Graph Neural Networks with Motif-aware for Tenuous Subgraph Finding

HELI SUN, MIAOMIAO SUN, XUECHUN LIU, LINLIN ZHU, LIANG HE, and XIAOLIN JIA, Xi’an Jiaotong University, China
YUAN CHEN, Information Center of Ministry of Science and Technology, Beijing, China

Tenuous subgraph finding aims to detect a subgraph with few social interactions and weak relationships among nodes. Despite significant efforts made on this task, they are mostly carried out in view of graph-structured data. These methods depend on calculating the shortest path and need to enumerate all the paths between nodes, which suffer the combinatorial explosion. Moreover, they all lack the integration of neighborhood information. To this end, we propose a novel model named Graph Neural Network with Motif-aware for tenuous subgraph finding (GNNM), a neighborhood aggregation-based GNN framework that can capture the latent relationship between nodes. We design a GNN module to project nodes into a low-dimensional vector combining the higher-order correlation within nodes based on a motif-aware module. Then we design greedy algorithms in vector space to obtain a tenuous subgraph whose size is greater than a specified constraint. Particularly, considering that existing evaluation indicators cannot capture the latent friendship between nodes, we introduce a novel Potential Friend concept to measure the tenuity of a graph from a new perspective. Experimental results on the real-world and synthetic datasets demonstrate that our proposed method GNNM outperforms existing algorithms in efficiency and subgraph quality.

CCS Concepts: • Theory of computation → Design and analysis of algorithms; Graph algorithms analysis;

Additional Key Words and Phrases: Social network, tenuous subgraph, graph neural network, network motif

ACM Reference format:
Heli Sun, Miaomiao Sun, Xuechun Liu, Linlin Zhu, Liang He, Xiaolin Jia, and Yuan Chen. 2023. Graph Neural Networks with Motif-aware for Tenuous Subgraph Finding. ACM Trans. Knowl. Discov. Data. 17, 8, Article 109 (May 2023), 19 pages.
https://doi.org/10.1145/3589643

The work was supported in part by the National Key RD Program of China 2020YFB1406900 and 2020AAA0107100, and the National Science Foundation of China grants 62072365 and 61772392, the Key Research and Development Program of Shaanxi (Program No. 2020KW-002), the Innovation Capability Support Plan of Shaanxi (Program No. 2021PT-010) and the High-performance Computing Platform of Xi’an Jiaotong University. Any opinions, findings, and conclusions expressed here are those of the authors and do not necessarily reflect the views of the funding agencies.

Authors’ addresses: H. Sun, State Key Laboratory of Communication Content Cognition and Xi’an Jiaotong University, Xi’an, China; email: hlsun@mail.xjtu.edu.cn; M. Sun, X. Liu, L. Zhu, L. He, and X. Jia, Xi’an Jiaotong University, Xi’an, China; emails: miaomiaosun@stu.xjtu.edu.cn, luhan0420@stu.xjtu.edu.cn, zhulinlin@stu.xjtu.edu.cn, lh*@mail.xjtu.edu.cn, xlinjia@mail.xjtu.edu.cn; Y. Chen (corresponding author), Information Center of Ministry of Science and Technology, Beijing, China; email: cheny@most.gov.cn.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.
1556-4681/2023/05-ART109 $15.00
https://doi.org/10.1145/3589643
1 INTRODUCTION

Recently, researchers have focused on the problem of tenuous subgraph finding. Different from dense groups finding \cite{2, 8, 9, 15, 21, 29, 37, 41, 43}, tenuous subgraph finding aims to obtain a subgraph with few social interactions and weak relationships among nodes. It has a wide range of applications and one of the important tasks is reviewer selection. Conference program chairs need to assign experts to review papers. Besides matching the expertise of reviewers with the topics of submissions, it is crucial to avoid assigning reviewers socially close to each other and the authors of the paper to ensure unbiased assessments. The problems of psycho-educational group formation and friend recommendation in weak social networks are also related to tenuous subgraph finding.

