GRAPH ATTENTION NETWORKS

Dong Li

Jan 2 , 2020

outlines

1 **GAT**

2 **HAN**

3 **Conclusion**

GRAPH ATTENTION NETWORKS (GAT)

ICLR 2018

outlines

1 Introduction

2 GAT

3 Experiments

Introduction

CNN V.S. GNN

CNN

- tasks: image classification, semantic segmentation, machine translation, etc.
- Object: grid-like structure data

GNN

- tasks : 3D meshes, social networks, telecommunication networks, biological networks, etc.
- Object : irregular domain

Introduction

➢ GNN

- **What is GNN?**
  - a generalization of recursive neural networks
  - directly deal with a more general class of graphs (cyclic, directed and undirected graphs)
  - consist of an iterative process, propagates the node states until equilibrium
  - followed by a neural network, which produces an output for each node based on its state
- **convolutions to the graph domain**
  - spectral approaches:

    -- working with a spectral representation of the graphs
    -- GCN
  - non-spectral approaches

    -- defining convolutions directly on the graph
    -- MoNet， GraphSAGE

Introduction

Attention Mechanisms

- **Attention**
  - Self-attention
  - Intra-attention/soft-attention
- **Advantage**
  - they allow for dealing with variable sized inputs, focusing on the most relvant parts of the input to make decisions

Introduction

➢ GRAPH ATTENTION NETWORKS (GAT)

- a novel convolution-style graph neural network, leverages attention mechanism for the homogeneous graph which includes only one type of nodes or links.

- **Object**: graph-structured data
- **Target**: to address the shortcomings of prior methods based on graph convolutions or their approximations
- **Method**: masked self-attentional layers
- **Advantage**: By stacking layers in which nodes are able to attend over their neghborhood's feature. We enables specifying different weights to different nodes in a neighborhood, without requiring any kinds of costly matrix operation or depending on knowing the graph structure upfront.
- **Application**: applicable to inductive problem and transductive problems

GAT

➢ GRAPH ATTENTIONAL LAYER

- Input: $\mathbf{h} = \{\vec{h}_1, \vec{h}_2, \ldots, \vec{h}_N\}, \vec{h}_i \in \mathbb{R}^F$
- Output: $\mathbf{h}' = \{\vec{h}'_1, \vec{h}'_2, \ldots, \vec{h}'_N\}, \vec{h}'_i \in \mathbb{R}^{F'}$
- self-attention: $a : \mathbb{R}^{F'} \times \mathbb{R}^{F'} \rightarrow \mathbb{R}$
- attention coefficient

  $$e_{ij} = a(\mathbf{W}\vec{h}_i, \mathbf{W}\vec{h}_j)$$

  $$\mathbf{W} \in \mathbb{R}^{F' \times F}$$

- masked attention

  compute e_{ij} for nodes $j \in \mathcal{N}_i$.

