \cdot \mathbf{Z}_{\Phi_i}$;
end
Calculate Cross-Entropy $L = -\sum_{l \in \mathcal{Y}_L} \mathrm{Y}_l \ln(\mathrm{C} \cdot \mathbf{Z}_l)$;
Back propagation and update parameters in HAN;
return Z, α, β.

Fig. 3. Overall algorithm

7 EVALUATION

- Does outperform other algs by a (substantial) margin.
- Uses: DBLP, ACM, IMDB
- Evaluates node classification task
- Evaluates for clustering task (apply K-Means afterwards)

8 CODE

- https://github.com/Jhy1993/HAN
  - Code is a little messy
  - a_{ij}^{Φ} : https://github.com/Jhy1993/HAN/blob/71bac29a07fb8fab908d50a806a7bc38aa6c6611/models/
  - w_{Φ}: https://github.com/Jhy1993/HAN/blob/71bac29a07fb8fab908d50a806a7bc38aa6c6611/models/
  - semantic level attention calculation ?
    - https://github.com/Jhy1993/HAN/blob/71bac29a07fb8fab908d50a806a7bc38aa6c6611/model

9 RESOURCES

- ...

10 DISCUSSION

- That algorithms seems so compute heavy

    - Especially that softmax (which is not adressed in analysis?)
- Query based attention / importance would be cool
    - But that is a transformer like thing
-