HDGI: An Unsupervised Graph Neural Network for Representation Learning in Heterogeneous Graph

Yuxiang Ren,[1] Bo Liu,[2] Chao Huang,[2] Peng Dai,[2] Liefeng Bo,[2] Jiawei Zhang,[1]
[1]Florida State University, IFM Lab
[2]JD Finance America Corporation, AI lab
yuxiang@ifmlab.org, kfliubo@gmail.com, chuang7@nd.edu, peng.dai@jd.com, liefeng.bo@jd.com, jiawei@ifmlab.org

Abstract

Graph representation learning is to learn universal node representations that preserve both node attributes and structural information. When a graph is heterogeneous, the problem becomes more challenging than the homogeneous graph node learning problem. Inspired by the emerging information theoretic-based learning algorithm, we propose an unsupervised graph neural network **H**eterogeneous **D**eep **G**raph **I**nfomax (*HDGI*) for heterogeneous graph representation learning. By maximizing local-global mutual information, *HDGI* effectively learns high-level node representations that can be utilized in downstream graph-related tasks. Experiment results show that *HDGI* remarkably outperforms state-of-the-art unsupervised graph representation learning methods on both classification and clustering tasks. By feeding the learned representations into a parametric model, we even achieve comparable performance in node classification tasks when comparing with supervised end-to-end GNN models. A full version of this paper can be accessed in (Ren et al. 2019).

Introduction

Traditional machine learning methods focus on the features of individual nodes, which obstructs their ability to process graph data. Graph neural networks (GNNs) for representation learning of graphs learn nodes' new feature vectors through a recursive neighborhood aggregation scheme (Xu et al. 2019), which complete the fusion of node attributes and structural information in essence. A rich body of successful supervised graph neural network models have been developed (Kipf and Welling 2017a; Velickovic et al. 2018; You et al. 2018). However, labeled data is not always available in graph representation learning tasks. To alleviate the training sample insufficiency problem, unsupervised graph representation learning has aroused extensive research interest. Most of the existing unsupervised graph representation learning models can be roughly grouped into factorization-based models and edge-based models. Factorization-based models capture the global graph information by factorizing the sample affinity matrix (Zhang et al. 2016; Yang et al. 2015; Zhang et al. 2016). Those methods tend to ignore the node attributes and local neighborhood relationships. Edge-based models exploit the local and higher-order neighborhood information by edge connections or random-walk paths. Nodes tend to have similar representations if they are connected or co-occur in the same path (Kipf and Welling 2017b; Duran and Niepert 2017; W. Hamilton and Leskovec 2017; Grover and Leskovec 2016). Edge-based models are prone to preserve limited order node proximity and lack a mechanism to preserve the global graph structure. The recently proposed deep graph infomax (DGI) (Veličković et al. 2019) model provides a novel direction that maximizes the mutual information between graph patch representations and the corresponding high-level summaries of graphs.

In this paper, we explore the mutual information maximization learning framework in heterogeneous graph representation problems. The networked data in the real-world usually contain very complex structures (involving multiple types of nodes and edges), which can be formally modeled as the heterogeneous information networks (HIN). In this paper, we will misuse the terminologies "HIN" and "HG" (heterogeneous graph) without any differentiation. Compared with homogeneous graphs, heterogeneous graphs contain more detailed information and rich semantics with complex connections among multi-typed nodes. Taking the bibliographic network in Figure 1 as an example, it contains three types of nodes (Author, Paper and Subject) as well as two types of edges (Write and Belong-to). Besides, the individual nodes themselves also carry abundant attribute information (e.g., paper textual contents). The relations between paper nodes can be expressed by Paper-Author-Paper (PAP) and Paper-Subject-Paper (PSP) which represent papers written by the same author and papers belonging to the same subject respectively. In heterogeneous graph studies, since Y. Sun, J. Han, et al. proposed the concept of meta-path in (Sun et al. 2011), meta-path has been widely used to represent the composite relations with different semantics. GNNs initially proposed for the homogeneous graphs may encounter challenges to handle relations with different semantics.