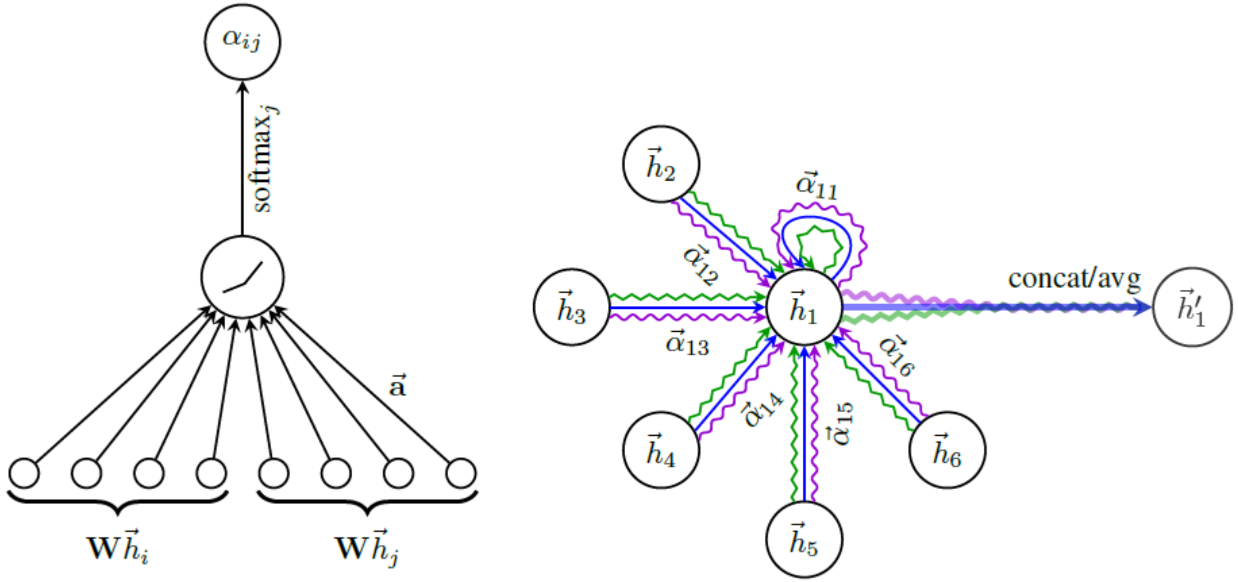


Figure 1: **Left:** The attention mechanism $a(\mathbf{W}\vec{h}_i, \mathbf{W}\vec{h}_j)$ employed by our model, parametrized by a weight vector $\vec{\mathbf{a}} \in \mathbb{R}^{2F'}$, applying a LeakyReLU activation. **Right:** An illustration of multi-head attention (with $K = 3$ heads) by node 1 on its neighborhood. Different arrow styles and colors denote independent attention computations. The aggregated features from each head are concatenated or averaged to obtain $\vec{h}'_1$.

GAT

➢ GRAPH ATTENTIONAL LAYER

- normalization:

$$\alpha_{ij} = \mathrm{softmax}_j(e_{ij}) = \frac{\exp(e_{ij})}{\sum_{k \in \mathcal{N}_i} \exp(e_{ik})}.$$

- a is a single-layer FNN

$$\alpha_{ij} = \frac{\exp\left(\mathrm{LeakyReLU}\left(\vec{\mathbf{a}}^T[\mathbf{W}\vec{h}_i \| \mathbf{W}\vec{h}_j]\right)\right)}{\sum_{k \in \mathcal{N}_i} \exp\left(\mathrm{LeakyReLU}\left(\vec{\mathbf{a}}^T[\mathbf{W}\vec{h}_i \| \mathbf{W}\vec{h}_k]\right)\right)}$$

$$\vec{\mathbf{a}} \in \mathbb{R}^{2F'}$$

$$\vec{h}'_i = \sigma\left(\sum_{j \in \mathcal{N}_i} \alpha_{ij} \mathbf{W}\vec{h}_j\right)$$

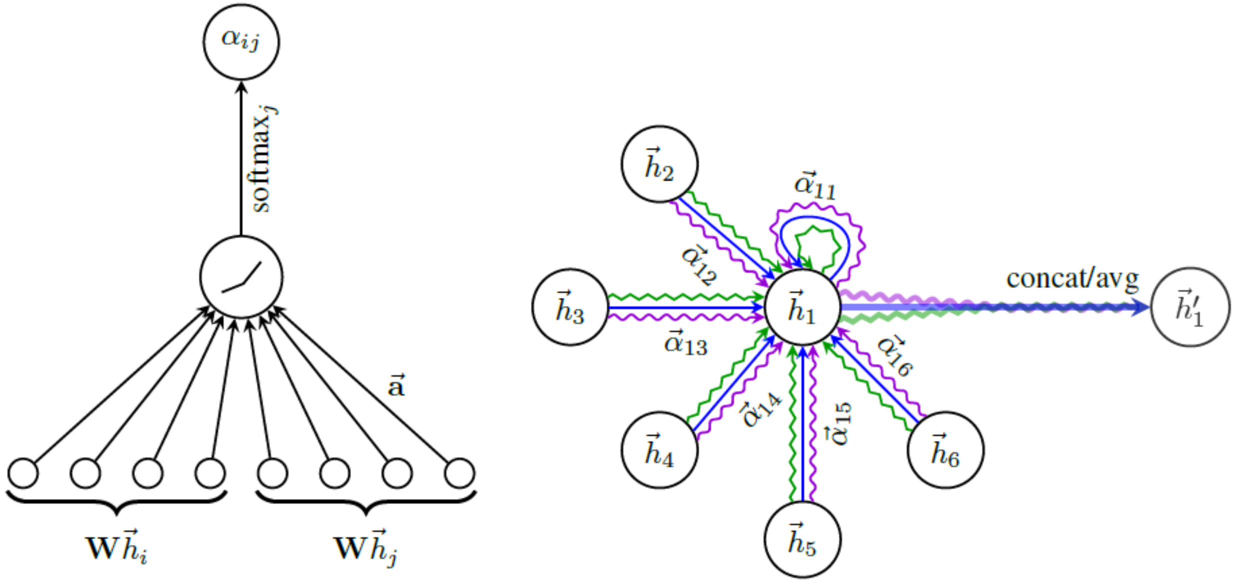


Figure 1: **Left:** The attention mechanism $a(\mathbf{W}\vec{h}_i, \mathbf{W}\vec{h}_j)$ employed by our model, parametrized by a weight vector $\vec{\mathbf{a}} \in \mathbb{R}^{2F'}$, applying a LeakyReLU activation. **Right:** An illustration of multi-head attention (with $K = 3$ heads) by node 1 on its neighborhood. Different arrow styles and colors denote independent attention computations. The aggregated features from each head are concatenated or averaged to obtain $\vec{h}'_1$.