Some works have focused on the problem of tenuous subgraph finding. Shen et al. \cite{30} modeled the problem of tenuous subgraph finding as Minimum $k$-Triangle Disconnected Group (M$k$TG), which proposes the concept of $k$-triangle to measure the tenuity of a subgraph. $K$-triangle is defined as a triple in a subgraph, in which the shortest path between any two nodes is less than $k$. The objective of M$k$TG is to get a subgraph in which the number of $k$-triangles is minimized. Li et al. \cite{38} then formulated this problem as $K$-Line-Minimized (KLM), which presents $k$-line to measure the tenuity of a subgraph. $K$-line is defined as a pair of nodes in a subgraph whose shortest path is less than $k$. KLM aims at obtaining a subgraph in which the number of $k$-lines is minimized. Although these methods could solve the problem of tenuous subgraph finding, there remain some concerns. When measuring the shortest path between two nodes, they need to enumerate all the paths between nodes, which brings the combinatorial explosion problem. Thereby, these methods suffer high time cost when computing the number of nodes involved in $k$-triangles or $k$-lines. Moreover, they lack the integration of neighborhood information and may do not work well in some special cases. More concretely, as shown in Figure 1, we may get a tenuous subset $T \in \{1, 2, 7\}$, because the nodes in it do not form any 1-line or 1-triangle. Nevertheless, it is not an optimal subset to form a tenuous subgraph, because these nodes have many common neighbors, which indicates that they may have frequent interaction and be familiar with each other. And we argue that another tenuous subset $T \in \{2, 7, 8\}$ may be a better choice.

Meanwhile, the weights between connected nodes should be adjusted based on the network motifs \cite{23} they participate in. Network motif, also known as the higher-order structure of a network, refers to a small sub-network structure. As shown in Figure 1, node 7 has four neighbors, 3, 4, 5, and 6, and nodes 4, 5, and 7 form a three-node motif. If only considering the neighbor information of nodes, then we could conclude that these nodes have the same influence on node 7. However, the existence of the three-node motif reveals that the relationship between the three nodes 4, 5, and 7 is as close as an iron triangle. Consequently, nodes 4 and 5 have greater influences on node 7.

Motivated by the aforementioned observations, we introduce a new model, Graph Neural Network with Motif-aware for tenuous subgraph finding (GNNM), by taking the above facts into tenuous subgraph finding. Specifically, to integrate high-order structures within the neighborhoods, we first design a motif-aware module to calculate the higher-order correlation between nodes based on three-node motifs and assign weights to different connected nodes. Then, we apply one kind of neighborhood aggregation-based GNNs to explicitly incorporate such correlation into the neighborhood aggregation, which maps the original network to a low-dimensional vector space. Finally, a heuristic algorithm is exploited to get a subgraph whose size is greater than a specified constraint. In terms of evaluation indicators, considering $k$-line or $k$-triangle is limited in some cases, we propose a new concept, Potential Friend (PF), to measure the tenuity of a subgroup. Experimental results demonstrate that the proposed method GNNM outperforms existing algorithms in efficiency and subgraph quality. The main contributions of this article are summarized as follows.
Graph Neural Networks with Motif-aware for Tenuous Subgraph Finding

We propose a novel model for tenuous subgraph finding problem utilizing motif-based network representation. To the best of our knowledge, it is the first attempt to address tenuous subgraph finding problem in the vector space.

We develop a GNN module combining higher-order structural information obtained by the motif-aware module that extracts local features from a node’s neighbors with different weights.

We introduce a new indicator PF to measure the tenuity of a subgraph, which unifies the indicators of k-line and k-triangle and combines both advantages of them, which can discover the latent friendship between nodes.

We conduct extensive experiments to validate the efficiency and effectiveness of our method. We also verify the effectiveness of motif-based network representation.

The rest of this article is organized as follows. Section 2 briefly introduces related work. Section 3 proposes the problem definition. Section 4 presents the framework of our proposed model in detail. In Section 5, we show the experimental results. Finally, Section 6 concludes the article.

