Master's Thesis
석사 학위논문

Exploiting various patterns for heterogeneous graph attention network

Eunjeong Yi(이 은 정 李 銀 淨)

Department of
Information and Communication Engineering

DGIST

2021

Master's Thesis
석사 학위논문

Exploiting various patterns for heterogeneous graph attention network

Eunjeong Yi(이 은 정 李 銀 淨)

Department of
Information and Communication Engineering

DGIST

2021

Exploiting various patterns for heterogeneous graph attention network

Advisor: Professor Jemin Lee
Co-advisor: Professor Min-Soo Kim

by

Eunjeong Yi
Department of Information and Communication Engineering
DGIST

A thesis submitted to the faculty of DGIST in partial fulfillment of the requirements for the degree of Master of Science in the Department of Information and Communication Engineering. The study was conducted in accordance with Code of Research Ethics[1]

06. 07. 2021

Approved by

Professor Jemin Lee (signature)
(Advisor)

Professor Min-Soo Kim (signature)
(Co-Advisor)

[1] Declaration of Ethical Conduct in Research: I, as a graduate student of DGIST, hereby declare that I have not committed any acts that may damage the credibility of my research. These include, but are not limited to: falsification, thesis written by someone else, distortion of research findings or plagiarism. I affirm that my thesis contains honest conclusions based on my own careful research under the guidance of my thesis advisor.

Exploiting various patterns for heterogeneous graph attention network

Eunjeong Yi

Accepted in partial fulfillment of the requirements for the degree of Master of Science.

06. 07. 2021

Head of Committee	Prof. Jemin Lee	~~(signature)~~
Committee Member	Prof. Min-Soo Kim	(signature)
Committee Member	Prof. Daehoon Kim	(signature)

MS/IC
201852013

이 은 정. Eunjeong Yi. Exploiting various patterns for heterogeneous graph attention network. Department of Information and Communication Engineering. 2021. 30p. Advisors Prof. Jemin Lee, Co-Advisors Prof. Min-Soo Kim

ABSTRACT

Recently, Graph neural networks(GNNs) have been improved under the influences by concepts of various deep learning techniques, such as attention, auto-encoder, and recurrent network. However, in the real world, since various graphs, such as social network, citation network, and the e-commerce data, have the multi-types of vertices and edges, most GNNs considering a homogeneous graph as input data is not suitable due to ignoring the heterogeneity. Meta-path based methods have been researched to capture both the heterogeneity and structural information of heterogeneous graphs. As meta-path is a kind of graph pattern, we extend utilizing meta-paths to exploiting graph patterns. In this paper, we propose a heterogeneous graph attention network for exploiting triangle patterns called TP-HAN and extend TP-HAN to utilize various graph patterns. Through experiments using real-world datasets, we show that both TP-HAN and VP-HAN has better performance than the state-of-art heterogeneous graph attention network.

Keywords: Graph convolutional networks, Heterogeneous graph, Graph pattern

List of Contents

List of tables

List of figures

I. INTRODUCTION

Recently, Graph Neural Networks(GNNs) have received attention in deep learning over graphs. There have been many attempts and applications of various techniques, which were successful in deep learning, to GNNs.

For example, Graph Convolutional Networks(GCNs) [1, 6] proposed neighbor aggregation method which is the basic of graph convolution operation. Veličković, P. et al. [2] proposed the attention mechanism that makes the neural network learn which neighbors are important. Chiang, W. L. et al. [8] proposed mini-batch algorithm based on graph clustering to reduce the memory and computational resource requirements for processing large-scale graphs.

Most studies have considered GCNs for homogeneous graphs. Actually, the real-world data, such as citation network and protein-protein interaction network, is a heterogeneous graph that has multi-types of vertices and edges. So, naïve approaches are to ignore multi-type of vertices and edges, and regard as homogeneous graph. Since they ignore the properties of heterogeneous graphs, a method exploiting the properties is needed. In recent studies, some methods utilizing meta-paths are proposed to consider the heterogeneity. Wang, X. et al. [4] transformed a heterogeneous graph into homogeneous graphs by manually selected meta-paths and applied the attention mechanism [2]. Because this is operated with extracted homogeneous graphs for each meta-path, it can have worse performance depending on manually selected meta-paths. Yun, S. et al. [5] transformed a heterogeneous

graph into useful meta-path based graphs by automatically generating meta-paths. As the number of stacked layers determines the maximum length of generated meta-paths, it requires heavy computation to generate long meta-paths. Also, since it learns to generate useful meta-paths as combinations of edge types, performance can be affected with more edge types.

We focus on that the meta-path is a kind of graph pattern. Our approach is to leverage graph patterns instead of meta-paths to solve the above limits of meta-path based methods. As the starting point of our approach, we propose a heterogeneous graph attention network exploiting triangle patterns called TP-HAN. TP-HAN improves the performance by exploiting triangle patterns compared with the heterogeneous graph attention network [4] called HAN. After verifying the effect of exploiting triangle patterns, we extend TP-HAN to exploit various pattern called VP-HAN. VP-HAN improves the performance of both TP-HAN and HAN. We verify that exploiting a specific combination of graph patterns shows high performance rather than simply using many patterns. We evaluate the performance of TP-HAN and VP-HAN by comparing with HAN.

The rest of this paper is organized as follows. Section 2 presents the preliminary. Section 3 presents the related works. We propose TP-HAN and VP-HAN in section 4. We show the experimental result in section 5. Section 6 summarizes this paper and discusses the future work.

II. Preliminary

In this section, we explain the heterogeneous graph in section 2.1, the meta-path in section 2.2, and graph pattern in section 2.3. We summarize the notations used in this paper in Table 1.

Table 1: Summary of notations

Notation	Description
Φ	Meta-path
P	Graph pattern
X	Vertex feature
SG_P	Subgraph which is monomorphic to pattern P
A_t	Adjacency matrix for edge type t
$N_{v_i}^P$	Pattern P based neighbors of vertex v_i
$\alpha_{v_i v_j}^{\Phi}$	The vertex-level attention of vertex pair (v_i, v_j) for meta-path Φ
$\beta_{v_i}^{\Phi}$	The semantic-level attention of vertex v_i for meta-path Φ
$h_{v_i}^{\Phi}$	The vertex-level embedding of vertex v_i for meta-path Φ
W	The parameters of the graph convolutional network
Z	The final embedding

2.1 Heterogeneous graph

Heterogeneous graph is a graph type that has multiple types of vertices and edges. As shown in Figure 1(a), an academic graph consists of Author(A), Paper(P), and Conference(C) types of vertices and multiple types of edges such as a relation between author and paper, a relation between paper and conference, and a relation between paper and paper.

2.2 Meta-path

Meta-path[3] is widely used structure to capture the semantic information of heterogeneous graph. It

is the composite relations of multiple relations, i.e., $v_1 \xleftarrow{r_1} v_2 \xleftarrow{r_2} \ldots \xleftarrow{r_{l-1}} v_l$, where $r_i \in R$ denotes the i-th edge type of meta-path. The composite relations $R = r_1 \cdot r_2 \cdot \ldots \cdot r_{l-1}$ is a l-length meta-path from v_1 and v_l, where $r_1 \cdot r_2$ denotes the composition of relation r_1 and r_2.

As shown in Figure 1(b), there are two examples of meta-paths: Author-Paper-Author(A-P-A) and Author-Paper-Conference-Paper-Author(A-P-C-P-A). For example, two authors a_1 and a_2 are connected via a paper p_3 at meta-path A-P-A. Two authors a_1 and a_3 are connected via a paper p_1, a conference c_2, and a paper p_3 on meta-path A-P-C-P-A.