GAT

- **GRAPH ATTENTIONAL LAYER**

  - Multi-head attention:

$$\vec{h}'_i = \Big\Vert_{k=1}^{K} \sigma\left(\sum_{j\in\mathcal{N}_i} \alpha_{ij}^{k} \mathbf{W}^k \vec{h}_j\right)$$

$$\vec{h}'_i = \sigma\left(\frac{1}{K}\sum_{k=1}^{K}\sum_{j\in\mathcal{N}_i} \alpha_{ij}^{k} \mathbf{W}^k \vec{h}_j\right)$$

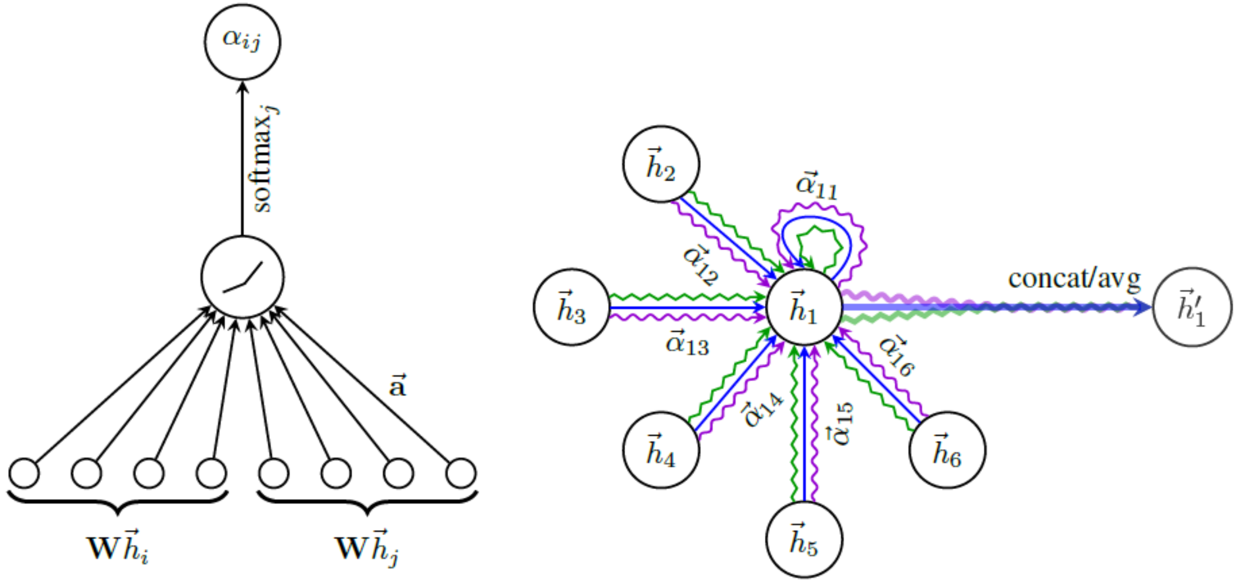


Figure 1: **Left:** The attention mechanism $a(\mathbf{W}\vec{h}_i, \mathbf{W}\vec{h}_j)$ employed by our model, parametrized by a weight vector $\vec{\mathbf{a}} \in \mathbb{R}^{2F'}$, applying a LeakyReLU activation. **Right:** An illustration of multi-head attention (with $K = 3$ heads) by node 1 on its neighborhood. Different arrow styles and colors denote independent attention computations. The aggregated features from each head are concatenated or averaged to obtain $\vec{h}'_1$.

GAT

➢ Analysis of GAT

- Computationally, it is highly efficient: the operation of the self-attentional layer can be parallelized across all edges, and the computation of output features can be parallelized across all nodes.
- GAT allows for (implicitly) assigning different importances to nodes of a same neighborhood, enabling a leap in model capacity.
- The attention mechanism is applied in a shared manner to all edges in the graph,whithout depending on upfront graph structure or all of its nodes.
- GAT works with the entirety of the neighborhood, and does not assume any ordering within it.
- GAT can be reformulated as a particular instance of MoNet.