To address the above challenges, we propose a novel meta-path based unsupervised graph neural network for heterogeneous graphs, namely **H**eterogeneous **D**eep **G**raph **I**nfomax (*HDGI*). In summary, our contributions in this paper can be summarized as follows:

- This paper presents the first model to apply mutual information maximization to representation learning in heterogeneous graphs.

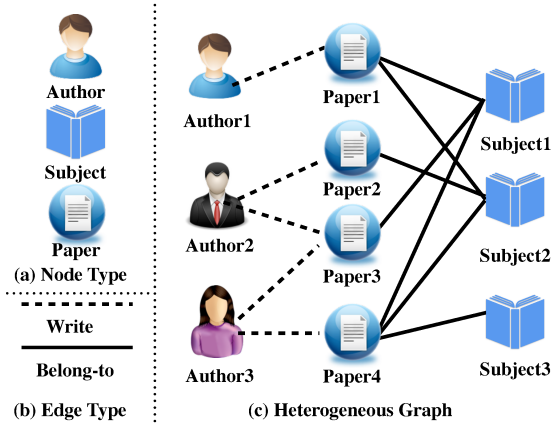

Figure 1: A heterogeneous bibliographic network.

- *HDGI* is a novel unsupervised graph neural network with the attention mechanism. It handles graph heterogeneity by utilizing an attention mechanism on meta-paths and deals with unsupervised setting by applying mutual information maximization.
- Our experiments demonstrate that the representations learned by *HDGI* are effective for both classification tasks and clustering tasks. Moreover, its performance can often beat state-of-the-art comparative supervised models.

Related Work

Graph representation learning. As a data type containing rich structural information, many models(Grover and Leskovec 2016; Tang et al. 2015) acting on graphs learn the representations of nodes based on the structure of the graph. DeepWalk (B. Perozzi and Skiena 2014) uses the set of random walks over the graph in SkipGram to learn node embeddings. Several methods (Ou et al. 2016; Wang et al. 2017) attempt to retrieve structural information through the matrix factorization. In order to handle the heterogeneity of graphs, metapath2vec (Dong, Chawla, and Swami 2017) samples random walks under the guidance of meta-paths and learns node embeddings through the skip-gram. HIN2Vec (Fu, Lee, and Lei 2017) learns the embedding vectors of nodes and meta-paths simultaneously while conducts prediction tasks. Wang et al. (Wang et al. 2019) consider the attention mechanism in heterogeneous graph learning.

Graph neural network. Graph neural networks (GNNs) (Zhang 2019) have made a lot of progress in graph representation learning. Most successful GNNs are based on supervised learning including GCN (Kipf and Welling 2017a), GAT (Velickovic et al. 2018), and GraphRNN (You et al. 2018). The unsupervised learning GNNs can be mainly divided into two categories, i.e., random walk-based (B. Perozzi and Skiena 2014; Grover and Leskovec 2016; Kipf and Welling 2017b; Duran and Niepert 2017; W. Hamilton and Leskovec 2017) and mutual information-based (Veličković et al. 2019).

Problem Formulation

In this section, we define critical concepts and formulate the problem of heterogeneous graph representation learning.

Definition 1. **Heterogeneous Graph**. A heterogeneous graph can be defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with a node type mapping function $\phi : \mathcal{V} \rightarrow \mathcal{T}$ and an edge type mapping function $\psi : \mathcal{E} \rightarrow \mathcal{R}$. Each node $v \in \mathcal{V}$ belongs to one particular node type in the node type set $\mathcal{T} : \phi(v) \in \mathcal{T}$, and each edge $e \in \mathcal{E}$ belongs to a particular edge type in the edge type set $\mathcal{R} : \psi(e) \in \mathcal{R}$. The sets of node types $\mathcal{T}$ and edge types $\mathcal{R}$ in heterogeneous graphs have the property that $|\mathcal{T}| + |\mathcal{R}| > 2$.

Meta path (Sun et al. 2011) is a well-used tool in heterogeneous graph analysis, and we will not re-introduce its definition here. Formally, we can represent the set of meta paths used in this paper as $\{\Phi_1, \Phi_2, \cdots, \Phi_P\}$.