2.3 Graph pattern

Graph pattern is a subgraph which is frequently seen. When there are two graphs $G = (V, E)$ and $G' = (V', E')$, graph G' is a subgraph of graph G if $V' \subseteq V$ and $E' \subseteq E$, denoted as $G' \subseteq G$. For a graph $G'' = (V'', E'')$, if there is a one-one correspondence between the vertices of G' and those of G'' such that the number of edges $(u'', v'') \in E''$ is equal to the number of edges $\big(f(u''), f(v'')\big) \in E'$ with mapping function $f: V'' \to V'$, two graphs G'' and G' are isomorphic[12]. If subgraph G' of graph G is frequently seen, graph G'', which is isomorphic to graph G', is a graph pattern.

For example, triangle is a basic and important pattern in graph systems because the triangle is a cycle and also a most minimal clique. Triangle counting and listing are widely used in graph systems. As shown in

Figure 1(c), there are two examples of triangles. For example, author a_1, paper p_1, and paper p_3 form a triangle. Also, the other triangle, which is formed by author a_1, paper p_1, paper p_3, and author a_2, appears only in heterogeneous graph. This type of triangle starts at a vertex and arrives at another vertex of the same type via two different vertices. The reason for these two types of triangles is the heterogeneity of heterogeneous graph.

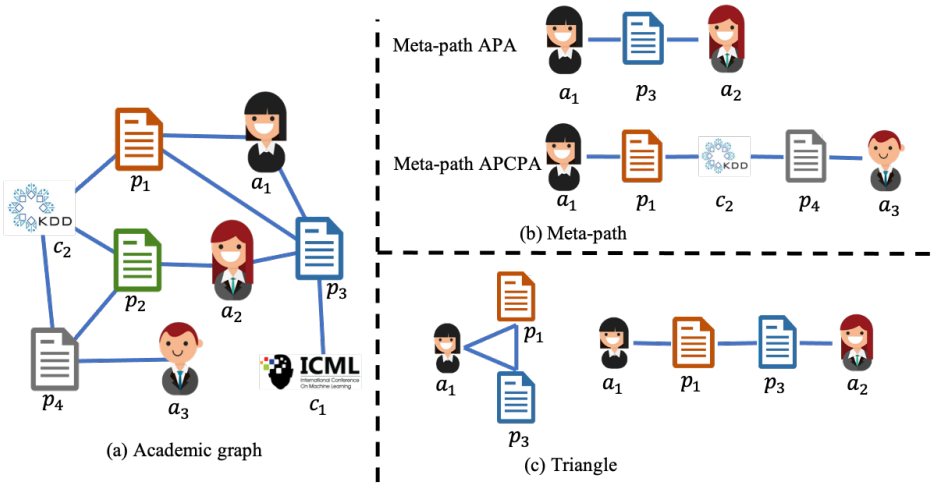


Figure 1: An example of an academic graph. (a) A heterogeneous graph consists of three types of vertices (i.e., author, paper, and conference) and three types of edges (i.e., author-paper relations, paper-conference relations, and paper-paper relations). (b) Examples of two types of meta-paths (i.e., Author-Paper-Author and Author-Paper-Conference-Paper-Author). (c) Examples of triangles.

III. Related work

In this section, we explain graph convolutional networks in section 3.1 and graph convolutional networks for heterogeneous graph in section 3.2.

3.1 Graph Convolutional Network(GCN)

Typical neural networks, which receive vector format as input data, are not suitable for processing graph that cannot be directly represented as vector. Graph Convolutional Networks(GCNs) are introduced to deal with graph structure. Recently, GCNs, which are proposed graph convolutional operation directly using graph structure, receive attention. Graph convolutional operation is based on neighbor aggregation, inspired by 2D convolutional operation that adjacent pixels are aggregated to a center pixel. Kipf, T. N. et al.[1] has proposed neighbor aggregation method using adjacency matrix multiplication. It is the layer-wise propagation rule:

$$H^{(l+1)} = \sigma\left(\widetilde{D}^{-\frac{1}{2}}\tilde{A}\widetilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right) , \quad (1)$$

where $H^{(l)}$ denotes the embedding of l-th layer. The adjacency matrix of graph with added self-connections is denoted as $\tilde{A} = A + I$. The degree matrix is denoted as $\widetilde{D}_{ii} = \Sigma_j \tilde{A}_{ij}$. Hamilton, W. L. et al. [6] proposed aggregator functions using various operations like mean, LSTM, and GCN[1]. Veličković, P. et al. [2] proposed Graph

Attention Networks(GATs) which apply the attention mechanism to GCNs. GATs assign the importance to neighbors for each vertex.

3.2 Graph convolutional networks for heterogeneous graph

Since many researches for GCNs assume that input graph is homogeneous graph, they are not suitable to process heterogeneous graphs. There have been proposed some meta-path based GCNs to process heterogeneous graph.

3.2.2 Heterogeneous graph Attention Network(HAN)

Wang, X. et al. [4] proposed the hierarchical attention mechanism, which consists of vertex-level attention and semantic-level attention. The vertex-level attention is to assign different importance to neighbors at the extracted homogeneous graph from the heterogeneous graph. The semantic-level attention is to assign different importance to each meta-path.

Figure 2 shows the overall process of HAN for author a_2 utilizing meta-paths A-P-A and A-P-C-P-A. First, homogeneous graphs are extracted from input graph for each meta-path. A path from author a_2 to author a_1 via paper p_3 is corresponding to meta-path A-P-A. So, it is regarded that author a_2 is connected with author a_1 at the graph based on meta-path A-P-A. Likewise, there are some paths matching to meta-path A-P-C-P-A. For example, author a_2 is connected with author a_1 because there is a path from author a_2 to author a_1 via

paper p_2, conference c_2, and paper p_1. HAN calculates the vertex-level attention between author a_1 and each neighbor on the extracted graph for each meta-path. For meta-path A-P-A, author a_1 is aggregated to author a_2 with the importance $\alpha_{a_2 a_1}^{\Phi_{APA}}$. Also, authors a_1 and a_3 are aggregated to author a_2 with the importance $\alpha_{a_2 a_1}^{\Phi_{APCPA}}$ and $\alpha_{a_2 a_3}^{\Phi_{APCPA}}$ for the meta-path A-P-C-P-A. After aggregating neighbors with the vertex-level attention, HAN calculates the semantic-level attentions of meta-path A-P-A and A-P-C-P-A. Author a_2's final embedding is generated as aggregating the vertex-level embedding $h_{a_2}^{\Phi_{APA}}$ and $h_{a_2}^{\Phi_{APCPA}}$ with the semantic-level attention $\beta_{a_2}^{\Phi_{APA}}$ and $\beta_{a_2}^{\Phi_{APCPA}}$, respectively. Since HAN uses the homogeneous graphs extracted based on the meta-paths, it requires manually selected meta-paths. If manual meta-paths are selected wrong, HAN can have worse performance.