Experiments

➢ Datasets

- Transductive Learning:

  -- three standard citation network benchmark datasets: Cora, Citeseer and Pubmed

- Inductive Learning:

  -- protein-protein interaction (PPI)

◻ Note

- Inductive Learning: Bayesian
- Transductive Learning: k近邻、SVM

				PPI
				Inductive
				44 (24 graphs)
				818716
				50
# Classes	7	6	3	121 (multilabel)
# Training Nodes	140	120	60	44906 (20 graphs)
# Validation Nodes	500	500	500	6514 (2 graphs)
# Test Nodes	1000	1000	1000	5524 (2 graphs)

Experiments

➢ Results

Transductive

Method	Cora	Citeseer	Pubmed
MLP	55.1%	46.5%	71.4%
ManiReg (Belkin et al., 2006)	59.5%	60.1%	70.7%
SemiEmb (Weston et al., 2012)	59.0%	59.6%	71.7%
LP (Zhu et al., 2003)	68.0%	45.3%	63.0%
DeepWalk (Perozzi et al., 2014)	67.2%	43.2%	65.3%
ICA (Lu & Getoor, 2003)	75.1%	69.1%	73.9%
Planetoid (Yang et al., 2016)	75.7%	64.7%	77.2%
Chebyshev (Defferrard et al., 2016)	81.2%	69.8%	74.4%
GCN (Kipf & Welling, 2017)	81.5%	70.3%	**79.0%**
MoNet (Monti et al., 2016)	$81.7 \pm 0.5\%$	—	$78.8 \pm 0.3\%$
GCN-64*	$81.4 \pm 0.5\%$	$70.9 \pm 0.5\%$	$\mathbf{79.0} \pm 0.3\%$
GAT (ours)	$\mathbf{83.0} \pm 0.7\%$	$\mathbf{72.5} \pm 0.7\%$	$\mathbf{79.0} \pm 0.3\%$

Inductive

Method	PPI
Random	0.396
MLP	0.422
GraphSAGE-GCN (Hamilton et al., 2017)	0.500
GraphSAGE-mean (Hamilton et al., 2017)	0.598
GraphSAGE-LSTM (Hamilton et al., 2017)	0.612
GraphSAGE-pool (Hamilton et al., 2017)	0.600
GraphSAGE*	0.768
Const-GAT (ours)	0.934 ± 0.006
GAT (ours)	$\mathbf{0.973} \pm 0.002$

Heterogeneous Graph Attention Network (HAN)

WWW 2019

outlines

1 **Introduction**

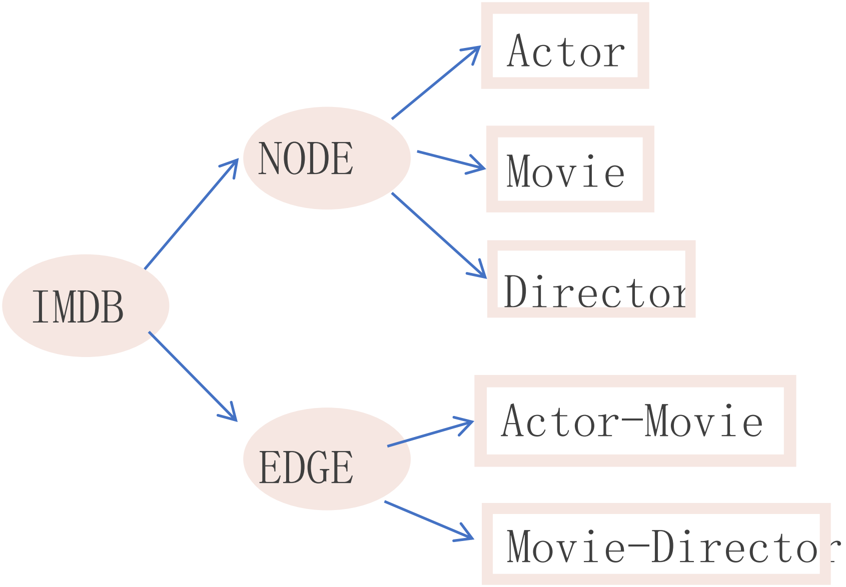
2 **HAN**

3 **Experiments**

Introduction

➢ Challenges

- Heterogeneous graph contains different types of nodes and links.
- Heterogeneous graph contains more comprehensive information and rich semantics.



