# Higher-Order Heterogeneous Graph Convolutional Network Based on Meta-Paths

Wanting Zhao, Hao Xu, Wenzhuo Huang
*School of Computer Science and Technology*
*East China Normal University*
Shanghai, China
{wantingzhao, haoxu, elop}@stu.ecnu.edu.cn

Jinkui Xie✉
*School of Computer Science and Technology*
*East China Normal University*
Shanghai, China
jkxie@cs.ecnu.edu.cn

***Abstract*—Graph convolution networks are potent methods in graph representation learning. Meta-paths, which connect different types of nodes, are extensively used to represent various semantic meanings in heterogeneous graphs. Inspired by the above, we design a higher-order heterogeneous graph convolutional network based on meta-paths. It not only chooses a few meta-paths but also captures higher-order meta-paths with important higher-order relations (such as communal relation). Besides, it contains a calculation method of higher-order meta-path-based adjacency matrices and a novel heterogeneous graph convolution network for generating node embeddings. At every message passing step, it linearly aggregates information from higher-order meta-path-based neighbors. The computational complexity analysis shows that our proposed model is of high efficiency and applies to large-scale heterogeneous graphs. Our proposed model outperforms the state-of-the-art results in three real-world heterogeneous graphs: DBLP, IMDB, and Amazon Kindle Review. The classification experiments show that the calculation of higher-order meta-path-based adjacency matrices brings 2.23% average accuracy improvement in DBLP and IMDB.**

***Index Terms*—heterogeneous graph, network embedding, graph convolutional network**

## I. Introduction

Many real-world data are intuitively modeled as graphs. These graphs containing different types of nodes or edges are usually called heterogeneous graphs (or heterogeneous information networks [1]), such as E-commerce networks, bibliographic networks, and knowledge graphs. Heterogeneous graphs not only model complex network structures but also retain rich original information. Hence it has a significant effect on data mining and has been increasingly used. In heterogeneous graphs, meta-path [2] is defined as a composite relation between the starting node type and the ending node type. For example, meta-path $APC$ (Author-Paper-Conference) starting with the node type "Author" and ending with the node type "Conference, " which describes the relation of authors who publish their papers at conferences. Meta-path is universally used to extract structural features and collect information from neighbors based on it. In some ways, the choice of meta-paths largely determines the performance of data mining. Generally, there are two primary methods for selecting meta-paths:

- Method I: Selecting optimal meta-paths by domain experts with prior knowledge [3]–[5]. The selected meta-paths are strict relationship descriptors, which can not accommodate variance in relations, resulting in only a small part of paths matching the meta-paths [6].
- Method II: Using all meta-paths no longer than a predefined length $maxLen$ [2], [7], [8]. However, selecting massive meta-paths is hard to avoid due to the number of meta-paths with the exponential growth in $maxLen$.

So a new meta-path selection method is needed to avoid the disadvantages of the above methods. In GraRep [9] and MixHop [10], higher-order neighbors (such as 2-hop neighbors and 3-hop neighbors) have proven useful for graph analysis tasks. As for heterogeneous graphs, we noticed that higher-order meta-paths (see the definition of higher-order meta-paths in section III) contain special meanings. Especially 2-multiples meta-paths, which are palindrome, they represent communal relations: $APA$ represents Co-Author relation, $PTP$ represents two papers include a common terminology, $APCPA$ represents two authors publishing papers at the same conference, etc. If two nodes are connected by a higher-order meta-path, there is a higher-order relation between them.

The higher-order meta-path is proposed in this paper, which considers the higher-order relations (such as communal relation) and selects a few meta-paths. Higher-order meta-paths can be obtained through the following two steps. First, choosing the meta-paths no longer than $K$, $K = 2$ is generally sufficient. Second, expanding each meta-path to higher-order meta-paths. For instance, meta-path $APC$ can be expanded to higher-order meta-paths $APC$, $APCPA$, $APCPAPC$, etc. 0ur $K$ can be much shorter than the $maxLen$ in method II. It is because our second step expands meta-path to higher-order meta-paths. For example, to get the meta-path $APCPA$, which is important in DBLP, the method II's $maxLen$ needs to be $4$, while our method simply used $K = 2$ to get the meta-path $APC$, then expand $APC$ to 2-multiples meta-path $APCPA$.

Graph neural networks (GNNs) [11]–[15], especially graph convolution networks (GCNs) [11], [12] are powerful methods for graph representation learning. Their embedded results are useful in various graph analysis tasks, including classification, link prediction, and visualization. However, a great majority of the existing GNNs are designed for homogeneous graphs, which cannot learn a general embedding to mix information from different meta-paths.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

For heterogeneous graphs analysis tasks, we design a novel **H**igher-**O**rder **H**eterogeneous **G**raph **C**onvolutional **N**etwork based on meta-paths, called **HOHGCN**. At each message passing step, HOHGCN collects information from different higher-order meta-paths, then mixes the information linearly. Specifically, our major contributions are as follows:

- We propose higher-order meta-paths, which not only reduce the number of selected meta-paths but also capture important higher-order relations, such as communal relations. Benefitting from higher-order meta-paths, HOHGCN dramatically reduces training time without affecting experimental results.
- A calculation method of the adjacency matrices based on higher-order meta-paths is proposed. We first normalize relation-based adjacency matrices, and then multiply them according to higher-order meta-paths. It reduces the impact of nodes with a lot of neighbors and brings 1.72%, 2.75%, and 0.19% accuracy improvement in DBLP, IMDB, and Amazon Kindle Review experiments, respectively.
- Our HOHGCN mixes the information from different higher-order meta-paths. We designed experiments to evaluate its performance, proving that our model has 0.97% to 1.45% performance improvement compared with the best result of state-of-art baselines. HOHGCN is of high efficiency and is suitable for large-scale heterogeneous graphs. Meanwhile, it is able to learn embeddings for nodes that never appear while training, which can be applied to dynamic graphs easily.