Definition 2. **Meta-path based Adjacency Matrix**. Given a meta-path Φ, if there exist instances of the meta-path Φ between node $v_i \in \mathcal{V}_t$ and node $v_j \in \mathcal{V}_t$, we define that v_i and v_j are "connected neighbors" based on the meta-path Φ. Such indirect neighboring information can be represented as an adjacent matrix $\mathcal{A}^{\Phi} \in \mathbb{R}^{|\mathcal{V}_t| \times |\mathcal{V}_t|}$.

Problem Definition. **Heterogeneous Graph Representation Learning**. Given a heterogeneous graph $\mathcal{G}$ and the set of node feature vectors X, the representation learning task in $\mathcal{G}$ is to learn a low dimensional node representation $\mathcal{H} \in \mathbb{R}^{|\mathcal{V}| \times d}$ which can contain both structural information from $\mathcal{G}$ and node attributes from X. The learned representation $\mathcal{H}$ can be applied to the downstream graph-related tasks such as node classification and node clustering, etc. Note that we only focus on learning the representations of one specific type of nodes in this paper. We can represent such a set of nodes as the target-type nodes $\mathcal{V}_t$.

HDGI Methodology

HDGI Architecture Overview

Our method *HDGI* is mainly inspired by DIM (Hjelm et al. 2019) and DGI (Veličković et al. 2019). The high-level structure of *HDGI* is described in Figure 2. The input of *HDGI* should be a heterogeneous graph $\mathcal{G}$ along with the set of node feature vectors X and the meta-path set $\{\Phi_1, \Phi_2, \ldots, \Phi_P\}$. Based on the original graph $\mathcal{G}$ and the meta-path set, the set of meta-path based adjacency matrices $\{\mathcal{A}^{\Phi_1}, \mathcal{A}^{\Phi_2}, \ldots, \mathcal{A}^{\Phi_P}\}$ can be calculated. Local representation encoder is a hierarchical structure: learning node representations in terms of every meta-path based adjacency matrix respectively and then aggregating them through semantic-level attention. With the support of the output node representation $\mathcal{H}$ from the meta-path based local representation encoder, the global representation encoder $\mathcal{R}$ will output a graph-level summary vector $\vec{s}$. Negative samples generator $\mathcal{C}$ is responsible for generating negative nodes for the graph $\mathcal{G}$, and these negative nodes along with the positive nodes from $\mathcal{G}$ will be used to train the discriminator $\mathcal{D}$ with the object to maximize mutual information between positive nodes and the graph-level summary vector $\vec{s}$.

Meta-path based local representation encoder

The meta-path based node encoder has a two-level structure. We first derive a node representation from each meta-path based adjacency matrix $\mathcal{A}^{\Phi_i}, i = 1, ..., P$, respectively. After that the node representations based on all of $\{\mathcal{A}^{\Phi_i}\}_{i=1}^{P}$ are aggregated by an attention mechanism.

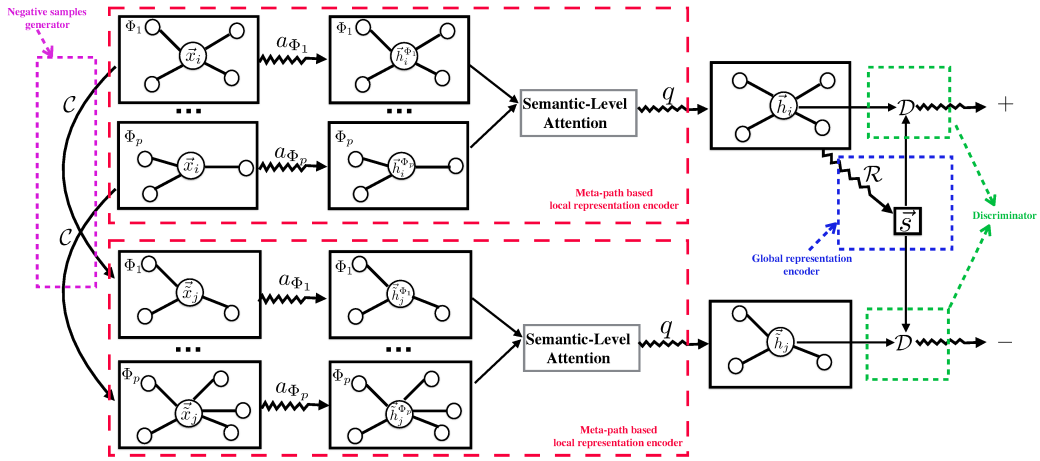

Figure 2: The high-level structure of Heterogeneous Deep Graph Infomax (*HDGI*)

