The Name of the Heterogenous graph neural networks

ABSTRACT

Heterogeneous information network. but also demonstrate its . HAE's source code and all data used in the paper are publicly available[1].

CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

KEYWORDS

datasets, neural networks, gaze detection, text tagging

ACM Reference Format:

. 2018. The Name of the Heterogenous graph neural networks. In *Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY*. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

introduction...

2 THE PROPOSED MODEL

In this section, we propose a novel semi-supervised graph neural network for heterogeneous graph. Figure XXX presents the whole framework of model. First, we propose heterogeneous similarity attention to better integrate the similarity of nodes based on heterogeneous graph information with the weights of neighbor nodes. and aggregate them to get the semantic-specific node embedding. After that, ?model? fuses meta-paths through attention mechanism to embed heterogeneous information networks and get the optimal weighted combination of the semantic-specific node embedding for the specific task.

2.1 Heterogeneous Similarity Attention

The input to our layer is a set of node features, $h = \{\vec{h}_1, \vec{h}_2, \cdots, \vec{h}_N\}$, $\vec{h}_i \in \mathbb{R}^F$, where N is the number of nodes, and F is the number of features in each node.

Due to the heterogeneity of nodes, different types of nodes have different feature spaces. Therefore, for each type of nodes, we design the type-specific transformation matrix M_{β_t} to project the features of different types of nodes into the same feature space. The projection process can be shown as follows:

$$h_i'^{\beta_t} = M_{\beta_t} \cdot h_i \tag{1}$$

where β is the set of meta-paths and β_t is one of the meta-paths. $h_i^{\beta_t}$ represents the projected feature of node i obtained by transforming the original feature h_i under the guidance of the meta-path β_t.

After that, we leverage node similarity matrix to learn the weight among various kinds of nodes. The similarity of two nodes essentially reflects the distance between the two in a certain metric space. The values in the similarity matrix can be regarded as the weights between neighboring nodes, which not only better reflects the heterogeneity of the network, but also explains the importance of the neighboring nodes. Please note that the similarity of itself in the similarity matrix is the largest, which also accords with the actual situation in reality. The meta-path-based neighbors of node i include itself. The similarity of meta-path based node pair (i, j) can be formulated as follows:

$$S_{ij}^{\beta_t} = \frac{2 \times \{\rho_{\to j}^{\beta_t} : \rho_{i\to j}^{\beta_t} \in P_{\beta_t}\}}{|\{\rho_{i\to i}^{\beta_t} : \rho_{i\to i}^{\beta_t} \in P_{\beta_t}\} + \{\rho_{j\to j}^{\beta_t} : \rho_{j\to j}^{\beta_t} \in P_{\beta_t}\}|} \tag{2}$$

Where $S_{ij}^{\beta_t}$ is the importance weight between node i and its neighbor node j under the meta-path β_t. $\{\rho_{i\to j}^{\beta_t} : \rho_{i\to j}^{\beta_t} \in P_{\beta_t}\}$ is the number of connected paths between node i and node j in the commuting matrix P_{β_t}. Similarly, $\{\rho_{i\to i}^{\beta_t} : \rho_{i\to i}^{\beta_t} \in P_{\beta_t}\}$ and $\{\rho_{j\to j}^{\beta_t} : \rho_{j\to j}^{\beta_t} \in P_{\beta_t}\}$. Neighbor node similarity calculation is defined in two parts: (1) the semantic overlap in the numerator, which is defined by the number of meta-paths between node i and node j ; and (2) the semantic broadness in the denominator, which is defined by the number of total meta-paths between themselves. Therefore, we can give a SimPath distance with weights for any two connected node.

Then, the meta-path based embedding of node i can be aggregated by the neighbor's projected features with the corresponding coefficients as follows:

$$z_i^{\beta} = \sigma(\sum_{j \in N_i^{\beta}} S_{ij}^{\beta} \cdot h_j'^{\beta}) \tag{3}$$

where z_i^{β} is the learned embedding of node i for the meta-path β. Every node embedding is aggregated by its neighors. Since the attention weight S_{ij}^{β} is generated for single meta-path, it is semantic-specific and able to caputre one kind of semantic information.