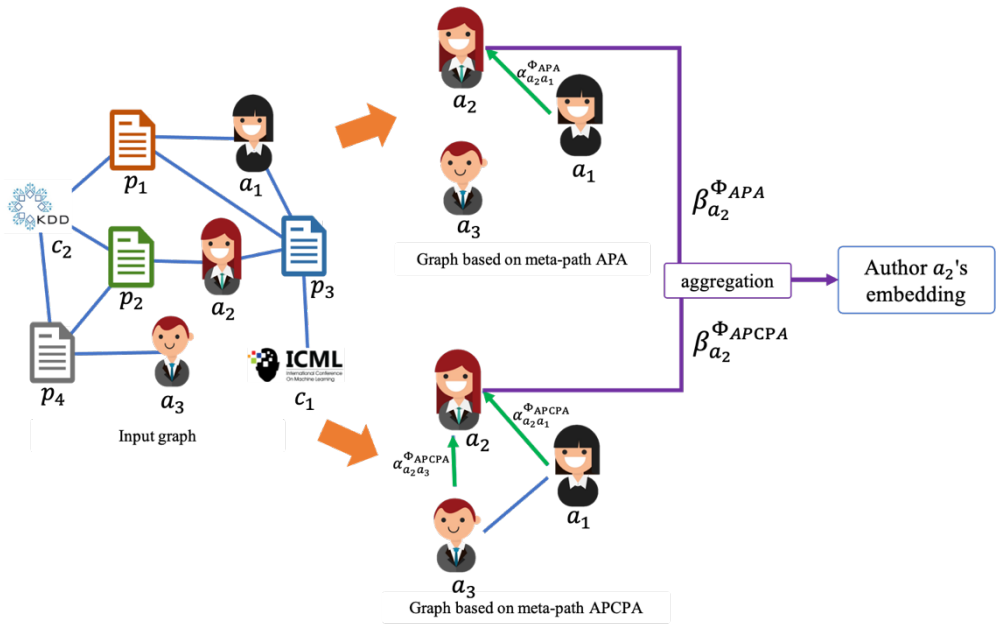


Figure 2: The overall process of HAN.

3.2.3 Graph Transformer Networks(GTNs)

Yun, S. et al. [5] proposed Graph transformer networks(GTNs) which automatically generates possible meta-paths using edge type as learning to produce useful meta-paths. GTN generates meta-paths by multiplying an adjacency matrix of an edge type with an adjacency matrix of another edge type. Figure 3 shows the process of automatically generating meta-paths.

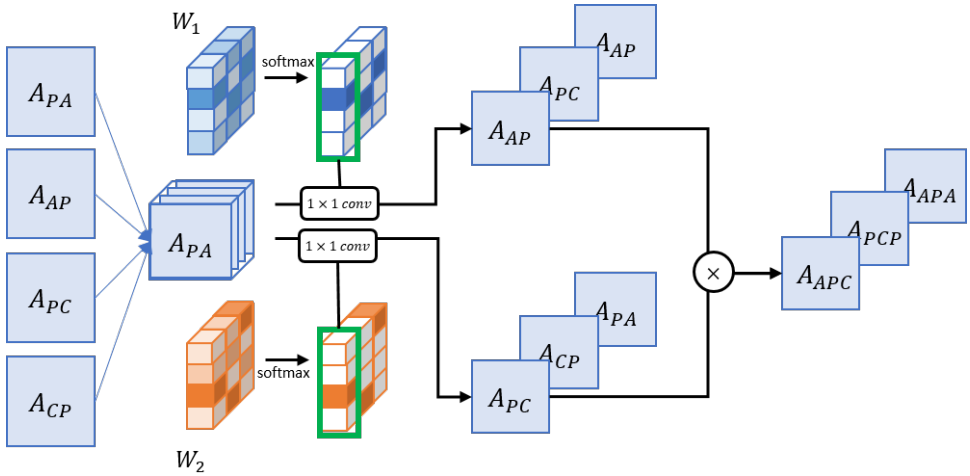


Figure 3: Automatically generating meta-paths in GTN.

As shown in Figure 3, there are four types of edges: Paper-Author(P-A), Author-Paper(A-P), Paper-Conference(P-C), and Conference-Paper(C-P). To select two edge types, GTN conducts a softmax operation for parameters W_1 and W_2. In Figure 3, the edge types A-P and P-C are respectively selected in the green boxes of the results of softmax operation for the parameters W_1 and W_2. GTN gets an adjacency matrix A_{APC} of a meta-path A-P-C by multiplying an adjacency matrix A_{AP} with an adjacency matrix A_{PC}. The number of generated meta-paths is determined by the channel of parameters. For example, in Figure 3, GTN generates three types of

meta-paths with parameters W_1 and W_2 $(W_1, W_2 \in \mathbb{R}^{4\times3})$ that have three channels. When increasing the number of stacked layers, the length of generated meta-path is increased by one. If there are a lot of edge types, the number of possible meta-paths increases exponentially. It can lead to performance degradation to learn by selecting a few meta-paths among a number of possible meta-paths. Also, since meta-paths are generated by matrix multiplication, it can produce heavy computation to generate various meta-paths or long meta-paths.

IV. Exploiting graph patterns

In this section, we propose the heterogeneous graph attention network exploiting triangle patterns (TP-HAN) and exploiting various graph patterns (VP-HAN). We present TP-HAN in section 4.1 and VP-HAN in section 4.2.

4.1 TP-HAN: Exploiting triangle patterns

We focus on that meta-path is a kind of graph pattern. Graph patterns that are the Eulerian path can be represented to meta-paths. For example, triangles in Figure 1(c) can be represented to meta-path Author-Paper-Paper-Author(A-P-P-A). However, more complex graph patterns like a kite pattern and a clique cannot be represented to meta-path or can be represented to long meta-paths. So, we expect that utilizing graph patterns can capture graph structure which is hard to consider with meta-path. We propose a heterogeneous graph attention network exploiting triangle patterns called TP-HAN as the starting point of the hypothesis that using graph patterns leads to performance improvement of GNNs.

Figure 4 shows process of extracting homogeneous graph using a triangle A-P-P-A. Subgraphs, which are matched to a triangle A-P-P-A, are extracted. A homogeneous graph is extracted by considering that two vertices on a triangle are connected. For example, there is a triangle that consists of author a_1, paper p_1, p_3, and

author a_2. Author a_1 and a_2 are connected at extracted homogeneous graph for author vertices.

As extracting homogeneous graph, the adjacency matrices for triangles are calculated by multiplying adjacency matrices of edge types that form a triangle. For example, for a triangle A-P-P-A, when there are three edge types A-P, P-P, and P-A. From a set $\mathrm{E}_{\mathrm{APPA}} = \{t_{AP}, t_{PP}, t_{PA}\}$, TP-HAN gets an adjacency matrix A_{APPA} by multiplying A_{AP}, A_{PP}, and A_{PA}.

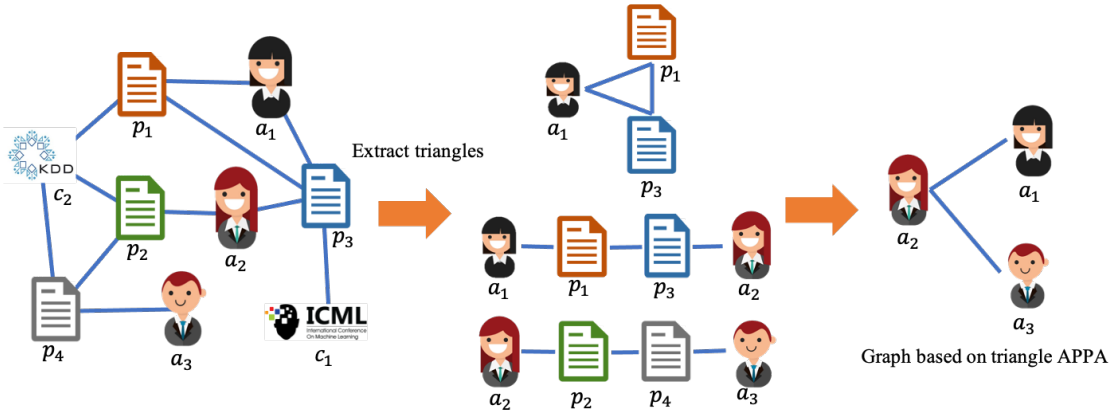


Figure 4: Extracting triangle-based graph.