Node-level learning Each of $\mathcal{A}^{\Phi_i}$ can be viewed as a homogeneous graph. At this step our target is to derive a node representation containing the information of initial node feature $X = \{\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_N\}, \vec{x}_i \in \mathbb{R}^F, N = |\mathcal{V}_t|$ and $\mathcal{A}^{\Phi_i}$. In *HDGI*, we try to use GCN (Kipf and Welling 2017a) and GAT (Velickovic et al. 2018) as components in the local representation encoder respectively.

For each meta-path $\Phi_m, m \in \{1, 2, \ldots, P\}$, a node-level encoder:

$$a_{\Phi_m}(X, \mathcal{A}^{\Phi_m}) = \mathcal{H}^{\Phi_m} = \{\vec{h}_1^{\Phi_m}, \vec{h}_2^{\Phi_m}, \ldots, \vec{h}_N^{\Phi_m}\} \tag{1}$$

a_{Φ_m} will be learned in order to output the high-level representation $\vec{h}_i^{\Phi_m} \in \mathbb{R}^{F'}$ for the node i. After the node-level learning, we can obtain the set of node representations $\{\mathcal{H}^{\Phi_1}, \mathcal{H}^{\Phi_2}, \ldots, \mathcal{H}^{\Phi_P}\}$ based on meta-path connections with different semantics. In the experiment section, we will show the performance along with the analysis of using these two GNNs as components.

Semantic-level learning In order to obtain the more general representations of the nodes, we need to fuse these representations $\{\mathcal{H}^{\Phi_1}, \mathcal{H}^{\Phi_2}, \ldots, \mathcal{H}^{\Phi_P}\}$. The key issue to accomplish this combination is exploring how much each meta-path should contribute to the final representations. Here we add a semantic attention layer L_{att} to learn the weights that each meta-path should be assigned:

$$\{S^{\Phi_1}, S^{\Phi_2}, \ldots, S^{\Phi_P}\} = L_{att}(\mathcal{H}^{\Phi_1}, \mathcal{H}^{\Phi_2}, \ldots, \mathcal{H}^{\Phi_P}) \tag{2}$$

Then fuse the representations of multiple semantics according to the learned weights $\{S^{\Phi_1}, S^{\Phi_2}, \ldots, S^{\Phi_P}\}$.

In order to make representations based on different meta-paths comparable, we first need to transform each node's representation with a linear transformation, parameterized by a shared weight matrix $W \in \mathbb{R}^{F'' \times F'}$ and a shared bias vector b. The importance of the representations based on different meta-paths will be measured by a shared attention vector $q \in \mathbb{R}^{F''}$. The importance of the meta-path Φ_i can be calculated as:

$$e^{\Phi_i} = \frac{1}{N} \sum_{j=1}^{N} \tanh(q^{\mathrm{T}} \cdot [W \cdot \vec{h}_j^{\Phi_i} + b]) \tag{3}$$

According to the importance of meta-paths, we will normalize them using the softmax function:

$$S^{\Phi_i} = \mathrm{softmax}(e^{\Phi_i}) = \frac{\exp(e^{\Phi_i})}{\sum_{j=1}^{P} \exp(e^{\Phi_j})} \tag{4}$$

Once obtained, the weights of different meta-paths are used as coefficients to conduct a linear combination:

$$\mathcal{H} = \sum_{i=1}^{P} S^{\Phi_i} \cdot \mathcal{H}^{\Phi_i} \tag{5}$$

$\mathcal{H}$ will serve as the final output local representations.