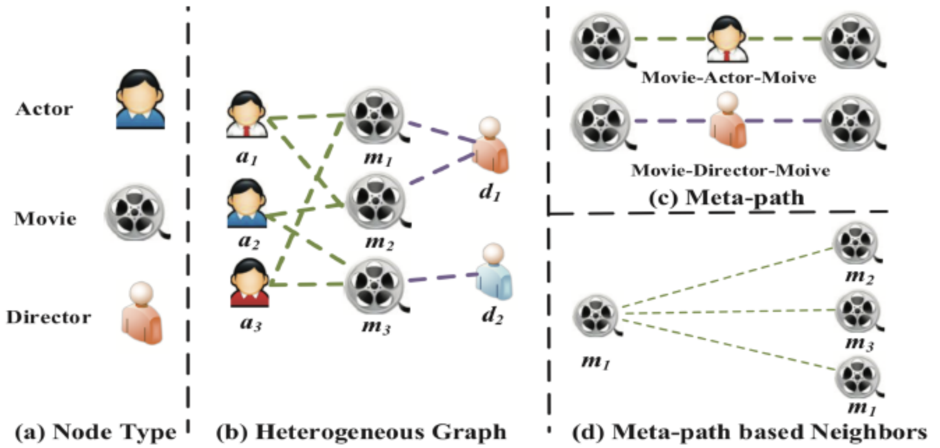


Figure 1: An illustrative example of a heterogenous graph (IMDB). (a) Three types of nodes (i.e., actor, movie, director). (b) A heterogenous graph IMDB consists three types of nodes and two types of connections. (c) Two meta-paths involved in IMDB (i.e., Moive-Actor-Moive and Movie-Director-Movie). (d) Moive m_1 and its meta-path based neighbors (i.e., m_1, m_2 and m_3).

Introduction

➢ Requirements to heterogeneous graph

- Heterogeneity of graph
  - ✓ How to handle such complex structural information and preserve the diverse feature information simultaneously ?
- Semantic-level attention
  - ✓ How to select the most meaningful **meta-paths** and fuse the semantic information for the specific task ?
- Node-level attention
  - ✓ How to distinguish the subtle difference of there neighbors and select some informative neighbors ?

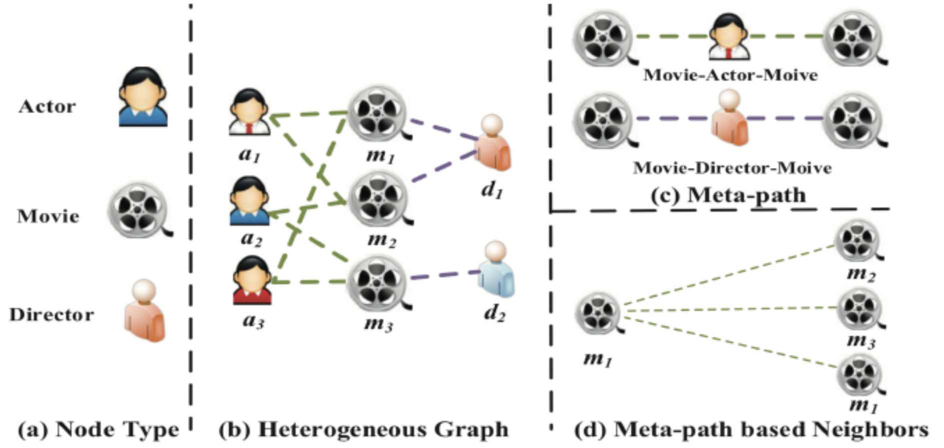


Figure 1: An illustrative example of a heterogenous graph (IMDB). (a) Three types of nodes (i.e., actor, movie, director). (b) A heterogenous graph IMDB consists three types of nodes and two types of connections. (c) Two meta-paths involved in IMDB (i.e., Moive-Actor-Moive and Movie-Director-Movie). (d) Moive m_1 and its meta-path based neighbors (i.e., m_1, m_2 and m_3).

HAN

➤ Framework of HAN

- **Hierarchical attention structure:**

node-level attention → semantic-level attention

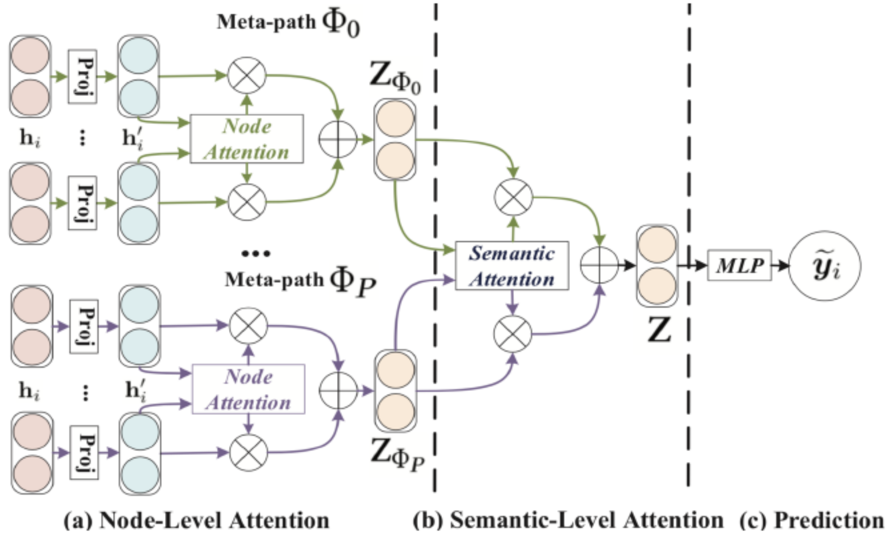


Figure 2: The overall framework of the proposed HAN. (a) All types of nodes are projected into a unified feature space and the weight of meta-path based node pair can be learned via node-level attention. (b) Joint learning the weight of each meta-path and fuse the semantic-specific node embedding via semantic-level attention. (c) Calculate the loss and end-to-end optimization for the proposed HAN.