Since heterogeneous graph present the property of scale free, the variance of graph data is quite high. To tackle the above challenge, we extend heterogeneous similarity attention to multihead attention so that the training process is more stable. Specially, if we perform multi-head attention on the final (prediction) layer of the network, we employ averaging, and delay applying the final nonlinearity (usually a softmax or logistic sigmoid for classification

[1]https://shengyp.github.io/hetero_graph/

Woodstock '18, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

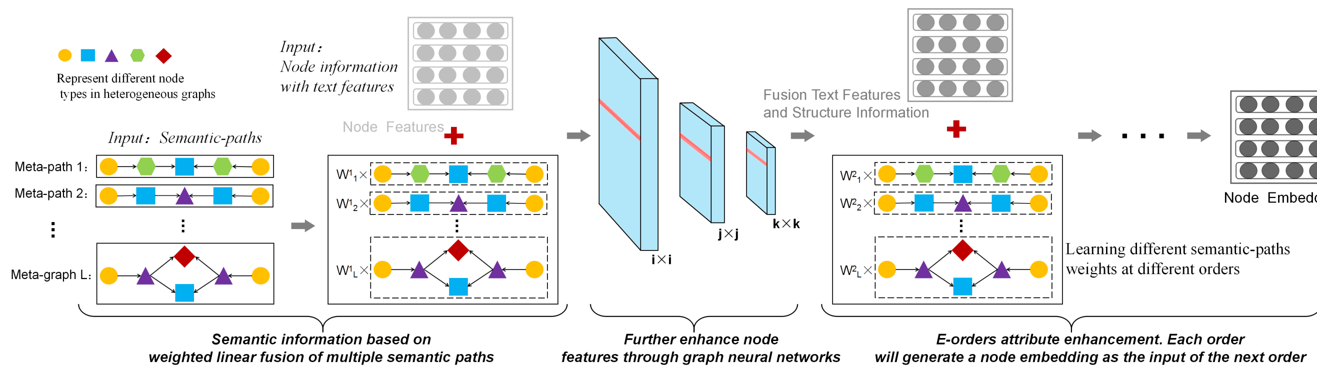


Figure 1: Overview of HAE. CESI first acquires side information of noun and relation phrases of Open KB triples. In the second step, it learns embeddings of these NPs and relation phrases while utilizing the side information obtained in previous step. In the third step, CESI performs clustering over the learned embeddings to canonicalize NP and relation phrases. Please see Section 3 for more details.

problems) until then:

$$z_i^{\beta} = \sigma\left(\frac{1}{K}\sum_{K=1}^{K}\sum_{j\in N_i^{\beta}} S_{ij}^{\beta}\cdot h_j'^{\beta}\right) \tag{4}$$

Given the meta-path set $\{\beta_1, \beta_2, \cdots, \beta_T\}$, after feeding node features into heterogeneous similarity attention, we can obtain T groups of semanticspecific node embeddings, denoted as $\{z_{\beta_1}, z_{\beta_2}, \cdots, z_{\beta_T}\}$.

3 EXPERIMENTS

3.1 Experimental Setup

Datasets. Our methods are evaluated on the following datasets.