Global Representation Encoder

The learning object of *HDGI* is to maximize the mutual information between local representations and the global representation. The local representations of nodes are included in $\mathcal{H}$, and we need the summary vector $\vec{s}$ to represent the global information of the entire graph. Based on $\mathcal{H}$, we examined three candidate encoder functions:

Averaging encoder function.

$$\vec{s} = \sigma\left(\frac{1}{N} \sum_{i=1}^{N} \vec{h}_i\right) \tag{6}$$

Pooling encoder function.

$$\vec{s}_{pool} = max(\{\sigma(W_{pool}\vec{h}_i + b), i \in \{1, 2, \ldots, N\}) \tag{7}$$

where max denotes the element-wise max operator and σ is a nonlinear activation function.

Set2vec encoder function. The final encoder function we examine is Set2vec (Oriol Vinyals 2016) which is based on an LSTM architecture.

Among these functions, the simple averaging function achieves the best performance in our experiments.

HDGI Learning

Negative samples generator The negative samples generator is responsible for generating negative samples (nodes not exist in the original graph), which will be used to train the mutual information based discriminator.

As our target is to maximize the mutual information between positive nodes and the graph-level summary vector, the generated negative samples will affect the structural information captured by the model. In heterogeneous graph $\mathcal{G}$, we have rich and complex structural information from the

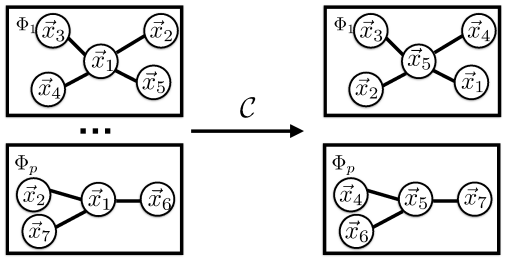

Figure 3: The example of generating negative samples

set of meta-path based adjacency matrices. In our negative samples generator:

$$(\tilde{X}, \{\mathcal{A}_{\Phi_1}, \mathcal{A}_{\Phi_2}, \ldots, \mathcal{A}_{\Phi_P}\}) = \mathcal{C}(X, \{\mathcal{A}_{\Phi_1}, \mathcal{A}_{\Phi_2}, \ldots, \mathcal{A}_{\Phi_P}\}) \tag{8}$$

we will keep all meta-path based adjacency matrices unchanged which can make the overall structure of $\mathcal{G}$ stable. Then we shuffle the rows of the independent node information matrix X, which changes the index of nodes in order to corrupt the node-level connections among them. According to the spectral theory, the structure of the whole graph does not change, but the initial feature corresponding to each node has changed. We provide a simple example to illustrate the procedure of $\mathcal{C}$ in Figure 3.

Mutual information based discriminator According to the proof in (Belghazi et al. 2018), the mutual information can be estimated by gradient descent over neural networks. Here, we estimate the mutual information by training a discriminator $\mathcal{D}$ to distinguish between $(\vec{h}_i, \vec{s})$ and $(\vec{\tilde{h}}_j, \vec{s})$. The sample $(\vec{h}_i, \vec{s})$ is denoted as positive because node $\vec{h}_i$ belongs to the original graph, and $(\vec{\tilde{h}}_j, \vec{s})$ is denoted as negative as the node $\vec{\tilde{h}}_j$ is the generated fake one. The discriminator $\mathcal{D}$ is a binary classifier:

$$\mathcal{D}(\vec{h}_i, \vec{s}) = \sigma(\vec{h}_i^{\mathrm{T}} W \vec{s}) \tag{9}$$

Based on the relationship (Hjelm et al. 2019) between Jensen-Shannon divergence and the mutual information, we can maximize the mutual information with the binary cross-entropy loss of the discriminator:

$$\mathcal{L}(\mathcal{H}, \tilde{\mathcal{H}}, \vec{s}) = \frac{1}{2N}\left(\sum_{i=1}^{N} \mathbb{E}_{(X)}[\log \mathcal{D}(\vec{h}_i, \vec{s})] + \sum_{j=1}^{N} \mathbb{E}_{(\tilde{X})}[\log(1 - \mathcal{D}(\vec{\tilde{h}}_j, \vec{s}))]\right)$$

The above loss can be optimized through the gradient descent, and the representations of nodes can be learned when the optimization is completed.