4.2 Exploiting various patterns

We extend TP-HAN to exploit more complex graph patterns like kite and clique. We expect that exploiting various graph patterns improves the performance. We propose a heterogeneous graph attention network exploiting various graph patterns called VP-HAN. VP-HAN extracts pattern-based graph considering that two vertices on a pattern are connected and aggregates neighborhoods with attention.

Figure 5 shows an example of extracting pattern-based graph using a kite pattern. A kite, which consists of two author a_2, a_3, two paper p_2, p_4 and a conference c_2, is extracted. Author a_2 and a_3 are connected at extracted homogeneous graph for author vertices.

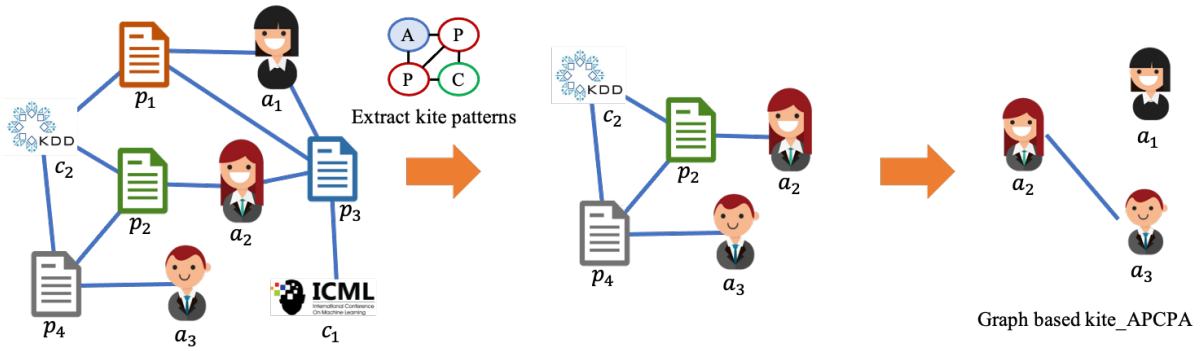


Figure 5: An example of extracting pattern-based graph.

Algorithm 1 shows the pseudo code of our VP-HAN. In line 1-4, VP-HAN extracts the pattern-based homogeneous graphs. In line 5-11, VP-HAN extracts the meta-path based homogeneous graphs. For each pattern p_r, VP-HAN calculates the vertex-level attention between vertices v_i and v_j, and aggregate neighbors $N_{v_i}^{p_r}$. VP-HAN adopts multi-head attention mechanism [2], which consists of K independent attentions. Line 22 represents concatenation of each vertex-level embedding for each attention head. In line 24, VP-HAN calculates the semantic-level attention for each pattern p_r. VP-HAN generates the final embedding Z by aggregating the semantic embedding h^{p_r} with the semantic-level attention β^{p_r}.

Algorithm 1. VP-HAN

Input: $G(V,E)$; /* heterogeneous graph*/
X; /* feature */
Y; /* label */
$\Phi = \{\Phi_1, \Phi_2, \ldots, \Phi_m\}$; /* meta-path set */
$P = \{P_1, P_2, \ldots, P_n\}$; /* pattern set */
K; /* the number of attention heads */

Output: Z; /* the final embedding */

For $P_i \in P$ **do**
  $SG_{P_i} \leftarrow$ find all subgraphs which are monomorphic to P_i;
  $A_{P_i} \leftarrow$ adjacency matrix of homogeneous graph extracted from SG_{P_i};
end for
For $\Phi_i \in \Phi$ **do**
  $A_{\Phi_i} \leftarrow I$;
  $R_{\Phi_i} \leftarrow \{r_j | \Phi_i = r_1 \cdot r_2 \cdot \ldots \cdot r_l, 1 \le j \le l\}$;
  For $r_j \in R_{\Phi_i}$ **do**
    $A_{\Phi_i} \leftarrow A_{\Phi_i} \times A_{r_j}$;
  end for
end for
$A \leftarrow A_P \cup A_\Phi$;
For $A_i \in A$ **do**
  For $k \leftarrow 1$ to K **do**
    For $v_j \in V$ **do**
      Find the pattern-based neighbors $N_{v_j}^{A_i}$;
      For $v_q \in N_{v_j}^{A_i}$ **do**
        Calculate the vertex-level attention $\alpha_{v_j v_q}^{A_i}$;
      end for
      Calculate the vertex-level embedding $h_{v_i}^{A_i} \leftarrow \sigma\left(\Sigma_{v_q \in N_{v_j}^{A_i}} \alpha_{v_j v_q}^{A_i} \times X_{v_j}\right)$;
    end for
    Concatenate the vertex-level embedding $z_{v_j}^{A_i} \leftarrow ||_{k=1}^{K} h_{v_i}^{A_i}$;
  end for
  Calculate the semantic-level attention β^{A_i};
  Fuse the semantic embedding $Z \leftarrow \Sigma_{i=1}^{m+n}(\beta^{A_i} \times h^{A_i})$;
end for
Calculate Cross-Entropy $L = -\Sigma_{l \in Y} Y_l \log(Z_l)$;
Calculate gradients $g \leftarrow \Delta L$;
Update parameters W;
Return Z;

V. Experiment

5.1 Experimental setup

5.1.1 Datasets

To evaluate the effect of utilizing a triangle, we extract the subsets of DBLP [9,11], ACM [12], and IMDB [10]. The detailed statistics of the heterogeneous graphs is shown in Table 2. Figure 6 shows utilized triangles in DBLP and IMDB.

Table 2: Statistics of datasets

Dataset	$\|V\|$			$\|E\|$			Feature	Training	Validation	Test
DBLP	A	4,805	20,049	P-A	7,442	33,382	6,823	800	400	3,605
	P	15,226		P-P	10,714					
	C	18		P-C	15,226					
ACM	A	3,025	9,102	P-A	9,980	18,202	1,902	600	300	2,125
	P	6,021		P-P	5,197					
	S	56		P-S	3,025					
IMDB	M	11,237	31,091	M-M	2,458	31,775	4,467	900	900	9,437
	A	17,401		M-A	25,314					
	D	2,453		M-D	4,003					

DBLP contains three types of vertices (papers(P), authors(A), and conferences(C)) and three types of edges (Paper-Author(P-A), Paper-Paper(P-P), and Paper-Conference(P-C)). Labels are the research area of authors. Author features are represented of the bag-of-words of abstract about papers. Manual meta-paths are meta-path A-P-A and A-P-C-P-A, which were also selected in [4]. In Figure 6(a), there is a triangle A-P-P-A in DBLP.

ACM contains three types of vertices (papers(P), authors(A), and subjects(S)) and three types of edges (Paper-Author(P-A), Paper-Paper(P-P), Paper-Subject(P-S)). Labels are the research field of papers. Paper

features are represented of the bag-of-words of abstract. Manual meta-paths are meta-path P-A-P and P-S-P, selected in [4]. In Figure 6(b), there are three types of triangles: P-P-P-P, P-A-P-P, and P-S-P-P.