- **DBLP**. This is a dataset from DBLP[2], which contains 14,328 papers (P), 4,057 authors (A), 20 conferences (C), 8,789 terms (T). We label each author's research area(Database, Data Mining, AI, Information Retrieval) according to the conferences they submitted. Author features are the elements of a bag-of-words represented of keywords.
- **IMDB**[3]. This is a subset of IMDB which contains 3,627 movies (M), 4,340 actors (A), 1,714 directors (D) and 11 content rating (R). The movies are divided into three classes (Action, Comedy, Drama) according to their genre. Movie features correspond to elements of a bag-of-words represented of plots.
- **HUAWEI**[4]. This is a dataset from HUAWEI, which contains 4,200 users (U), 6,131 applications (A) and 40 Types (T). We label each user according to their age, user features extraction is based on behavioral attributes and emotional information.
- **Douban**[5]. Here we extract a dataset of Douban which contains 13,367 users (U), 12,677 movies (M), 2,753 groups (G), 349 locations (L), 2,449 directors (D), and 6,311 actors (A). We set the type with the most views per user as the label.

Baseline algorithms. We compare HAE to a suite of classic and state-of-the-art baselines, as follows:

- Multi-layer perceptron (MLP) [7]: This is a feedforward neural network that has . Here, we adopt a four-layer connected neural network.
- Deepwalk [4]: A random walk based network embedding method for homogeneous network. Here we run DeepWalk on whole HIN and ignore the heterogeneity of nodes.
- Metapath2vec [1]: This is a network.
- GCN [3]: This is a network.
- GAT [9]: This is a semi-supervised neural network which considers the attention mechanism on the homogeneous graph.
- HAN [11]: This is a network.
- GTN []: This is a.
- **Ours**$_{SubInput}$. This is the variant of our method. The collection $\{(x_{hypo}, y_{hyper}), (x_{hypo}, d_y)$. We use the sub-script *SubInput* to denote this setting.

Metrics. Each author must be defined separately for accurate metadata identification.

Implementation details. We implement HAN by using Tensorflow[6]. The models are trained on NVIDIA Tesla P100. We randomly choose 80% of samples as the training data, 10% as validation and 10% for testing on each dataset. We initialize the hyper-parameters for the baselines by following the corresponding paper and carefully tune them to ensure that they achieve the optimal performance.

A more comprehensive description of experimental settings can be found in the supplementary material[7].

[2]https://dblp.uni-trier.de
[3]https://www.imdb.com
[4]https://www.imdb.com
[5]https://movie.douban.com
[6]https://www.tensorflow.org
[7]The supplemental material is available at https://shengyp.github.io/hetero_graph/supplemental.pdf

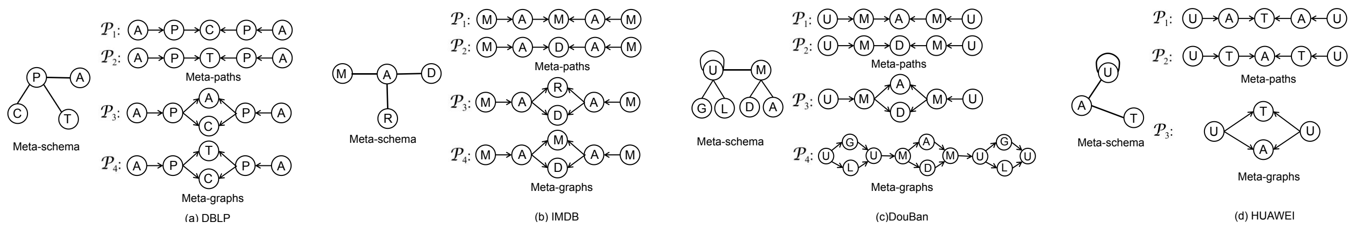


Figure 2: The meta-paths and meta-graphs of each dataset used in the experiments

3.2 Performance on node classification and clustering tasks

As for our GraphHeat model, it outperforms all baseline methods, achieving state-of-the-art results on all the four datasets.

The node classification results of different methods over the DBLP dataset are shown in Figure 1.

The clustering results of different methods over the DBLP dataset are shown in Figure 2.

3.3 Influence of hyper-parameter d and E

Influence of Hyper-parameter

3.4 Case study

We conduct a case study to demonstrate the effectiveness of learned weight information.