Evaluation

Datasets

We evaluate the performance of *HDGI* in three heterogeneous graphs, and the detailed descriptions of them are shown in Table 1.

Experimental Setup

The most commonly used tasks to measure the quality of learned representations are node classification and node clustering (Wang et al. 2019) in graph-related research works. We evaluate *HDGI* from both two kinds of tasks.

Table 1: Summary of heterogeneous graphs in experiments

Dataset	Node-type	# Nodes	Edge-type	# Edges	Meta-path
ACM	Paper (**P**)	3025	Paper-Author	9744	PAP
	Author (**A**)	5835	Paper-Subject	3025	PSP
	Subject (**S**)	56			
IMDB	Movie (**M**)	4275	Movie-Actor	12838	MAM
	Actor (**A**)	5431	Movie-Director	4280	MDM
	Director (**D**)	2082	Movie-keyword	20529	MKM
	Keyword (**K**)	7313			
DBLP	Author (**A**)	4057	Author-Paper	19645	APA
	Paper (**P**)	14328	Paper-Conference	14328	APCPA
	Conference (**C**)	20	Paper-Term	88420	APTPA
	Term (**T**)	8789			

Comparison methods We compare our method *HDGI* to the following state-of-the-art methods including both supervised and unsupervised methods:

Unsupervised methods

- *Raw Feature*: It represents the bag-of-words embedding, and we will directly test them in tasks.
- *Metapath2vec (Dong, Chawla, and Swami 2017)*: A meta-path based heterogeneous graph embedding method, but it can only handle specific one meta-path.
- *DeepWalk (B. Perozzi and Skiena 2014)*: A random walk based graph embedding method, but it is designed to deal with homogeneous graph.
- *DeepWalk+Raw Feature(DeepWalk+F)*: We concatenate the embeddings learned from DeepWalk and the bag-of-words embeddings as the final representations.
- *DGI (Veličković et al. 2019)*: A mutual information based unsupervised learning method which is proposed for homogeneous graph.
- *HDGI-C*: The proposed method which uses graph convolutional network to capture local representations.
- *HDGI-A*: The proposed method which uses attention mechanism (GAT (Velickovic et al. 2018)) to learn local representations.

Supervised methods

- *GCN (Kipf and Welling 2017a)*: GCN is a semi-supervised methods for the node classification in homogeneous graphs.
- *GAT (Velickovic et al. 2018)*: GAT applies the attention mechanism on homogeneous graphs which requires supervised setting.
- *HAN (Wang et al. 2019)*: HAN employs node-level attention and semantic-level attention to capture the information from all meta-paths.

For methods designed for homogeneous graphs including *DeepWalk*, *DGI*, *GCN*, *GAT*, we test the graph ignoring the heterogeneity and graphs constructed from every meta-path based adjacency matrix respectively, then report the best result. *Metapath2vec* can only handle one kind of meta-path, thus we test all meta-paths for it and report the best results.

Table 2: The results of node classification tasks

Available data			X	A		X, A				X, A, Y		
Dataset	Train	Metric	Raw Feature	Metapath2vec	DeepWalk	DeepWalk+F	DGI	HDGI-A	HDGI-C	GCN	GAT	HAN
ACM	20%	Micro-F1	0.8590	0.6125	0.5503	0.8785	0.9104	0.9178	**0.9227**	0.9250	0.9178	0.9267
		Macro-F1	0.8585	0.6158	0.5582	0.8789	0.9104	0.9170	**0.9232**	0.9248	0.9172	0.9268
	80%	Micro-F1	0.8820	0.6378	0.5788	0.8965	0.9175	0.9333	**0.9379**	0.9317	0.9250	0.9400
		Macro-F1	0.8802	0.6390	0.5825	0.8960	0.9155	0.9330	**0.9379**	0.9317	0.9248	0.9403
DBLP	20%	Micro-F1	0.7552	0.6985	0.2805	0.7163	0.8975	0.9062	**0.9175**	0.8192	0.8244	0.8992
		Macro-F1	0.7473	0.6874	0.2302	0.7063	0.8921	0.8988	**0.9094**	0.8128	0.8148	0.8923
	80%	Micro-F1	0.8325	0.8211	0.3079	0.7860	0.9150	0.9192	**0.9226**	0.8383	0.8540	0.9100
		Macro-F1	0.8152	0.8014	0.2401	0.7799	0.9052	0.9106	**0.9153**	0.8308	0.8476	0.9055
IMDB	20%	Micro-F1	0.5112	0.3985	0.3913	0.5262	0.5728	0.5482	**0.5893**	0.5931	0.5985	0.6077
		Macro-F1	0.5107	0.4012	0.3888	0.5293	0.5690	0.5522	**0.5914**	0.5869	0.5944	0.6027
	80%	Micro-F1	0.5900	0.4203	0.3953	0.6017	0.6003	0.5861	**0.6592**	0.6467	0.6540	0.6600
		Macro-F1	0.5884	0.4119	0.4001	0.6049	0.5950	0.5834	**0.6646**	0.6457	0.6550	0.6586