The Internet Movie Database(IMDB) contains three types of vertices (Movies (M), Actors(A), Directors(D)) and three types of edges (Movie-Movie(M-M), Movie-Actor(M-A), and Movie-Director(M-D)). Labels are the genre of movies. Movie features are represented of the bag-of-words of plots. Manual meta-paths are meta-paths M-A-M, M-D-M, and M-M-M. Meta-path M-A-M and M-D-M were selected in [4]. In Figure 6(c), there are three types of triangles: M-M-M-M, M-A-M-M, and M-D-M-M.

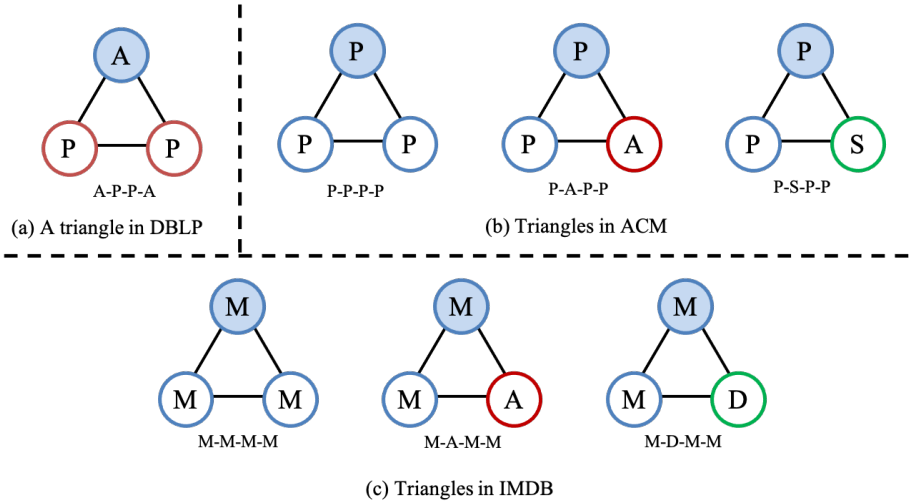


Figure 6: The utilized triangles. (a) A triangle pattern in DBLP. (b) Three types of triangle patterns in ACM (i.e., Paper-Paper-Paper-Paper(P-P-P-P), Paper-Author-Paper-Paper(P-A-P-P), and Paper-Subject-Paper-Paper(P-S-P-P)). (c) Three types of triangle patterns in IMDB (i.e., Movie-Actor-Movie-Movie(M-A-M-M), Movie-Director-Movie-Movie(M-D-M-M), and Movie-Movie-Movie-Movie(M-M-M-M)).

5.1.2 Experimental environment

We conduct the experiment to compare TP-HAN and VP-HAN with HAN. For the experiment, we use a machine with a 40-core 2.2GHz Intel Xeon CPU, 512GB of main memory, and a Tesla V100 GPU. For both

HAN and VP-HAN, we set the learning rate to 0.005, the number of attention heads to 8, the dropout to 0.6, the regularization parameter to 0.001, and the dimension of the semantic-level attention vector to 128. The model is optimized by Adam [7] optimizer.

5.2 Finding meta-paths and various patterns

Figure 6 shows the utilized triangle types for each dataset. We extract the triangle-based homogeneous graphs using adjacency matrix multiplications. To exploit various patterns, we select some graph patterns and extract subgraphs matched to graph patterns using VF2 algorithm [12, 13] for graph isomorphism testing. Since the VF2 algorithm works on the homogeneous graph, VP-HAN finds all subgraphs mapped by graph pattern and additionally categorizes patterns for vertex types. Figure 7 shows an example of extracting kite patterns in IMDB. There are five kinds of kite patterns in IMDB. VP-HAN finds all subgraphs matched to the kite pattern and classifies which the subgraphs belong to kite_MMMM, kite_MAMM, kite_MDMM, kite_MMAM, or kite_MMDM.

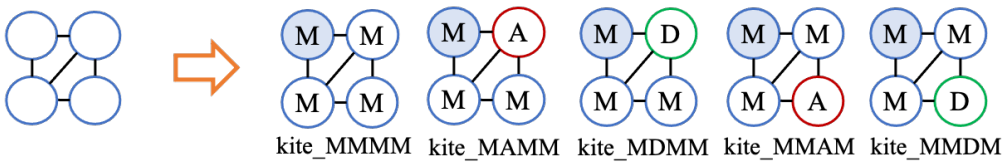


Figure 7: An example of extracting kite patterns.

Figure 8 shows the utilized graph patterns in DBLP and IMDB. Figure 8(a) shows that there are a triangle A-P-P-A, two rectangles A-P-P-P-A and A-P-A-P-A, three kite patterns kite_APPA, kite_APPPA, and

kite_APCPA, and a clique in DBLP. Figure 8(b) shows three triangles M-M-M-M, M-A-M-M, and M-D-M-M, six rectangles M-M-M-M-M, M-A-M-M-M, M-D-M-M-M, M-A-M-A-M, M-D-M-D-M and M-A-M-D-M, five kite patterns kite_MMMM, kite_MAMM, kite_MDMM, kite MMAM, and kite_MMDM in IMDB.