## II. Related Work

**Graph neural network.** In recent years, graph neural networks [11]–[13], [15] that model graph-structured data as neural networks are popular topics. They aggregate information from neighbors to generate node embeddings. For example, in GCN [12], a normalized adjacency matrix is used to collect information from neighbors, then a convolutional aggregator is used to learn embeddings as follows:

$$\hat{A} = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}} \tag{1}$$

$$H^{(l+1)} = \sigma\left(\hat{A}H^{(l)}W^{(l)}\right) \tag{2}$$

where $\tilde{A}$ denotes an adjacency matrix with self-connections, $\tilde{D}$ is a diagonal matrix where $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$, $H^{(l)}$ denotes the output of layer $l$, $W^{(l)}$ denotes a trainable weight matrix, and $\sigma$ is an activation function. In GraphSAGE [11], mean aggregator, pooling aggregator, and LSTM [16] aggregator are used to collect neighbors' information. MixHop [10] mixes features of neighbors at various distances to obtain neighbor mixing relationships. GAT [13] learns the influence weights of each neighbor via self-attention mechanism to generate node embeddings. HAN [3] extends GAT to heterogeneous graphs with hierarchical attention, in which the significance of different nodes and different meta-paths are considered. HetGNN [17] first samples heterogeneous neighbor nodes through a random walk, second group neighbor nodes by their types, then aggregates the information of nodes for every group, and finally aggregates the information of different groups into final embeddings. The final embeddings can be used for node classification, clustering, link prediction, and recommendation.

**Network Embedding.** It aims to learn a low dimensional latent representation of nodes from graph structure. Inspired by word2vec [18], in DeepWalk [19] and node2vec [20], random walks are used to generate node sequences (analogy to sentences), then SkipGram is used to learn node embeddings from node context. Similar to DeepWalk, LINE [21] learns node embeddings from the similarity of direct neighbors and the similarity of 2-hop neighbors. Besides, there are many other approaches, such as the self-encoded based methods [22], [23], matrix factorization based methods [9], [24], and methods based on adversarial generation networks [25]–[28]. The above methods are designed for homogeneous graphs. Regarding model heterogeneous graphs, metapath2vec [29], which random walks based on meta-paths is proposed. Moreover, HIN2vec [7] learns embeddings of nodes and meta-paths from relationships among nodes.

## III. Preliminaries and Problem Definition

Before introducing our model, some necessary basic definitions should be introduced firstly.

**Definition 1.** Directed Heterogeneous Graph [30]. A directed heterogeneous graph is defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{T}, \mathcal{R})$ with a node type mapping function $\mathcal{V} \to \mathcal{T}$ and an edge type mapping function $\mathcal{E} \to \mathcal{R}$, where $\mathcal{V}$ denotes a set of nodes, $\mathcal{E}$ denotes a set of directed edge, $\mathcal{T}$ denotes a set of node types, $\mathcal{R}$ is a set of edge types (edge types are called relations in this paper), and $|\mathcal{T}| + |\mathcal{R}| > 2$.

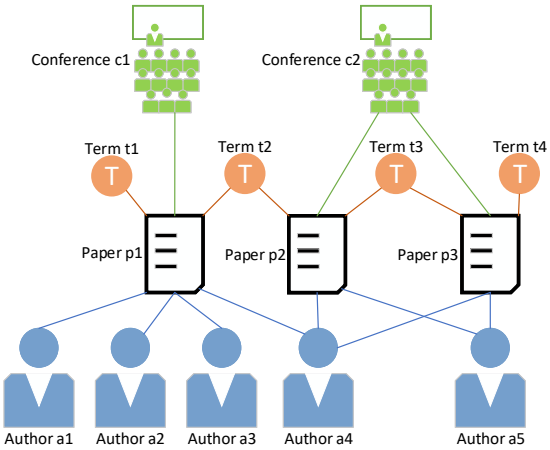



Fig. 1: Example of a heterogeneous graph.

**Example.** We model DBLP (a bibliography dataset) as a heterogeneous graph in Fig. 1. It includes multiple node types, such as Authors ($A$), Papers ($P$), Conferences ($C$), and Terms ($T$). It also includes multiple edge types. For instance, $A \to P$ represents the relation of authors to papers, $P \to C$ represents the relation of papers to conferences, $P \to T$ represents the relation of papers to their terms. Besides, $P \to A$, $C \to P$, and