Table 3: Evaluation results on the node clustering task

Data	ACM		DBLP		IMDB	
Method	NMI	ARI	NMI	ARI	NMI	ARI
DeepWalk	25.47	18.24	7.40	5.30	1.23	1.22
Raw Feature	32.62	30.99	11.21	6.98	1.06	1.17
DeepWalk+F	32.54	31.20	11.98	6.99	1.23	1.22
Metapath2vec	27.59	24.57	34.30	37.54	1.15	1.51
DGI	41.09	34.27	59.23	61.85	0.56	2.6
HDGI-A	**57.05**	**50.86**	52.12	49.86	0.8	1.29
HDGI-C	54.35	49.48	**60.76**	**62.67**	**1.87**	**3.7**

Results

Node classification task In the node classification task, we will train a logistic regression classifier for unsupervised learning methods, while the supervised methods can output the classification result as end-to-end models. We conduct the experiments with two different training-ratios (20% and 80%). To keep the results stable, we repeat the classification process for 10 times and report the Macro-F1 and Micro-F1 of all methods in Table 2. We can observe that *HDGI-C* outperforms all other unsupervised learning methods. When compared with the supervised learning methods but designed for homogeneous graphs like *GCN* and *GAT*, *HDGI* can perform much better as well which proves that the type information and semantic information are very important and need to be handled carefully instead of directly ignoring them in heterogeneous graphs. *HDGI* is also competitive with the result reported from the supervised model *HAN* which is designed for heterogeneous graphs. The reason should be that *HDGI* can capture more global structural information when the mutual information plays a strong role in reconstructing the representation, while supervised loss based GNNs overemphasize the direct neighborhoods (Veličković et al. 2019). This, on the other hand, also suggests that the features learned through supervised learning in graphs may have limitations, either from the structure or from a task-based preference.

Node clustering task In the node clustering task, we use the KMeans to conduct the clustering based on the learned representations. The number of clusters K is set as the number of the node classes. We will not use any label in this unsupervised learning task and make the comparison among all unsupervised learning methods. We repeat the clustering process for 10 times and report the average NMI and ARI of all methods in Table 2. *DeepWalk* can not perform well because they are not able to handle the heterogeneity of graphs. *Metapath2vec* can not handle diversity semantic information simultaneously which makes the representations not effective enough. The verification based on node clustering tasks also demonstrates that *HDGI* can learn effective representation considering the structural information, the semantic information and the node independent information simultaneously.

HDGI-A vs HDGI-C From the comparison between *HDGI-C* and *HDGI-A* in node classification tasks, the difference in results between them reflects some interesting things. *HDGI-C* has better performance than *HDGI-A* in all experiments, which means that the graph convolution works better than the attention mechanism in capturing local representation. We insist that the reason is that the graph attention mechanism is strictly limited to the direct neighbors of nodes, the graph convolution considering hierarchical dependencies can see farther than the graph attention.

VI Conclusion

In this paper, we propose an unsupervised graph neural network model, *HDGI*, which learns node representations in heterogeneous graphs. We demonstrate the effectiveness of learned representations in three heterogeneous graphs. *HDGI* is particularly competitive in node classification tasks with state-of-the-art supervised methods, where they have the additional supervised label information. We are optimistic that mutual information maximization will be a promising future direction for unsupervised representation learning.