HAN

➢ Definition of HAN

- Heterogeneous Graph

$\mathcal{G} = (\mathcal{V}, \mathcal{E})$

$\phi : \mathcal{V} \rightarrow \mathcal{A}$

$\psi : \mathcal{E} \rightarrow \mathcal{R}$

$|\mathcal{A}| + |\mathcal{R}| > 2$

- Meta-path (Φ)

$$A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \cdots \xrightarrow{R_l} A_{l+1}$$

$$A_1 A_2 \cdots A_{l+1}$$

- Meta-path based Neighbors
  - ✓ the set of nodes which connect with node i via meta-path Φ, including itself

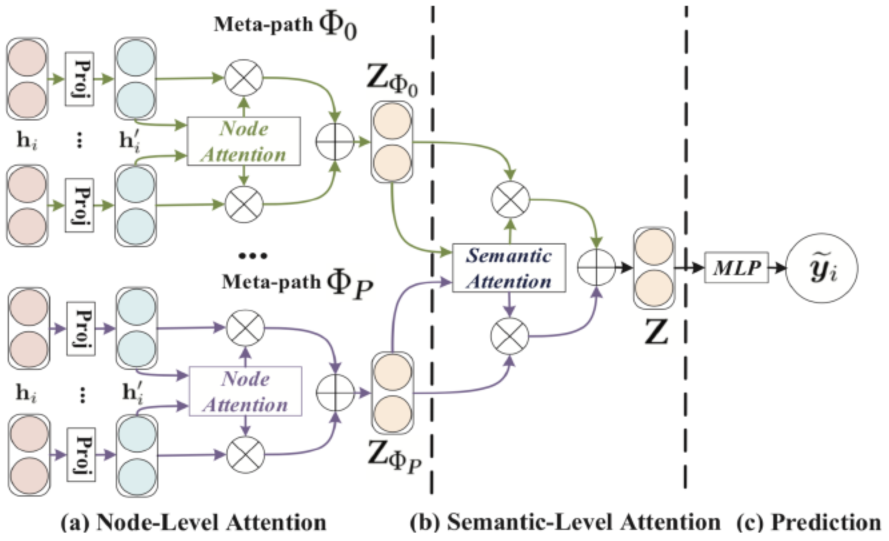


Figure 2: The overall framework of the proposed HAN. (a) All types of nodes are projected into a unified feature space and the weight of meta-path based node pair can be learned via node-level attention. (b) Joint learning the weight of each meta-path and fuse the semantic-specific node embedding via semantic-level attention. (c) Calculate the loss and end-to-end optimization for the proposed HAN.

HAN

➢ Framework of HAN

- **Node-level Attention**

$$\mathbf{h}'_i = \mathbf{M}_{\phi_i} \cdot \mathbf{h}_i.$$

$$e^{\Phi}_{ij} = att_{node}(\mathbf{h}'_i, \mathbf{h}'_j; \Phi).$$

$$\alpha^{\Phi}_{ij} = softmax_j(e^{\Phi}_{ij}) = \frac{\exp\left(\sigma(\mathbf{a}^{\mathrm{T}}_{\Phi} \cdot [\mathbf{h}'_i \| \mathbf{h}'_j])\right)}{\sum_{k \in \mathcal{N}^{\Phi}_i} \exp\left(\sigma(\mathbf{a}^{\mathrm{T}}_{\Phi} \cdot [\mathbf{h}'_i \| \mathbf{h}'_k])\right)},$$

$$\mathbf{z}^{\Phi}_i = \sigma\left(\sum_{j \in \mathcal{N}^{\Phi}_i} \alpha^{\Phi}_{ij} \cdot \mathbf{h}'_j\right)$$

$$\mathbf{z}^{\Phi}_i = \Big\Vert_{k=1}^{K} \sigma\left(\sum_{j \in \mathcal{N}^{\Phi}_i} \alpha^{\Phi}_{ij} \cdot \mathbf{h}'_j\right).$$