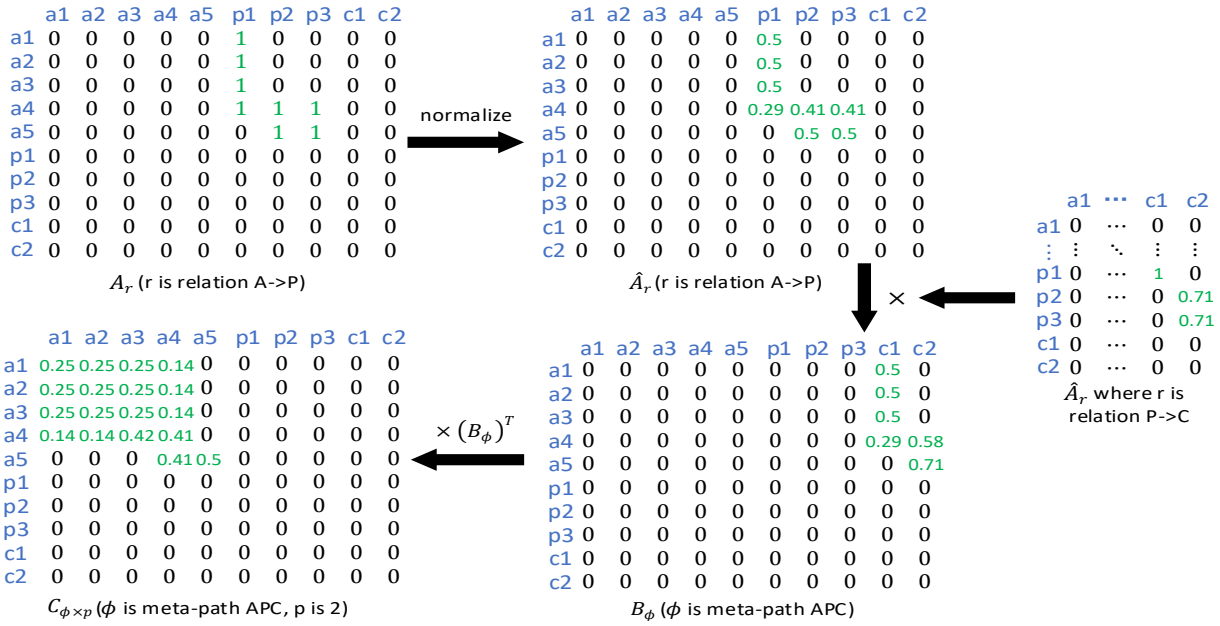



Fig. 2: Example of calculating higher-order meta-path-based directed adjacency matrix $\hat{C}_{\phi \times p}$ (using data in Fig. 1), where $p = 2$, $\phi$ represents meta-path $APC$ ( $Author \rightarrow Paper \rightarrow Conference$). $A_r$ is the original binary directed adjacency matrix of relation $r$. $\hat{A}_r$ denotes a normalized relation-based adjacency matrix. $\hat{B}_\phi$ denotes the meta-path-based adjacency matrix of meta-path $\phi$.

$T \rightarrow P$ represents the reverse relations of $A \rightarrow P$, $P \rightarrow C$, and $P \rightarrow T$, respectively.

**Definition 2.** Meta-Path [2]. A meta-path $\phi$ with a length of $l$ is usually represented as $t_1 \xrightarrow{r_1} t_2 \xrightarrow{r_2} \cdots \xrightarrow{r_l} t_{l+1}$, where $t_1, t_2, \cdots, t_{l+1}$ denote node types, $r_1, r_2, \cdots, r_l$ denote relations (different types of edge). It can be abbreviated as $t_1 t_2 \cdots t_{l+1}$ or $r_1 \circ r_2 \circ \cdots \circ r_l$, where $\phi = r_1 \circ r_2 \circ \cdots \circ r_l$ denotes a composite relation of node types $t_1$ to $t_{l+1}$, $\circ$ is a composition operator.

**Definition 3.** Higher-Order Meta-Path. Given a meta-path $\phi$ and a multiplier $p$, the higher-order meta-path $\phi \times p$ is defined as $p$-multiples of $\phi$. Besides, the length of $\phi \times p$ is $p$ times the length of $\phi$. For example, if $\phi$ is $APC$, $\phi \times 1$ is $APC$, $\phi \times 2$ is $APCPA$, $\phi \times 3$ is $APCPAPC$, and $\phi \times 4$ is $APCPAPCPA$. Given a higher-order meta-path $\phi \times p$, if nodes $u$ and $v$ are connected by $\phi \times p$, $v$ can be seen as a higher-order meta-path-based neighbor of $u$.

**Definition 4.** Relation-based Directed Adjacency Matrix. Given a relation $r$, the relation-based adjacency matrix $A_r$ is a sparse matrix. Only when there is a $r$ type edge connects node $u$ to node $v$, $A_{r,u,v} = 1$.

**Definition 5.** Meta-Path-Based Directed Adjacency Matrix. Given a meta-path $\phi = r_1 \circ r_2 \circ \cdots \circ r_l$, the meta-path-based directed adjacency matrix $\hat{B}_\Phi$ is defined as the product of $\hat{A}_{r_1} \hat{A}_{r_2} \cdots \hat{A}_{r_l}$, where $\hat{A}_{r_1}, \hat{A}_{r_2}, \cdots, \hat{A}_{r_l}$ are relation-based directed adjacency matrices of relations $r_1, r_2, \cdots, r_l$, respectively.

**Definition 6.** Higher-Order Meta-Path-Based Directed Adjacency Matrix. Given a meta-path $\phi$ and a multiplier $p$, the higher-order meta-path-based adjacency matrix $\hat{C}_{\phi \times p}$ is defined as $p$-multiples of $\hat{B}_\phi$($\hat{B}_\phi$ is a meta-path-based Adjacency Matrix). For example, if meta-path $\phi = r_1 \circ r_2$, then $\hat{B}_\Phi = \hat{A}_{r_1} \hat{A}_{r_2}$ and $\hat{C}_{\phi \times 3} = \hat{B}_\phi \left(\hat{B}_\phi\right)^T \hat{B}_\phi = \hat{A}_{r_1} \hat{A}_{r_2} \left(\hat{A}_{r_2}\right)^T \left(\hat{A}_{r_1}\right)^T \hat{A}_{r_1} \hat{A}_{r_2}$.

**Definition 7.** Binary Higher-Order Meta-Path-Based Directed Adjacency Matrix. Given a meta-path $\phi$ and a multiplier $p$, the binary higher-order meta-path-based adjacency matrix $C_{\phi \times p}$ is defined as a sparse binary matrix, where only when $\phi \times p$ connects node $u$ to node $v$, $C_{\phi \times p, u, v} = 1$.

## IV. Our Proposed Model

We are interested in messaging based on higher-order meta-paths, where nodes collect information from their higher-order meta-path-based neighbors at every step. As is known to all, GCNs are powerful in information collection via neighbors. So we design our model based on GCNs. The model learns to generate representations from neighbor information. Firstly, the method of obtaining higher-order meta-path-based adjacency matrices is introduced. After that, our proposed HOHGCN is described, which learns representations from above adjacency matrices and nodes' features. Finally, the classification method using the output of HOHGCN is given.