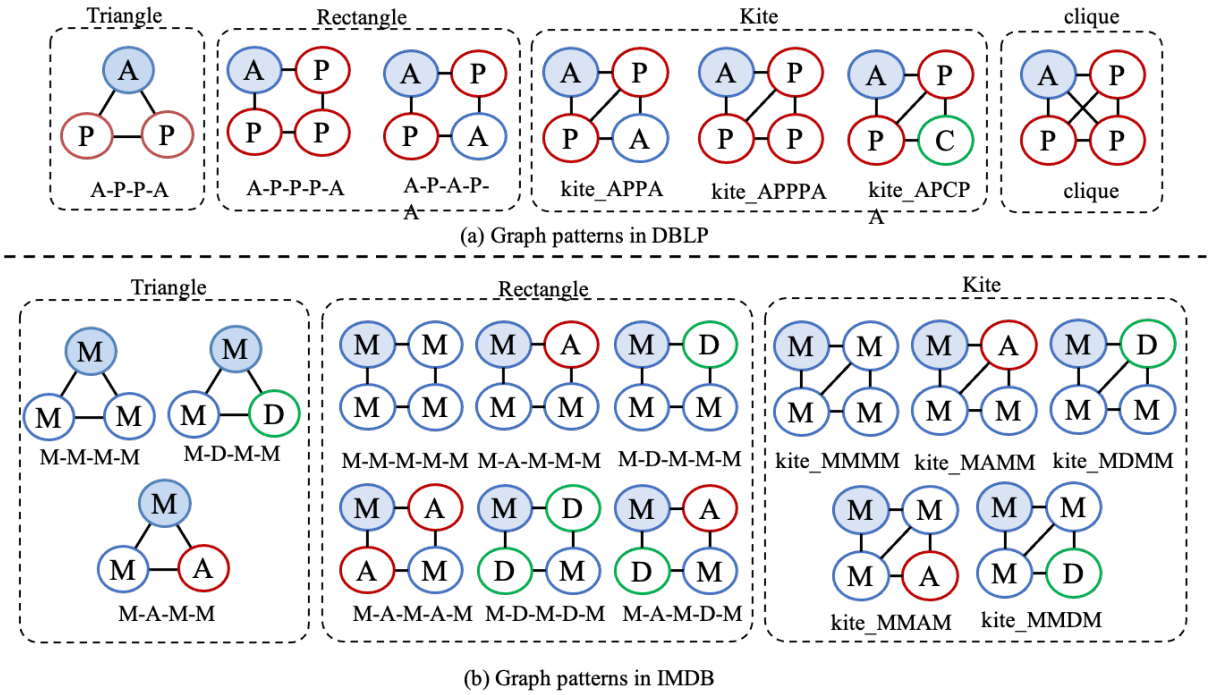


Figure 8: The utilized graph patterns in DBLP and IMDB. (a) Graph patterns in DBLP (i.e., a triangle A-P-P-A, two rectangles A-P-P-P-A and A-P-A-P-A, three kite patterns kite_APPA, kite_APPPA, and kite_APCPA). (b) Graph patterns in IMDB (i.e., three triangles M-M-M-M, M-A-M-M, and M-D-M-M, six rectangles M-M-M-M-M, M-A-M-M-M, M-D-M-M-M, M-A-M-A-M, M-D-M-D-M and M-A-M-D-M, five kite patterns kite_MMMM, kite_MAMM, kite_MDMM, kite MMAM, and kite_MMDM).

Table 3 presents the number of subgraphs matched to graph patterns. The most frequent subgraph is a meta-path A-P-C-P-A, a triangle P-S-P-P, and a rectangle M-A-M-A-M, respectively in DBLP, ACM and IMDB.

Table 3: The number of subgraphs matched to meta-paths and graph patterns

Dataset	Pattern type		The number of sub-graphs
DBLP	Meta-path	A-P-A	11,176
		A-P-C-P-A	4,983,312
	Triangle	A-P-P-A	3,351
	Rectangle	A-P-P-P-A	3,919
		A-P-A-P-A	61,526
	Kite	kite_APPA	161
		kite_APPPA	1,913
		kite_APCPA	1,250
	Clique	clique	180
ACM	Meta-path	P-A-P	40,792
		P-S-P	2,174,333
	Triangle	P-A-P-P	93,540
		P-S-P-P	3,961,869
		P-P-P-P	22,028
IMDB	Meta-path	M-A-M	95,628
		M-D-M	20,203
		M-M-M	18,552
	Triangle	M-A-M-M	46,130
		M-D-M-M	9,105
		M-M-M-M	134,364
	Rectangle	M-M-M-M-M	1,517,240
		M-A-M-M-M	289,569
		M-D-M-M-M	59,424
		M-A-M-A-M	5,420,686
		M-D-M-D-M	529,011
		M-A-M-D-M	377,380
	Kite	kite_MMMM	307,093
		kite_MAMM	3,298
		kite_MDMM	3,590
		kite_MMAM	2,281
		kite_MMDM	938

5.3 Vertex classification

5.3.1 Performance of TP-HAN

TP-HAN exploits both the meta-paths and triangle patterns. We measure the macro F1, the micro F1, precision, and recall for vertex classification to compare performance according to the presence of a triangle.

Table 4 presents the metrics for the vertex classification. In Table 4, TP-HAN using both meta-paths and triangles performs better than HAN using only manual meta-paths in ACM. TP-HAN shows similar or a little better performance to HAN in DBLP and IMDB.

Table 4: Results on the vertex classification task

Dataset	Method	Metrics			
		Micro F1	Macro F1	Precision	Recall
DBLP	HAN	**0.969**	**0.956**	**0.945**	0.968
	TP-HAN	**0.969**	**0.956**	0.943	**0.971**
ACM	HAN	0.917	0.918	0.920	0.918
	TP-HAN	**0.929**	**0.930**	**0.931**	**0.929**
IMDB	HAN	**0.560**	0.502	**0.504**	**0.608**
	TP-HAN	**0.560**	**0.503**	**0.504**	0.607

We compare the performance of TP-HAN by varying the number and type of utilized triangle patterns with the meta-paths as a default. Table 5 presents the micro F1, the macro F1, precision, and recall for the vertex classification depending on the number and type of utilized triangle patterns. TP-HAN performs better for all of the metrics than HAN in ACM. TP-HAN utilizing two triangles M-D-M-M and M-M-M-M achieves the best performance for the micro F1 compared to others in IMDB. TP-HAN exploiting a triangle M-D-M-M obtains better performance for both the macro F1 and precision than others. Regardless of the number and type of utilized triangle patterns, TP-HAN consistently shows better performance than HAN. These results represent that TP-HAN improves the performance compared with HAN by exploiting triangle patterns.

Table 5: Results varying the number of triangle patterns for the classification task

Dataset	Method	Utilized patterns		Metrics			
				Micro F1	Macro F1	Precision	Recall
ACM	HAN	meta-path	P-A-P, P-S-P	0.917	0.918	0.920	0.918
	TP-HAN	A triangle	P-A-P-P	0.929	0.930	0.930	0.929
			P-S-P-P	0.921	0.921	0.922	0.921
			P-P-P-P	0.928	0.929	0.929	0.929
		Two triangle	P-A-P-P, P-S-P-P	0.925	0.926	0.927	0.926
			P-A-P-P, P-P-P-P	**0.930**	**0.931**	**0.931**	**0.930**
			P-S-P-P, P-P-P-P	0.928	0.928	0.929	0.928
		All of triangle	P-A-P-P, P-S-P-P, P-P-P-P	0.929	0.930	**0.931**	0.929
IMDB	HAN	meta-path	M-A-M, M-D-M, M-M-M	0.560	0.502	0.504	**0.608**
	TP-HAN	A triangle	M-A-M-M	0.565	0.504	0.505	0.606
			M-D-M-M	0.567	**0.505**	**0.507**	0.607
			M-M-M-M	0.563	0.503	0.506	**0.608**
		Two triangle	M-A-M-M, M-D-M-M	0.564	0.504	0.505	0.607
			M-A-M-M, M-M-M-M	0.559	0.500	0.504	0.605
			M-D-M-M, M-M-M-M	**0.576**	0.504	0.506	0.595
		All of triangle	M-A-M-M, M-D-M-M, M-M-M-M	0.560	0.503	0.504	0.607

5.3.2 Performance of VP-HAN

We evaluate the performance of VP-HAN while varying the combinations of various graph patterns. There are 127 combinations and 16,383 combinations respectively in DBLP and IMDB. Table 6 shows the performance of exploiting some combinations of graph patterns. VP-HAN consistently performs better for all of the metrics than HAN in both DBLP and IMDB. VP-HAN obtains the best performance exploiting a triangle A-P-P-A, a rectangle A-P-A-P-A, two kites kite_APPA and kite_APCPA, and a clique in DBLP. Also, VP-HAN performs better for the micro F1, the macro F1, and precision than others by exploiting two rectangles M-A-M-A-M and M-A-M-D-M, and three kites kite_MMMM, kite_MDMM, and kite_MMAM. As a result, We confirm that VP-HAN improves the performance of what combination of graph patterns is used rather than exploiting many

graph patterns. Also, we verify that VP-HAN shows that exploiting various graph patterns improves the performance compared with using manually selected meta-paths.