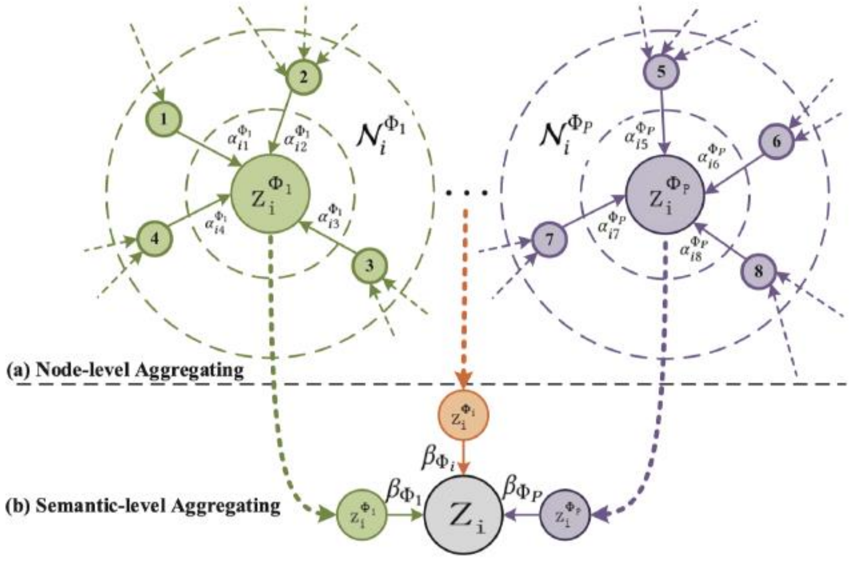


Figure 3: Explanation of aggregating process in both node-level and semantic-level.

HAN

➢ Framework of HAN

● **Semantic-level Attention**

$$(\beta_{\Phi_0}, \beta_{\Phi_1}, \ldots, \beta_{\Phi_P}) = att_{sem}(\mathbf{Z}_{\Phi_0}, \mathbf{Z}_{\Phi_1}, \ldots, \mathbf{Z}_{\Phi_P}).$$

$$w_{\Phi_i} = \frac{1}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} \mathbf{q}^{\mathrm{T}} \cdot \tanh(\mathbf{W} \cdot \mathbf{z}_i^{\Phi} + \mathbf{b}).$$

$$\beta_{\Phi_i} = \frac{\exp(w_{\Phi_i})}{\sum_{i=1}^{P} \exp(w_{\Phi_i})}$$

$$\mathbf{Z} = \sum_{i=1}^{P} \beta_{\Phi_i} \cdot \mathbf{Z}_{\Phi_i}$$

$$L = -\sum_{l \in \mathcal{Y}_L} \mathbf{Y}^l \ln(\mathbf{C} \cdot \mathbf{Z}^l),$$

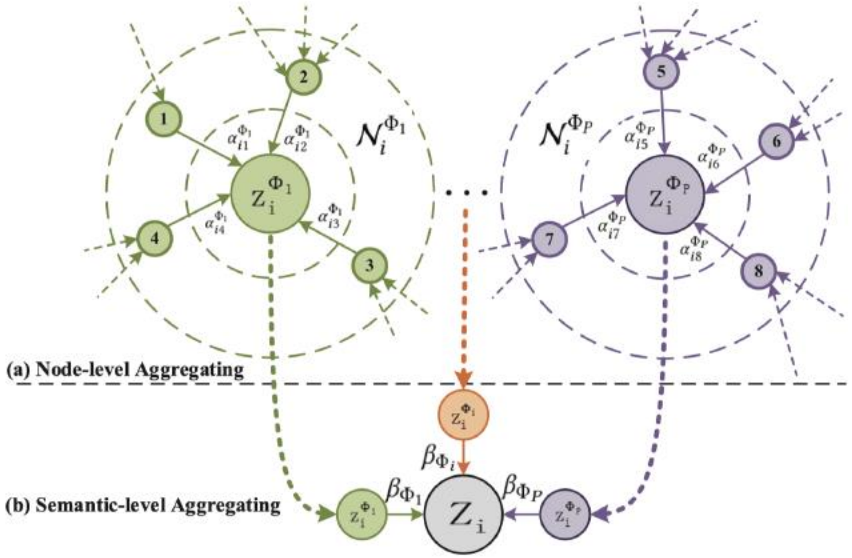


Figure 3: Explanation of aggregating process in both node-level and semantic-level.

HAN

➢ Analysis of HAN

- HAN deal with various types of nodes and relations and fuse rich semantics in a heterogeneous graph.
- HAN is highly efficient and can be easily parallelized.
- The hierarchical attention is shared for the whole heterogeneous graph which means the number of parameters is not dependent on the scale of a heterogeneous graph and can be used for the inductive problems. (Inductive means the model can generate node embeddings for previous unseen nodes or even unseen graph.)
- HAN has potentionally good interpretability for the learned node embedding.
- HAN is a big advantage for heterogeneous graph analysis.