### A. Higher-Order Meta-Path-Based Directed Adjacency Matrix

As shown in Fig. 1, if we directly utilize the binary higher-order meta-path-based adjacency matrix $C_{\phi \times p}$ to node classification, the nodes with a lot of neighbors may have too much influence. For instance, given a meta-path $\phi = AP$ and a multiplier $p = 2$, then higher-order meta-path $\phi \times p$ is

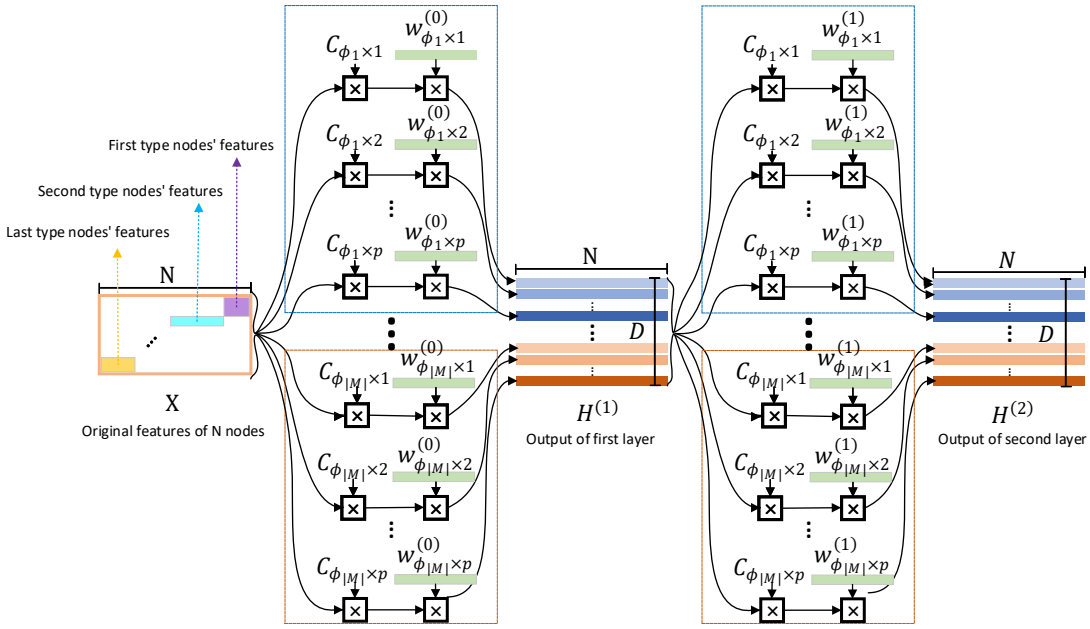



Fig. 3: Our proposed higher-order heterogeneous graph convolution network based on meta-paths (HOHGCN), in which $N$ denotes the number of nodes, $X$ denotes an original feature matrix that contains features from different types of nodes. $\hat{C}_{\phi_i \times p}$ denotes a higher-order meta-path-based adjacency matrix, where $\phi_j$ is the j$^{\text{th}}$ meta-path and $p$ denotes the multiplier. $W^{(i)}_{\phi \times p}$ denotes the trainable weight matrix of higher-order meta-path $\phi \times p$ in layer $i$, $D$ represents the embedding dimension, and $H^{(i)}$ represents the output of layer $i$.

$APA$, the binary adjacency matrix $C_{\phi \times p}$ represents co-author relation. It is already known that authors a1, a2, and a3 belong to category C1 while author a5 belongs to category C2, now we need to classify author a4. Based on the binary matrix $C_{\phi \times p}$, author a4 will be classified as category C1, because author a4 has only one co-author belongs to category C2, but three co-authors belong to category C1. It is unreasonable because author a4 published two papers with author a5 but only one paper with a1, a2, and a3.

So, it is necessary to compute higher-order meta-path-based adjacency matrix $\hat{C}_{\phi \times p}$, which is not a binary matrix. The calculation steps are shown in Fig. 2 and the heterogeneous graph is shown in Fig. 1. First step is to normalize relation-based adjacency matrix by:

$$\hat{A}_{r,i,j} = \begin{cases} \frac{A_{r,i,j}}{\sqrt{\left(\sum_{k=1}^{N} A_{r,i,k}\right)\left(\sum_{k=1}^{N} A_{r,k,j}\right)}} & , A_{r,i,j} > 0 \\ 0 & , A_{r,i,j} = 0 \end{cases} \tag{3}$$

where $A_r$ is the original binary directed adjacency matrix based on relation $r$. $N$ denotes the number of nodes, $\sum_{k=1}^{N} A_{r,i,k}$ and $\sum_{k=1}^{N} A_{r,k,j}$ are the out-degree of node $i$ and the in-degree of node $j$ in $A_r$. The result $\hat{A}_r$ is called the normalized relation-based (directed) adjacency matrix.

(3) is similar to (1) in GCN [12], but the difference is that the self-connections in (1) may not exist in the relation-based adjacency matrix. For example, given a relation $A \to P$ that represents the relation between authors and papers, an author can not be connected with itself in the relation-based adjacency matrix. Therefore, sometimes the degrees $D_{ii}$ is 0, that is why (1) can not be used in this paper. Our normalization reduces the impact of nodes with a lot of neighbors.

The meta-path-based directed adjacency matrix is calculated as follows:

$$\hat{B}_{\phi} = \prod_{i=1}^{|\phi|} \hat{A}_{r_i} \tag{4}$$

where $\phi = r_1 \circ r_2 \circ \cdots \circ r_{|\phi|}$ describes a meta-path, $|\phi|$ is the length of $\phi$, $r_i$ is a relation (a type of directed edge), $\hat{A}_{r_i}$ is the normalized adjacency matrix based on relation $r_i$. For example, meta-path $\phi$ is $APC$ which represents the path of $Author \to Paper \to Conference$, the length $|APC| = 2$, $\hat{A}_{r_1} = \hat{A}_{A \to P}$ and $\hat{A}_{r_2} = \hat{A}_{P \to C}$ are the relation-based adjacency matrices of relations $Author \to Paper$ and $Paper \to Conference$ respectively.