4 RELATED WORKS

4.1 Network representation embedding

Here we give a brief introduction to existing network representation learning (NRL) methods [12], this method is proposed to embed net-work into a low dimensional space while preserving the network structure and property so that the learned embeddings can be applied to the downstream network tasks. DeepWalk [4] employs Skip-gram model, which is originally used in word representation learning, on random walks for NRL. node2vec [2] further generalizes DeepWalk with Breadth First Search (BFS) and Depth-First Search (DFS) on random walks. LINE models [8] first-order and second-order proximities between vertices for learning large-scale network embeddings. M-NMF [10] uses matrix decomposition to learn network structure characteristics.

4.2 Heterogeneous graph embedding

However, all these algorithms are proposed for the homogeneous graphs [5**?**], heterogeneous graph embedding mainly focuses on preserving the meta-path based structural information. Metapath2vec [1] designs a meta-path based random walk and utilizes skip-gram to perform heterogeneous graph embedding. However, metapath2vec can only utilize one meta-path and may ignore some useful information. [13] proposes a embedding model metagraph2vec, where both the structures and semantics are maximally preserved for malware detection. [6] proposes meta-graph-based network embedding models, which simultaneously considers the hidden relations of all meta information of a meta-graph.

5 CONCLUSION

In this paper, we proposed HIV, a novel method for canonicalizing Open KBs using learned embeddings and side information. CESI solves a joint objective to learn noun and relation phrase embeddings, while utilizing relevant side information in a principled manner. These learned embeddings are then clustered together to obtain canonicalized noun and relation phrase clusters. In this paper, we also propose ReVerb45K, a new and larger dataset for Open KB canonicalization. Through extensive experiments on this and other real-world datasets, we demonstrate HIV's effectiveness over state-of-the-art baselines.

REFERENCES

[1] Yuxiao Dong, Nitesh V Chawla, and Ananthram Swami. 2017. metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 135–144.

[2] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 855–864.

[3] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *https://arxiv.org/abs/1609.02907* (2016).

[4] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 701–710.

[5] Richard P Stanley. 2017. Linear homogeneous diophantine equations and magic labelings of graphs. *Selected Works of Richard P. Stanley* 25 (2017), 81.

[6] Lichao Sun, Lifang He, Zhipeng Huang, Bokai Cao, Congying Xia, Xiaokai Wei, and S Yu Philip. 2018. Joint embedding of meta-path and meta-graph for heterogeneous information networks. In *2018 IEEE International Conference on Big Knowledge*. IEEE, 131–138.

[7] Jiexiong Tang, Chenwei Deng, and Guang-Bin Huang. 2015. Extreme learning machine for multilayer perceptron. *IEEE Transactions on Neural Networks and Learning Systems* 27, 4 (2015), 809–821.

[8] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. Line: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 1067–1077.

[9] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *https://arxiv.org/abs/1710.10903* (2017).

[10] Xiao Wang, Peng Cui, Jing Wang, Jian Pei, Wenwu Zhu, and Shiqiang Yang. 2017. Community preserving network embedding. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*.

[11] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. 2019. Heterogeneous Graph Attention Network. In *Proceedings of the 28th International Conference on World Wide Web*. ACM, 2022–2032.

[12] Cheng Yang, Zhiyuan Liu, Deli Zhao, Maosong Sun, and Edward Chang. 2015. Network representation learning with rich text information. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*.