Table 6: Exploiting some combinations of graph patterns for the classification task

Dataset	Method	Utilized patterns		Metrics			
				Micro F1	Macro F1	Precision	Recall
DBLP	HAN	meta-path	A-P-A, A-P-C-P-A	0.969	0.956	0.945	0.968
	VP-HAN (Base + Patterns)	A triangle + a kite	A-P-P-A, kite_APPPA	0.972	0.962	0.953	0.972
		A triangle + a kite	A-P-P-A, kite_APPA	0.974	0.965	0.955	**0.976**
		A rectangle + two kite	A-P-P-P-A, kite_APPA, kite_APPPA	0.974	0.964	0.953	0.975
		A triangle + a rectangle + a kite + a clique	A-P-P-A, A-P-P-P-A, kite_APCPA, clique	0.970	0.958	0.947	0.971
		A triangle + a rectangle + two kite + a clique	A-P-P-A, A-P-A-P-A kite_APPA, kite_APCPA, clique	**0.975**	**0.966**	**0.957**	**0.976**
		Two rectangle + three kite + a clique	A-P-P-P-A, A-P-A-P-A kite_APPA, kite_APCPA, kite_APPPA, clique	0.974	0.965	0.956	0.975
IMDB	HAN	meta-path	M-A-M, M-D-M, M-M-M	0.560	0.502	0.504	0.608
	VP-HAN (Base + Patterns)	A triangle + a rectangle + a kite	M-D-M-M, M-A-M-D-M, kite_MAMM	0.566	0.506	0.507	0.609
		Two triangle + two rectangle + a kite	M-A-M-M, M-D-M-M, M-A-M-A-M, M-A-M-D-M, kite_MMDM	0.570	0.509	0.508	**0.610**
		A triangle + three rectangle + a kite	M-A-M-M, M-A-M-A-M, M-D-M-D-M, M-A-M-D-M, kite_MMAM	0.572	0.510	0.508	0.609
		four rectangle + two kite	M-A-M-M-M, M-A-M-A-M, M-D-M-D-M, M-A-M-D-M, kite_MMMM, kite_MAMM	0.570	0.507	0.507	0.606
		two rectangle + three kite	M-A-M-A-M, M-A-M-D-M, kite_MMMM, kite_MDMM, kite_MMAM	**0.578**	**0.514**	**0.509**	0.608
		Two triangle + three rectangle + three kite	M-A-M-M, M-D-M-M, M-A-M-M-M, M-A-M-A-M, M-A-M-D-M, kite_MMMM, kite_MMAM, kite_MMDM	0.565	0.504	0.505	0.606

5.4 Clustering

We conduct the clustering task to evaluate the embeddings. We utilize K-means clustering algorithm and the number of clusters K is set to the number of classes. We measure normalized mutual information(NMI) and adjusted rand index(ARI) for clustering task.

5.4.1 Performance of TP-HAN

We measure NMI and ARI for clustering task to compare TP-HAN with HAN. Table 7 shows the performance of TP-HAN by varying the number and type of utilized triangle patterns. TP-HAN consistently achieves better performance than HAN. Also, TP-HAN exploiting a triangle P-A-P-P shows the best performance for both NMI and ARI.

Table 7: Results varying the number of triangle patterns for the clustering task

Dataset	Method	Utilized triangle patterns		Metrics	
				NMI	ARI
ACM	HAN	Meta-path	P-A-P, P-S-P	0.716	0.759
	TP-HAN	A triangle	P-A-P-P	**0.745**	**0.791**
			P-S-P-P	0.719	0.764
			P-P-P-P	0.735	0.781
		Two triangles	P-A-P-P, P-S-P-P	0.732	0.781
			P-A-P-P, P-P-P-P	0.742	0.790
			P-S-P-P, P-P-P-P	0.732	0.784
		All of triangles	P-A-P-P, P-S-P-P, P-P-P-P	0.738	0.784

5.4.2 Performance of VP-HAN

We compare VP-HAN with HAN while varying the combinations of various graph patterns for the clustering task. Table 8 represents the performances of VP-HAN exploiting some combinations of graph patterns

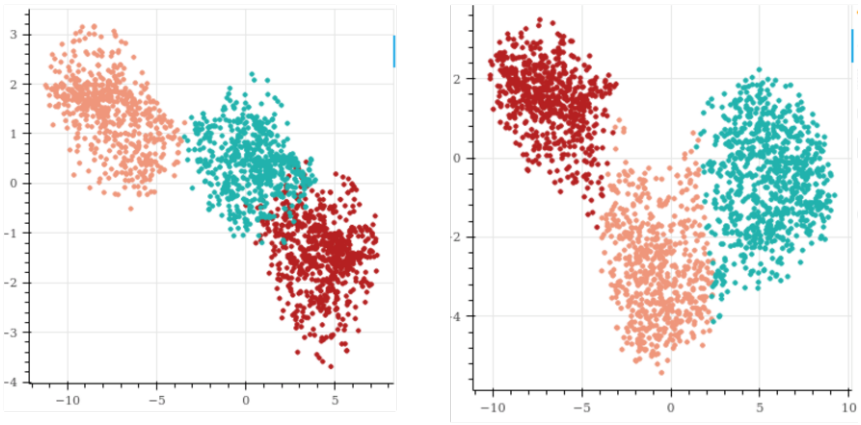
for the clustering task. VP-HAN utilizing a triangles A-P-P-A, a rectangle A-P-A-P-A, two kites kite_APPA, kite_APCPA, and a clique obtains the best performance for clustering task. Like the experimental results on the classification task, these results show that exploiting the specific combination of graph patterns achieves better performance than utilizing many patterns.

Table 8: Results varying the combinations of graph patterns for the clustering task

Dataset	Method	Utilized triangle patterns		Metrics	
				NMI	ARI
DBLP	HAN	Meta-path	A-P-A, A-P-C-P-A	0.868	0.927
	VP-HAN (Base + Patterns)	a clique	clique	0.875	0.930
		A triangle + a kite	A-P-P-A, kite_APPA	0.887	0.938
			A-P-P-A, kite_APCPA	0.879	0.933
		A triangle + a kite + a clique	A-P-P-A, kite_APCPA, clique	0.883	0.934
		A rectangle + a kite + a clique	A-P-P-P-A, kite_APCPA, clique	0.877	0.932
		A rectangle + two kites	A-P-P-P-A, kite_APPA, kite_APPPA	0.884	0.936
		Two rectangles + a kite	A-P-P-P-A, A-P-A-P-A, kite_APPA	0.884	0.934
		Two rectangles + two kites	A-P-P-P-A, A-P-A-P-A, kite_APCPA, kite_APPPA	0.871	0.927
		A triangle + two rectangles + two kites	A-P-P-A, A-P-P-P-A, A-P-A-P-A, kite_APPA, kite_APCPA	0.885	0.937
		A triangle + a rectangle + two kites + a clique	A-P-P-A, A-P-A-P-A, kite_APPA, kite_APCPA, clique	**0.890**	**0.939**
		A rectangle + three kites + a clique	A-P-P-P-A, kite_APPA, kite_APPPA, kite_APCPA, clique	0.884	0.936
		A triangle + two rectangles + two kites + a clique	A-P-P-A, A-P-P-P-A, A-P-A-P-A, kite_APPPA, kite_APCPA, clique	0.876	0.931