Higher-order meta-path-based directed adjacency matrix can be calculated as follows:

$$\hat{C}_{\phi \times p} = \left(\hat{B}_{\phi} \hat{B}_{\phi}^{T}\right)^{\lfloor \frac{p}{2} \rfloor} \left(\hat{B}_{\phi}\right)^{p \bmod 2} \tag{5}$$

where $\phi$ denotes a meta-path, $p$ denotes the multiplier, $\phi \times p$ denotes a higher-order meta-path, $\hat{C}_{\phi \times p}$ is the higher-order meta-path-based adjacency matrix of $\phi \times p$, $\hat{B}_{\phi}^{T}$ is the transposed matrix of $\hat{B}_{\phi}$, $\lfloor \frac{p}{2} \rfloor$denotes the floor of $\frac{p}{2}$, and $p \bmod 2$ is the remainder of $p$ divided by 2. For example, meta-path $\phi$ is $APC$ ($Author \to Paper \to Conference$), the multiplier $p = 2$, the 2-multiples of meta-path $\phi$ is $APCPA$, $\hat{C}_{\phi \times 2} = \hat{B}_{\phi} \hat{B}_{\phi}^{T} = \left(A_{A \to P} A_{P \to C}\right) \left(A_{A \to P} A_{P \to C}\right)^{T} = A_{A \to P} A_{P \to C} A_{C \to P} A_{P \to A}$

TABLE I: Statistics of the datasets.

| *Dataset* | *Node or Edge* | *Type* | *Number* | *Feature* | *Training* | *Validation* | *Test* |
|---|---|---|---|---|---|---|---|
| DBLP | Nodes | Author | 4,057 | 330 | 800 | 400 | 2,857 |
| | | Conferences | 20 | 4 | | | |
| | | Paper | 14,328 | 0 | | | |
| | | Term | 8,789 | 0 | | | |
| | Edges | Paper-Author | 19,645 | 0 | | | |
| | | Paper-Conferences | 14,328 | 0 | | | |
| | | Paper-Term | 88,420 | 0 | | | |
| IMDB | Nodes | Movie | 3,088 | 6,045 | 300 | 300 | 2,488 |
| | | Actor | 4,410 | 0 | | | |
| | | Director | 1,678 | 0 | | | |
| | Edges | Movie-Actor | 9,255 | 0 | | | |
| | | Movie-Director | 3,017 | 0 | | | |
| Amazon Kindle Review | Nodes | Review | 184,992 | 101 | 4,000 | 4,000 | 176,992 |
| | | Product | 46,934 | 0 | | | |
| | | User | 39,375 | 0 | | | |
| | Edges | Product-Review | 184,992 | 0 | | | |
| | | User-Review | 184,992 | 0 | | | |

### *B. Higher-Order Graph Convolution Layer Based on Meta-Paths*

As shown in Fig. 3, we design a multi-layer higher-order graph convolutional network based on meta-paths (HOHGCN), the layer-wise propagation rule is as follows:

$$H^{(i+1)} = \|_{\phi \in M} \|_{p=1}^{P} \sigma \left( \hat{C}_{\phi \times p} H^{(i)} W_{\phi \times p}^{(i)} \right) \tag{6}$$

where $\|$ denotes column-wise concatenation. $M$ is a set, which includes an empty path and all the meta-paths no longer than a predefined length $K$. $P$ indicates the maximum multiplier of higher-order meta-path. $\sigma(\cdot)$ is the ReLU activation function. $\hat{C}_{\phi \times p}$ is the higher-order meta-path-based adjacency matrix. $H^{(i)} \in \mathbb{R}^{N \times D}$ denotes the output of the $i^{th}$ layer. $W_{\phi \times p}^{(i)}$ represents a trainable weight matrix, and $H^{(0)} = X$ is the original features of nodes.

### *C. Node Classification*

For classification tasks, the final layer of our model outputs the probability distribution of predict labels:

$$\tilde{Y} = softmax \left( H^{(L)} W_o \right) \tag{7}$$

where $H^L$ denotes the output of layer $L$. $W_o \in \mathbb{R}^{N \times D}$ denotes a trainable weight matrix, $softmax$ is a normalized exponential function of a probability distribution. Our model parameters are learned through minimizing cross-entropy loss with a L2 regularization, where cross-entropy loss is calculated only on the nodes that belonging to a particular type with known labels, similar to [12].

### *D. Computational Complexity*

In the calculation of $\hat{C}_{\phi \times p} H^{(i)}$, there is no need to calculate $\hat{C}_{\phi \times p}$. We expand $\hat{C}_{\phi \times p} H^{(i)}$ to $\hat{A}_{r_1} \hat{A}_{r_2} \cdots H^{(i)}$and calculate $\hat{C}_{\phi \times p} H^{(i)}$ with right-to-left multiplication. For example, if meta-path $\phi$ is $APC$, $p = 2$, then $\hat{C}_{\phi \times p} H^{(i)} = \left( \hat{A}_{A \to P} \left( \hat{A}_{P \to C} \left( \hat{A}_{C \to P} \left( \hat{A}_{P \to A} H^{(i)} \right) \right) \right) \right)$. The adjacency matrices $\hat{A}_*$ is a sparse matrix with no more than $S$ non-zero entries. Our layer (as shown in (6)) takes $\mathcal{O}\left(|M| KPSD\right)$ computational time, where $|M|$ denotes the count of meta-paths, $K$ denotes the maximum length of meta-paths, $P$ denotes the maximum multiplier of higher-order meta-path, and $D$ is the embedding dimension of $H^{(i)}$.

In general, $|M|$, $P$, and $D$ are small numbers. In most experiments, $P = 2$, $K = 2$, and $D = 64$, therefore, the number of meta-paths $|M|$ is lower than $|R|^2 + |R| + 1$, where $|R|$ is the number of edge types. Our HOHGCN is of high efficiency and is appropriate for large-scale heterogeneous graphs easily.

## V. Experiments

### *A. Datasets*

We select three real heterogeneous graph datasets for evaluation and summarize their statistics in TABLE I.