Experiments

➢ Datasets

Table 2: Statistics of the datasets.

Dataset	Relations(A-B)	Number of A	Number of B	Number of A-B	Feature	Training	Validation	Test	Meta-paths
DBLP	Paper-Author	14328	4057	19645	334	800	400	2857	*APA*
	Paper-Conf	14328	20	14328					*APCPA*
	Paper-Term	14327	8789	88420					*APTPA*
IMDB	Movie-Actor	4780	5841	14340	1232	300	300	2687	*MAM*
	Movie-Director	4780	2269	4780					*MDM*
ACM	Paper-Author	3025	5835	9744	1830	600	300	2125	*PAP*
	Paper-Subject	3025	56	3025					*PSP*

Experiments

➢ Results

Table 3: Qantitative results (%) on the node classification task.

Datasets	Metrics	Training	DeepWalk	ESim	metapath2vec	HERec	GCN	GAT	HAN_{nd}	HAN_{sem}	HAN
ACM	Macro-F1	20%	77.25	77.32	65.09	66.17	86.81	86.23	88.15	89.04	**89.40**
		40%	80.47	80.12	69.93	70.89	87.68	87.04	88.41	89.41	**89.79**
		60%	82.55	82.44	71.47	72.38	88.10	87.56	87.91	**90.00**	89.51
		80%	84.17	83.00	73.81	73.92	88.29	87.33	88.48	90.17	**90.63**
	Micro-F1	20%	76.92	76.89	65.00	66.03	86.77	86.01	87.99	88.85	**89.22**
		40%	79.99	79.70	69.75	70.73	87.64	86.79	88.31	89.27	**89.64**
		60%	82.11	82.02	71.29	72.24	88.12	87.40	87.68	**89.85**	89.33
		80%	83.88	82.89	73.69	73.84	88.35	87.11	88.26	89.95	**90.54**
DBLP	Macro-F1	20%	77.43	91.64	90.16	91.68	90.79	90.97	91.17	92.03	**92.24**
		40%	81.02	92.04	90.82	92.16	91.48	91.20	91.46	92.08	**92.40**
		60%	83.67	92.44	91.32	92.80	91.89	90.80	91.78	92.38	**92.80**
		80%	84.81	92.53	91.89	92.34	92.38	91.73	91.80	92.53	**93.08**
	Micro-F1	20%	79.37	92.73	91.53	92.69	91.71	91.96	92.05	92.99	**93.11**
		40%	82.73	93.07	92.03	93.18	92.31	92.16	92.38	93.00	**93.30**
		60%	85.27	93.39	92.48	93.70	92.62	91.84	92.69	93.31	**93.70**
		80%	86.26	93.44	92.80	93.27	93.09	92.55	92.69	93.29	**93.99**
IMDB	Macro-F1	20%	40.72	32.10	41.16	41.65	45.73	49.44	49.78	**50.87**	50.00
		40%	45.19	31.94	44.22	43.86	48.01	50.64	52.11	50.85	**52.71**
		60%	48.13	31.68	45.11	46.27	49.15	51.90	51.73	52.09	**54.24**
		80%	50.35	32.06	45.15	47.64	51.81	52.99	52.66	51.60	**54.38**
	Micro-F1	20%	46.38	35.28	45.65	45.81	49.78	55.28	54.17	55.01	**55.73**
		40%	49.99	35.47	48.24	47.59	51.71	55.91	56.39	55.15	**57.97**
		60%	52.21	35.64	49.09	49.88	52.29	56.44	56.09	56.66	**58.32**
		80%	54.33	35.59	48.81	50.99	54.61	56.97	56.38	56.49	**58.51**

Experiments

➢ Results

Table 4: Qantitative results (%) on the node clustering task.

Datasets	Metrics	DeepWalk	ESim	metapath2vec	HERec	GCN	GAT	HAN_{nd}	HAN_{sem}	HAN
ACM	NMI	41.61	39.14	21.22	40.70	51.40	57.29	60.99	61.05	**61.56**
	ARI	35.10	34.32	21.00	37.13	53.01	60.43	61.48	59.45	**64.39**
DBLP	NMI	76.53	66.32	74.30	76.73	75.01	71.50	75.30	77.31	**79.12**
	ARI	81.35	68.31	78.50	80.98	80.49	77.26	81.46	83.46	**84.76**
IMDB	NMI	1.45	0.55	1.20	1.20	5.45	8.45	9.16	10.31	**10.87**
	ARI	2.15	0.10	1.70	1.65	4.40	7.46	7.98	9.51	**10.01**

Conclusion

Conclusion

- highly efficient and can be easily parallelized.
- good interpretability
- Capturing more comprehensive information and rich semantics
- OpenQA Research

TANK YOU

I am the master of my destiny…Ldruth 2020