2 RELATED WORK

2.1 Tenuous Subgraph Finding

Tenuous subgraphs have many practical significances, e.g., reviewer selection and psychoeducational group formation. Watrigant et al. [3] studied the problem of finding a subgraph of k nodes with the fewest induced edges on the chordal graph. Shen et al. [31] proposed a method for finding a group in which individuals are not familiar with each other. Later, Shen et al. [30] proposed the MkTG problem for finding tenuous subgraphs, which uses the number of k-triangles in the subgraph to measure the tenuity of the subgraph and designs a TERA algorithm to solve the problem. The MkTG problem aims to find a subgraph in which the number of k-triangles is as few as possible, while the number of nodes is as many as possible. A k-triangle is a triplet of nodes \{u, v, w\} ∈ G, such that \(d_G(u, v) \leq k\), \(d_G(u, w) \leq k\), \(d_G(v, w) \leq k\), where \(k\) can be any positive integer, \(d_G(u, v)\) is the shortest distance between \(u, v\) in graph \(G\). Although TERA can find tenuous subgraphs, as mentioned before, k-triangle is limited in some cases. What is more, it needs to calculate the number of k-triangles that each node participates in which leads to high computation. Based on the MkTG problem, Li et al. [38] then proposed the KLM problem for tenuous subgraph finding, which uses k-line to measure the tenuity of a subgraph and designs several algorithms to solve the problem. K-line is a node pair \{u, v\} ∈ G, where \(d_G(u, v) \leq k\). The KLM problem aims to find a subgraph in which the number of k-lines is as few as possible, while the number of nodes is as many as possible. However, k-line is also limited in some cases. Different from the previous work, Hsu et al. [14] proposed the UTNA problem using tenuous subgraph finding to solve the...
problem of group therapy. Group therapy is one of the main clinical methods for the treatment of mental illness, but the formation of the treatment team is quite challenging. UTNA aims to automatically find the largest and reasonable subgraph so that the number of connected edges is as few as possible, and nodes satisfy Noah’s Ark Principle. Li [19] studied the problem of finding tenuous subgraphs in attributed networks that contain specific vertices.

The above-mentioned methods are all based on the graph-structured data to find tenuous subgraphs. They all propose different definitions of tenuous subgraphs, and then design an algorithm based on these definitions. Different from previous methods, in this article, we solve the problem of tenuous subgraph finding by mapping the original graph-structured data into the low-dimensional vector.

2.2 Network Representation

The problem of network representation [5, 45] aims to map each node \( v \in V \) into a low-dimensional vector, i.e., learning a function \( f : v \rightarrow R^h \), where \( h \) is is the dimension of vector and usually far less than \( |V| \). By mapping each node in the network as a low-dimensional dense vector, the topological structure and feature information of the original network can be efficiently stored in the learned latent vector. Then existing machine learning algorithms can be directly used in subsequent graph analysis tasks such as node classification [27], link prediction [12], community detection [20, 32], and so on.

DeepWalk [25] performs a fixed-length random walk and then uses the Skip-Gram model to learn the representation vector of nodes. Node2Vec [11] designs a biased random walk method based on Deepwalk, which is a trade-off between depth-first and breadth-first search. LINE [33] tries to preserve the first-order and second-order similarity in the network. GraRep [6] first obtains a co-occurrence matrix, and then uses SVD to obtain the low-dimensional vectors. TADW [42] is a network representation method for text attribute networks.

More recently, success in extending deep learning to graphs brought about Graph Neural Networks (GNNs) [40, 46]. Being powerful yet efficient, neighborhood aggregation-based GNNs has hence attracted the attention of numerous research works. The majority of GNNs learn the representation vectors of nodes via aggregating and transforming the features within their neighborhoods. Graph autoencoders are one of the methods. GAE [16] is a kind of variational graph autoencoder that replaces the linear layer with the graph convolutional layer. Structural Deep Network Embedding (SDNE) [36] maintains the proximity between 2-hop neighbors through a deep autoencoder. Deep Neural Network for Graph Representations (DNGR) [7] uses a random surfing strategy to capture graph structure information, and then trains stacked denoising autoencoders to embed nodes. GAT [35] introduces self-attention mechanism in propagation process, specifying different weights to different nodes in a neighborhood. SGC [39] aims to reduce the excess complexity of GCNs, making computationally more efficient and fitting significantly fewer parameters.