- DBLP[1]. Similar to [3], we select a subgraph from DBLP dataset includes four node types, use bag-of-words feature vectors as author features represent their keywords, use one-hot vectors of research areas as conference features. Besides, we tag authors into four categories based on their research areas.
- IMDB[2]. We select the movies that belong to a single class of action, comedy, or drama from IMDB. It contains 3,088 movies, 4,410 actors, and 1,678 directors. Movie features are bag-of-words feature vectors represent their plots.
- Amazon Kindle Review[3]. We choose a subset from the Amazon Kindle Review dataset. The subset contains 184,992 reviews, 46,934 products, and 39,375 users. Reviews are divided into two classes, helpful and unhelpful, which are tagged by whether more than 50% of people think they are helpful. Review features are the one-hot vector of their scores and the bag-of-words feature vectors of their review texts.

[1]https://dblp.uni-trier.de/
[2]https://www.imdb.com/
[3]https://www.amazon.com/

TABLE II: Node classification accuracy in different number of training data per class.

| *Dataset* | *Training per class* | *DeepWalk* | *metapath2vec* | *GCN* | *GAT* | *HetGNN* | *HOHGCN-binary* | *HOHGCN* |
|---|---|---|---|---|---|---|---|---|
| DBLP | 20 | 80.29 | 90.93 | 91.39 | 91.28 | 90.34 | 92.54 | **93.98** |
| | 50 | 88.31 | 92.47 | 91.50 | 91.49 | 92.19 | 92.19 | **94.12** |
| | 100 | 89.71 | 92.44 | 92.76 | 90.79 | 93.66 | 93.14 | **94.54** |
| | 200 | 90.93 | 92.44 | 92.62 | 90.97 | 93.80 | 92.37 | **94.49** |
| IMDB | 20 | 38.14 | 39.49 | 48.51 | 49.32 | 44.21 | 48.79 | **50.56** |
| | 50 | 42.16 | 39.69 | 50.64 | 52.25 | 48.91 | 50.92 | **54.50** |
| | 100 | 45.02 | 42.38 | 53.82 | 56.31 | 53.38 | 54.14 | **57.03** |
| Amazon Kindle Review | 20 | 61.31 | 65.78 | 68.69 | 68.82 | 61.87 | 72.70 | **72.83** |
| | 200 | 73.69 | 72.66 | 79.29 | 78.42 | 74.29 | 80.08 | **80.10** |
| | 2000 | 77.30 | 78.31 | **84.43** | 81.25 | 80.56 | 82.09 | 82.52 |

### *B. Baselines*

We compare our method against homogeneous and heterogeneous methods. To validate our higher-order meta-path-based adjacency matrix, we also test a variant of HOHGCN.

- DeepWalk [19]: It is a homogeneous network embedding method via random walk. In this paper, we ignore the node types and edge types, i.e., test heterogeneous graphs as homogeneous graphs.
- metapath2vec [29]: An embedding method designed for heterogeneous graphs. It random walks based on meta-paths, then embeds heterogeneous graphs using skip-gram. We show the best result of experiments on every higher-order meta-paths.
- GCN [12]: A graph convolutional network which learns node embeddings in homogeneous networks. We test every higher-order meta-path-based directed adjacency matrices ($\hat{C}_{\phi\times p}$), then report their best result.
- GAT [13]: A graph neural network, which learns the influence weights of each neighbor via self-attention mechanism. We experiment on every higher-order meta-path-based adjacency matrices, then show their best result.
- HetGNN [17]: A heterogeneous graph convolutional network, which first samples heterogeneous neighbor nodes through a random walk, second group neighbor nodes by their types, then aggregates the information of nodes for every group, and finally aggregates the information of different groups into final embeddings. It treats different types of nodes separately but ignores differences in edge types.
- HOHGCN-binary: A variant of our model, which uses binary higher-order meta-path-based adjacency matrices ($C_{\phi\times p}$ is proposed in section III). When there are some nodes with a lot of neighbors, HOHGCN-binary may be significantly affected by these nodes. It is used to verify whether the calculation method of higher-order meta-path-based adjacency matrix is available.
- HOHGCN: Our proposed model in section IV.

### *C. Implementation Details*

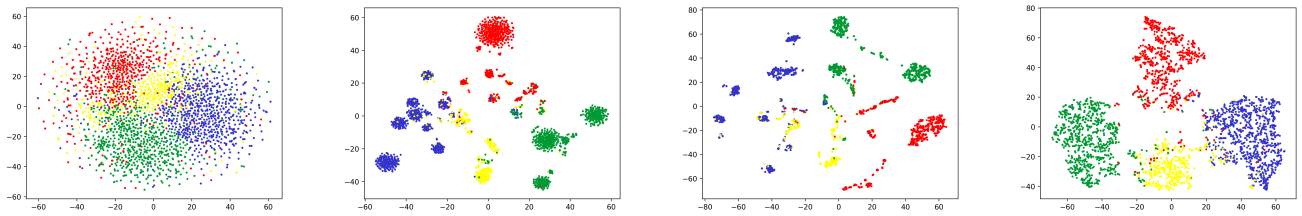
For our algorithm, we train a two-layer network as described by (6) using TensorFlow [31]. The predefined length $K = 2$, the maximum multiplier of meta-paths $P = 2$, and the embedding dimension $D = 64$ are used in our experiments. The above parameters are determined by parameter experiments in V-F. We use the Gradient Descent optimizer to train our model with a 0.0005 L2 regularization. Meanwhile, the learning rate $\alpha_t = 0.05 \times 0.997^t$ is used in epoch $t$, dropout is used in input and hidden layers.

For a fair comparison, the same training set, validation set, and test set are used for all algorithms. Besides, the embedding dimension is 64 for all baselines. Other settings for DeepWalk and metapath2vec include: walk length is 40, walks per node is 10, window size is 5, and the number of negative sample is 5.