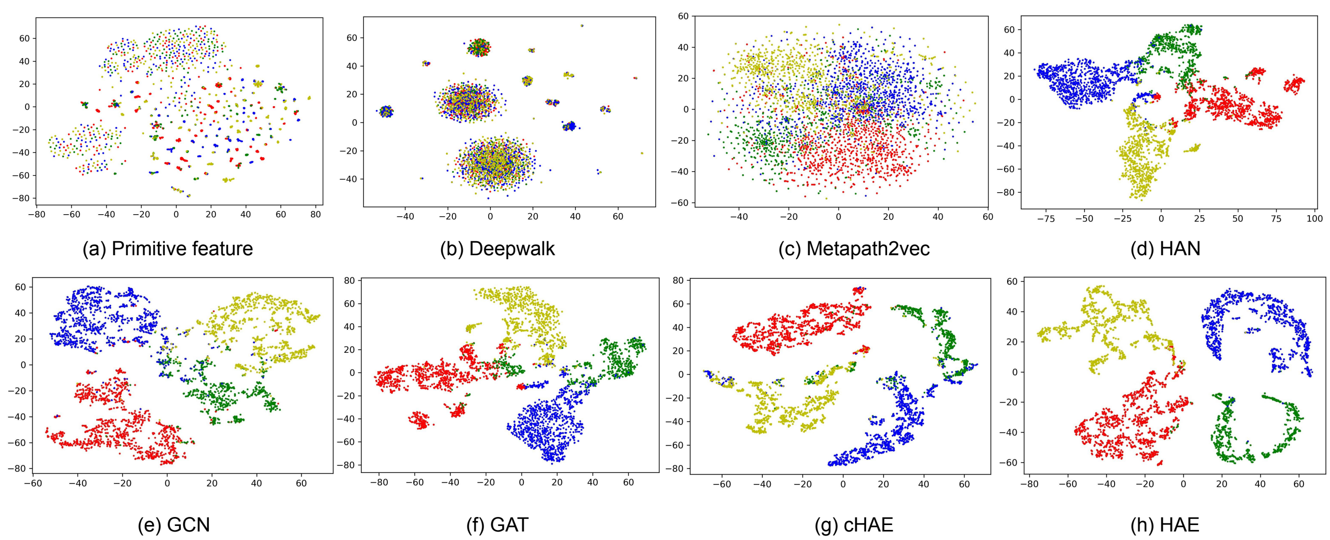
[13] Daokun Zhang, Jie Yin, Xingquan Zhu, and Chengqi Zhang. 2018. MetaGraph2Vec: complex semantic path augmented heterogeneous network embedding. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 196–208.

Table 1: The performance of different methds on the four datasets. The best performance in each column is boldfaced (the higher, the better). Improvements over the best baseline are shown in the last row.

Datasets	DBLP			IMDB			Douban			Huawei		
Measures	Micro	Macro	FMI	Micro	Macro	FMI	Micro	Macro	FMI	Micro	Macro	FMI
SVM	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.2	
MLP	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.2	
XGboost	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.2	1.2
Deepwalk	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.2	1.2
Metapath2vec	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.2	1.2
GCN	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.2	1.2
GAT	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.2	1.2
HAN	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.2	1.2
GTN	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.2	1.2
	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.2	1.23
	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.2	1.23
Ours	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23
Improv.	4.84%	1.56%	1.56%	1.56%	1.56%	1.56%	1.56%	1.56%	1.56%	1.56%	1.56%	1.56%

Table 2: The performance of different methds on the four datasets. The best performance in each column is boldfaced (the higher, the better). Improvements over the best baseline are shown in the last row.

Datasets	DBLP			IMDB			Douban			Huawei		
Measures	Micro	Macro	FMI	Micro	Macro	FMI	Micro	Macro	FMI	Micro	Macro	FMI
SVM	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.2	
MLP	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.2	
XGboost	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.2	1.2
Deepwalk	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.2	1.2
Metapath2vec	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.2	1.2
GCN	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.2	1.2
GAT	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.2	1.2
HAN	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.2	1.2
GTN	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.2	1.2
	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.2	1.23
	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.2	1.23
Ours	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23
Improv.	4.84%	1.56%	1.56%	1.56%	1.56%	1.56%	1.56%	1.56%	1.56%	1.56%	1.56%	1.56%



(a) Primitive feature (b) Deepwalk (c) Metapath2vec (d) HAN

(e) GCN (f) GAT (g) cHAE (h) HAE

Figure 3: Visualization results on DBLP dataset. Each point indicates a author and its color indicates the research area.