Acknowledgment

This work is partially supported by FSU and by NSF through grant IIS-1763365.

References

B. Perozzi, R. A.-R., and Skiena, S. 2014. Deepwalk: Online learning of social representations. In *KDD*.

Belghazi, M. I.; Baratin, A.; Rajeswar, S.; Ozaira, S.; Bengio, Y.; Courville, A.; and Hjelm, R. D. 2018. Mine: Mutual information neural estimation. In *ICML*.

Dong, Y.; Chawla, N. V.; and Swami, A. 2017. metapath2vec: Scalable representation learning for heterogeneous networks. In *KDD*.

Duran, A. G., and Niepert, M. 2017. Learning graph representations with embedding propagation. In *NIPS*.

Fu, T.-Y.; Lee, W.-C.; and Lei, Z. 2017. Hin2vec: Explore meta-paths in heterogeneous information networks for representation learning. In *CIKM*.

Grover, A., and Leskovec, J. 2016. node2vec: Scalable feature learning for networks. In *KDD*.

Hjelm, R. D.; Fedorov, A.; Lavoie-Marchildon, S.; Grewal, K.; Bachman, P.; Trischler, A.; and Bengio, Y. 2019. Learning deep representations by mutual information estimation and maximization. In *ICLR*.

Kipf, T. N., and Welling, M. 2017a. Semi-supervised classification with graph convolutional networks. In *ICLR*.

Kipf, T. N., and Welling, M. 2017b. Variational graph auto-encoders. In *ICLR*.

Oriol Vinyals, Samy Bengio, M. K. 2016. Order matters: Sequence to sequence for sets. In *ICLR*.

Ou, M.; Cui, P.; Pei, J.; Zhang, Z.; and Zhu, W. 2016. Asymmetric transitivity preserving graph embedding. In *KDD*.

Ren, Y.; Liu, B.; Huang, C.; Dai, P.; Bo, L.; and Zhang, J. 2019. Heterogeneous deep graph infomax. *arXiv preprint arXiv:1911.08538*.

Sun, Y.; Han, J.; Yan, X.; Yu, P.; and Wu, T. 2011. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. *PVLDB*.

Tang, J.; Qu, M.; Wang, M.; Zhang, M.; Yan, J.; and Mei, Q. 2015. Line: Large-scale information network embedding. In *WWW*.

Velickovic, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; and Bengio, Y. 2018. Graph attention networks. In *ICLR*.

Veličković, P.; Fedus, W.; Hamilton, W. L.; Liò, P.; Bengio, Y.; and Hjelm, R. D. 2019. Deep graph infomax. *International Conference on Learning Representation*.

W. Hamilton, Z. Y., and Leskovec, J. 2017. Inductive representation learning on large graphs. In *NIPS*.

Wang, X.; Cui, P.; Wang, J.; Pei, J.; Zhu, W.; and Yang, S. 2017. Community preserving network embedding. In *AAAI*.

Wang, X.; Ji, H.; Shi, C.; Wang, B.; Cui, P.; Yu, P.; and Ye, Y. 2019. Heterogeneous graph attention network. In *WWW*.

Xu, K.; Hu, W.; Leskovec, J.; and Jegelka, S. 2019. How powerful are graph neural networks? In *ICLR*.

Yang, C.; Liu, Z.; Zhao, D.; Sun, M.; and Chang, E. 2015. Network representation learning with rich text information. In *International Joint Conference on Artificial Intelligence*.

You, J.; Ying, R.; Ren, X.; Hamilton, W.; and Leskovec, J. 2018. Graphrnn: Generating realistic graphs with deep auto-regressive models. In *ICML*.

Zhang, D.; Yin, J.; Zhu, X.; and Zhang, C. 2016. Collective classification via discriminative matrix factorization on sparsely labeled networks. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, 1563–1572. ACM.

Zhang, J. 2019. Graph neural networks for small graph and giant network representation learning: An overview. Technical report, IFM lab.