### *D. Node Classification Experiments*

We report node classification accuracy in TABLE II. According to the results with different numbers of training nodes, such as 20, 50, and 100 training nodes per class, we have the following observations:

- Compared with the best performing baselines, the average accuracy of our model is improved by 1.45%, 1.40%, and 0.97% in DBLP, IMDB, and Amazon Kindle Review datasets. Our model outperforms all baselines, except GNN when using 2000 training nodes per class in Amazon Kindle Review dataset. It is because when using a large number of training sets on the reviews usefulness classification, the information collected from edge type "User-Review" is sufficient, but the edge type "Products-Review" brings the noise to HOHGCN. In general, with the increase of training nodes, the accuracy of classification increases. Moreover, it is noteworthy that for different datasets, the more relevant the neighbors are, the better our approach performs.
- Compared with DeepWalk and metapath2vec, our HOHGCN performs better, especially when using few training nodes. It is because our HOHGCN not only considers the heterogeneity of nodes and edges but also utilizes the node features of neighbors to generate better node embeddings.
- The graph convolutional networks, such as GAT, GCN, and HetGNN gain better performance compared with DeepWalk. It illustrates that collecting information from connected nodes is effective. Compared with our model,



(a) DeepWalk (b) metapath2vec (c) GCN (d) HOHGCN

Fig. 4: Embeddings visualization. Spots represent authors which are colored according to their research area.

GAT, GCN, and HetGNN have lower accuracy. It is because our model gathers information from different meta-paths, and the calculation of higher-order meta-path-based adjacency matrices reduces the impact of nodes with a lot of neighbors.

- Compare with the variant HOHGCN-binary, HOHGCN improves the average accuracy by 1.72%, 2.75%, and 0.19% in DBLP, IMDB, and Amazon Kindle Review experiments, respectively. It shows that learning embedding in higher-order meta-path-based adjacency matrices can achieve better performance than in binary adjacency matrices. It verifies the validity of using normalized relation-based adjacency matrices when calculating higher-order meta-path-based adjacency matrices.

### *E. Visualization*

We compare the visualization results of different embedding methods in Fig. 4. First, we learn the author embeddings on DBLP with different methods, such as DeepWalk, metapath2vec, GCN, and our proposed HOHGCN. Second, we visualize the authors' embeddings using t-SNE [32], which gives each author a location in a 2-dimensional map and colors the nodes according to their research areas.

In the visualization using DeepWalk, authors in various research areas are mixed. In metapath2vec and GCN, the distributions of authors belonging to the same research areas are dispersed. From Fig. 4, we find that the visualization of HOHGCN performs best. In HOHGCN, the authors with the same colors are distributed close. Meanwhile, there are distinct boundaries between different color groups. The above analysis demonstrates that our HOHGCN learns meaningful node embeddings.

### *F. Parameters Experiments*

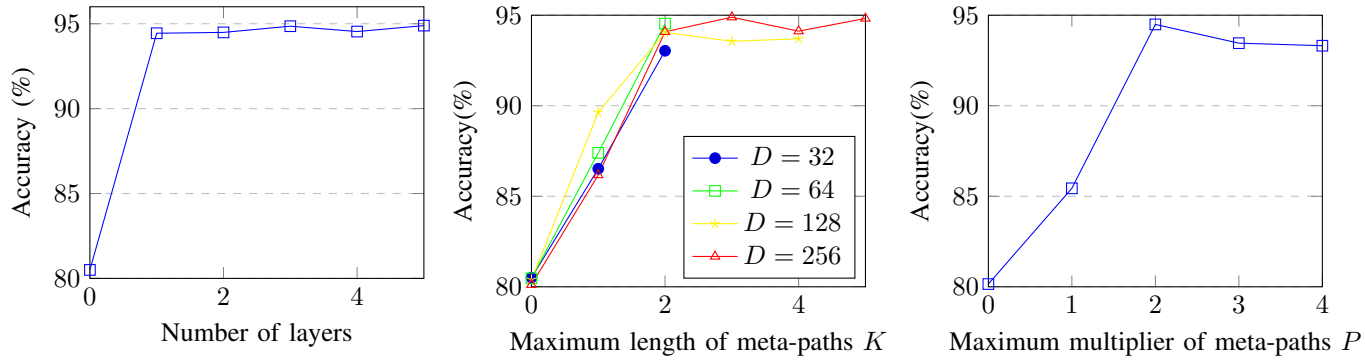
We discuss our parametric sensitivity via experiments in DBLP dataset.

- **The number of layers** $L$. Fig. 5a shows that with the layer number from 0 to 5, the accuracy rises initially and then almost unchanged. This is because HOHGCN needs to collect information from a suitable range of neighbor nodes, but an oversized range is unnecessary.
- **The dimension of embeddings** $D$. Some results are not shown in Fig. 5b because their embedding dimension is less than the number of higher-order meta-paths. When there are enough dimensions to encode information, embedding dimensions have little effect on accuracy.
- **The maximum length of meta-paths** $K$. In Fig. 5b, the accuracy increases rapidly when the maximum length of meta-paths increases from 0 to 2. It is because the meta-paths $AP$ and $APC$ are significant for author classification in the DBLP dataset. When the length exceeds 2, the accuracy is almost unchanged.
- **The maximum multiplier of meta-paths** $P$. As shown in Fig. 5c, when using 0-multiples meta-paths (just using the original features of nodes), the accuracy is only 80.15%. When using 2-multiples meta-paths, the accuracy rises to 94.49% because the 2-multiples meta-path $APCPA$ is important for DBLP. However, with the growth of the maximum multiplier, the accuracy drops slowly may because of the noise from distant nodes.