5.5 Visualization

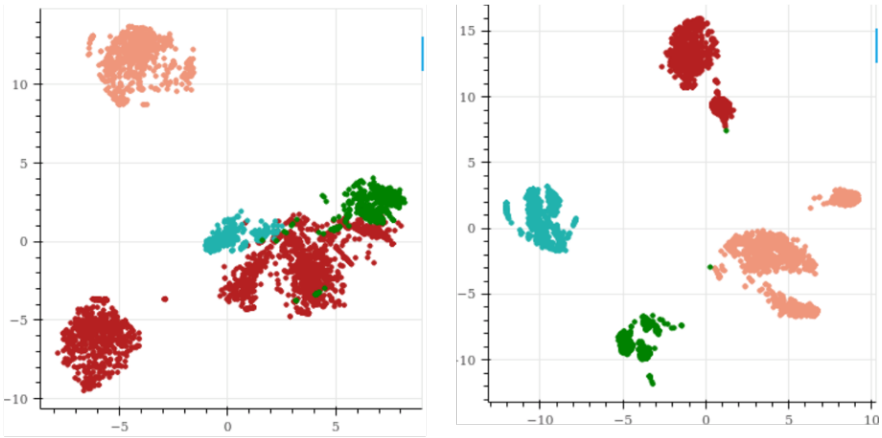
For more intuitive comparison, we utilize t-stochastic neighbor embedding [14] to visualize the author embeddings projected into 2-dimensional space. We visualize the embeddings generated by HAN and TP-HAN exploiting a triangle PAPP which obtains the best performance for clustering task. Figure 9 shows the visualization on HAN and TP-HAN. The boundary is blurry in Figure 9(a). We can see that the visualization of TP-HAN in Figure 9(b) is better than HAN′s.



(a) The visualization of HAN (b) The visualization of TP-HAN

Figure 9: Visualization on ACM. (a) The visualization of HAN. (b) The visualization of TP-HAN.

We visualize the embeddings generated by HAN and VP-HAN exploiting a triangles A-P-P-A, a rectangle A-P-A-P-A, two kites kite_APPA, kite_APCPA, and a clique which achieves the best performance for clustering task. Figure 10 presents the visualizations of embeddings in DBLP. From Figure 10, We can find that the authors belong to the same research area are close together in the visualization of VP-HAN. Also, in Figure 10(b), each cluster is clearly distinct.



(a) The visualization of HAN (b) The visualization of VP-HAN

Figure 10: Visualization on DBLP. (a) The visualization of HAN. (b) The visualization of VP-HAN.

VI. Discussion and Future work

In this paper, we have proposed a heterogeneous graph attention network exploiting triangle patterns and various graph patterns called TP-HAN and VP-HAN, respectively. Both TP-HAN and VP-HAN achieve the improvement of performance compared with HAN. We confirm that exploiting various graph patterns is better than using only meta-paths. Through the experimental results for classification and clustering task, We demonstrate that exploiting the combination of graph patterns performs better than exploiting a number of graph patterns. It means that finding useful combinations of graph patterns is more important than exploiting many of graph patterns.

We expect that finding useful combinations of graph patterns leads to improve the performance of GCN. Therefore, the development of GCN exploiting graph patterns is expected to have the potential addressed as future work. Future work remains to automatically find useful combinations of graph patterns. This direction is expected to overcome the problems such as heavy computation, and the requirement of manual meta-paths.

References

[1] Kipf, T. N. and Welling, M. "Semi-supervised classification with graph convolutional networks." *5th The International Conference on Learning Representations (ICLR) 2017*, Toulon, France.

[2] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y. "Graph attention networks." *6th the International Conference on Learning Representations (ICLR) 2018*, Vancouver, Canada.

[3] Dong, Y., Chawla, N. V., and Swami, A. "metapath2vec: Scalable representation learning for heterogeneous networks." *23rd ACM SIGKDD international conference on knowledge discovery and data mining (KDD) 2017*, Nova Scotica, Canada, 2017, pp. 135-144.

[4] Wang, X., Ji, H., Shi, C., Wang, B., Ye, Y., Cui, P., and Yu, P. S. "Heterogeneous graph attention network." *The World Wide Web conference (WWW) 2019*, San Francisco, United States, 2019, pp. 2022—2032.

[5] Yun, S., Jeong, M., Kim, R., Kang, J., and Kim, H. J. "Graph Transformer Networks." *33rd International Conference on Neural Information Processing Systems (NeurIPS) 2019*, Vancouver, Canada, 2019, vol. 32, pp. 11983-11993.

[6] Hamilton, W. L., Ying, R., and Leskovec, J. "Inductive representation learning on large graphs." *31st International Conference on Neural Information Processing Systems (NeurIPS) 2017*, Long Beach, United States, 2017, pp.1025-1035.

[7] Diederik P. K., and Jimmy B. "Adam: A method for stochastic optimization." *3rd the International Conference on Learning Representations (ICLR) 2015*, San Diego, United States.

[8] Chiang, W. L., Liu, X., Si, S., Li, Y., Bengio, S., and Hsieh, C. J. "Cluster-gcn: An efficient algorithm for training deep and large graph convolutional networks." *25rd ACM SIGKDD international conference on knowledge discovery and data mining (KDD) 2019*, Anchorage, Alaska, United States, 2019, pp. 257-266.

[9] DBLP, https://dblp.uni-trier.de/.

[10] Internet Movie Database, https://www.imdb.com/.

[11] Tang, J., Zhang, J., Yao, L., Li, J., Zhang, L., and Su, Z. "ArnetMiner: Extraction and Mining of Academic Social Networks." *14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD) 2008*, Las Vegas, United States, 2008, pp.990-998.

[12] Cordella, L. P., Foggia, P., Sansone, C., & Vento, M. "A (sub) graph isomorphism algorithm for matching large graphs." *IEEE transactions on pattern analysis and machine intelligence*, 26(10), 2004, 1367-1372.

[13] Cordella, L. P., Foggia, P., Sansone, C., & Vento, M. "An improved algorithm for matching large graphs." *3rd IAPR-TC15 workshop on graph-based representations in pattern recognition 2001*, Ischia, Italy, 2001, pp. 149-159.

[14] Van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of machine learning research*, *9*(11).

요 약 문

이종 그래프 분석을 위한 이종 그래프 어텐션 네트워크 기반의 다양한 그래프 패턴 활용 기법

현실에 존재하는 소셜 네트워크, 인용 네트워크, 단백질 상호작용 네트워크 등과 같은 그래프 타입의 데이터는 다양한 타입의 정점과 간선을 가진다. 그래프 데이터를 분석하기 위한 딥 러닝 기반의 기술로 그래프 컨볼루션 네트워크가 제안되었고 어텐션 메커니즘, 오토인코더, 회귀 신경망 등의 적용을 통해 발전해왔다. 하지만, 기존에 연구된 그래프 컨볼루션 네트워크는 단일 타입의 정점과 간선을 가진 동종 그래프를 대상으로 개발되었으므로 다양한 타입의 정점과 간선을 가지는 이종 그래프를 분석하기에는 적합하지 않다. 이러한 이종 그래프를 분석하기 위해 메타 경로를 이용하는 그래프 컨볼루션 네트워크가 연구되었다. 우리는 메타 경로가 일종의 그래프 패턴이라 보았고, 메타 경로를 이용한 방법을 그래프 패턴을 활용하도록 확장시키고자 하였다. 따라서, 우리는 그래프 패턴 중에서 가장 기본적이고 중요하게 여겨지는 삼각형 패턴을 활용하는 TP-HAN 과 삼각형 패턴을 활용하는 것에서 한단계 더 나아가 다양한 패턴을 활용하는 VP-HAN 을 제안한다. 실험 결과, 우리가 제안한 TP-HAN 은 기존의 HAN 과 비슷하거나 조금 더 나은 성능을 보였고, VP-HAN 은 기존의 HAN 보다 더 나은 성능을 보여 준다.

핵심어: 그래프 컨볼루션 네트워크, 그래프 패턴, 이종 그래프