## VI. Conclusion

In this paper, we analyze the popular methods of meta-paths selected and show their disadvantages. To address this, we propose higher-order meta-paths, which not only lead to a few meta-paths but also contain various special meanings (such as communal relations). To reduce the impact of nodes with a lot of neighbors, a calculation method of higher-order meta-paths-based adjacency matrices is designed, which improves the accuracy by 1.72% and 2.75% in node classification experiments of DBLP and IMDB respectively. A novel heterogeneous graph convolution network is proposed to generate node representations, which collects information from higher-order meta-path-based neighbors. The analysis of computational complexity shows that our proposed model is of high efficiency and can be used for large-scale heterogeneous graphs. Our visualization and classification experiments prove that our HOHGCN outperforms state-of-the-art methods on three real-world datasets. Compared with the best performing baselines, there is 0.97% to 1.45% accuracy improvement in node classification tasks.

For graphs with numerous types of edges, we have to use a large embedding dimension to encode information from various higher-order meta-paths. In the future, we plan to learn the significance of every path, then generate node embeddings based on those important paths.




(a) Number of HOHGCN layers $L$

(b) Meta-paths' length & embedding dimension

(c) Maximum multiplier of meta-paths

Fig. 5: Parametric sensitivity w.r.t. the count of layers $L$ (excluding output layer), the dimension of embeddings $D$, the maximum length of meta-paths $K$, and the of meta-paths $P$.

## ACKNOWLEDGMENT

This work was supported by the Science and Technology Commission of Shanghai Municipality (Grant No. 19JC1420100 and 18dz2271000). We acknowledge its aid dearly.

## REFERENCES

[1] C. Shi, Y. Li, J. Zhang, Y. Sun, and P. S. Yu, "A survey of heterogeneous information network analysis," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 1, pp. 17–37, 2017.

[2] Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu, "Pathsim: Meta path-based top-k similarity search in heterogeneous information networks," *PVLDB*, vol. 4, no. 11, pp. 992–1003, 2011.

[3] X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, and P. S. Yu, "Heterogeneous graph attention network," in *WWW*, 2019, pp. 2022–2032.

[4] C. Shi, B. Hu, W. X. Zhao, and P. S. Yu, "Heterogeneous information network embedding for recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 2, pp. 357–370, 2019.

[5] Y. Lu, C. Shi, L. Hu, and Z. Liu, "Relation structure-aware heterogeneous information network embedding," in *AAAI*, 2019, pp. 4456–4463.

[6] D. Zhang, J. Yin, X. Zhu, and C. Zhang, "Metagraph2vec: Complex semantic path augmented heterogeneous network embedding," in *PAKDD*, 2018, pp. 196–208.

[7] T. Fu, W. Lee, and Z. Lei, "Hin2vec: Explore meta-paths in heterogeneous information networks for representation learning," in *CIKM*, 2017, pp. 1797–1806.

[8] T. Chen and Y. Sun, "Task-guided and path-augmented heterogeneous network embedding for author identification," in *WSDM*, 2017, pp. 295–304.

[9] S. Cao, W. Lu, and Q. Xu, "Grarep: Learning graph representations with global structural information," in *CIKM*, 2015, pp. 891–900.

[10] S. Abu-El-Haija, B. Perozzi, A. Kapoor, N. Alipourfard, K. Lerman, H. Harutyunyan, G. V. Steeg, and A. Galstyan, "Mixhop: Higher-order graph convolutional architectures via sparsified neighborhood mixing," in *ICML*, 2019, pp. 21–29.

[11] W. L. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *NIPS*, 2017, pp. 1024–1034.

[12] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *ICLR*, 2017.

[13] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *ICLR*, 2018.

[14] Y. Li, D. Tarlow, M. Brockschmidt, and R. S. Zemel, "Gated graph sequence neural networks," in *ICML*, 2016.

[15] F. Chen, S. Pan, J. Jiang, H. Huo, and G. Long, "DAGCN: dual attention graph convolutional networks," in *IJCNN*, 2019, pp. 1–8.

[16] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, pp. 1735–80, 12 1997.

[17] C. Zhang, D. Song, C. Huang, A. Swami, and N. V. Chawla, "Heterogeneous graph neural network," in *KDD*, 2019, pp. 793–803.

[18] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *NIPS*, 2013, pp. 3111–3119.

[19] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: online learning of social representations," in *KDD*, 2014, pp. 701–710.

[20] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *KDD*, 2016, pp. 855–864.

[21] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "LINE: large-scale information network embedding," in *WWW*, 2015, pp. 1067–1077.

[22] D. Wang, P. Cui, and W. Zhu, "Structural deep network embedding," in *KDD*, 2016, pp. 1225–1234.

[23] D. Zhu, P. Cui, D. Wang, and W. Zhu, "Deep variational network embedding in wasserstein space," in *KDD*, 2018, pp. 2827–2836.

[24] L. Chen, H. Zhang, and J. Wu, "Integrating dual user network embedding with matrix factorization for social recommender systems," in *IJCNN*, 2019, pp. 1–8.

[25] H. Wang, J. Wang, J. Wang, M. Zhao, W. Zhang, F. Zhang, X. Xie, and M. Guo, "Graphgan: Graph representation learning with generative adversarial nets," in *AAAI*, 2018, pp. 2508–2515.

[26] Q. Dai, Q. Li, J. Tang, and D. Wang, "Adversarial network embedding," in *AAAI*, 2018, pp. 2167–2174.

[27] Y. Sun, S. Wang, T. Hsieh, X. Tang, and V. G. Honavar, "MEGAN: A generative adversarial network for multi-view network embedding," in *IJCAI*, 2019, pp. 3527–3533.

[28] Y. Ban, J. Pu, Y. Chen, and Y. Wang, "NEGAN: network embedding based on generative adversarial networks," in *IJCNN*, 2018, pp. 1–8.

[29] Y. Dong, N. V. Chawla, and A. Swami, "metapath2vec: Scalable representation learning for heterogeneous networks," in *KDD*, 2017, pp. 135–144.

[30] Y. Sun and J. Han, "Mining heterogeneous information networks: a structural analysis approach," *SIGKDD Explorations*, vol. 14, no. 2, pp. 20–28, 2012.

[31] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. A. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, "Tensorflow: A system for large-scale machine learning," in *USENIX*, 2016, pp. 265–283.

[32] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.