In this article, we mainly use the graph autoencoder based on graph convolutional network [10, 17] to learn the node representations from graph topological structure and node features information.

2.3 Network Motif

There may be some important high-order structures in the network, which will affect the correlation between nodes. We call these structures network motifs. Network motifs are building blocks in complex networks and some works have proved that motifs are essential to the understanding of higher-order structural information in networks. For example, the existence of a triangle may indicate that the relationship between these three nodes in a social network is as close as a iron
triangle. Researchers have shown great interests in the field of network motif recognition technology. The recognition of motifs in the network has attracted more and more attention, and it has been widely used in social networks [28, 34], biology networks, and so on. Most of the existing works mainly focus on how to count motif subgraphs [13, 24, 26]. There are some works have been proven that motifs can also be used for graph clustering or community detection [1, 44] tasks. While GNNs have achieved remarkable results in a variety of applications, recent studies exposed important shortcomings in their ability to capture the structure of the underlying graph. Some works propose to use motif structure to capture higher-order interactions between nodes in the graph [22]. Lee [18] generalizes GCNs by introducing multiple weighted motif-induced adjacency that capture various higher-order neighborhoods and introduce a novel attention mechanism to choose the best neighborhood for each node to integrate information from. Bouritsas [4] proposed a topologically-aware message passing scheme based on substructure encoding to go beyond isotropic, i.e., locally symmetric, aggregation functions and ensure that GNNs are aware of the structural characteristics of the graph.

Compared with previous studies, we explore the high-order structures based on three-node motifs, and then fuse this high-order relationship into network representation to extract local features from a node’s neighbors with different weights. To the best of our knowledge, it is the first attempt to apply network motif in the tenuous subgraph finding problem to avoid introducing more potential relationship between nodes.

### 3 PROBLEM DEFINITION

In this section, we first give some definitions. Then, we present the formal definition of tenuous subgraph finding. The notations in this article are summarized in Table 1.

To find tenuous subgraphs, we need to define the tenuity of a subgraph first. Considering existing evaluation indicators cannot capture the latent friendship between nodes, we introduce a novel notion \(PF\) for measuring the tenuity of subgraphs.

Given an undirected, unweighted graph \(G = (V, E)\). Let \(N_k(u)\) denote the set of nodes at the shortest distance exactly no more than \(k\) from \(u\) in \(G\). Note that \(N_1(u)\) denotes the set of neighbors of node \(u\).

**Definition 1.** Common Neighbors (\(CN_k\))

For any two nodes \((u, v) \in E\), the \(k\) hop common neighbors of them are defined as follows:

\[
CN_k(u, v) = \{(u, v, w) \mid w \in (N_k(u) \cap N_k(v))\}.
\]

Table 1. Notations

| Symbols | Descriptions                     |
|---------|----------------------------------|
| A       | graph adjacency matrix           |
| X       | node features matrix             |
| N       | number of nodes in graph         |
| M       | co-occurrence matrix             |
| Z       | node representations             |
| T       | tenuous subset                   |
| h       | hidden dimension                 |
| D       | number of node features          |
| A'      | reconstructed adjacency matrix    |
| D'      | diagonal degree matrix           |
| S       | set for defining motif-based adjacency matrix |
| K       | size of tenuous subgraph         |
Note that common neighbors of two nodes used here do not contain the nodes themselves.

**Definition 2.** Potential Friends (PF) Given an undirected, unweighted subgraph $T = (V, E)$, for any two nodes $u$ in $T$, $v$ in $T$, $PF_T$ are defined as follows:

$$PF_T = \begin{cases} \{CN_k(u, v)\} & \text{if } v \notin N_k(u), \\ \{CN_k(u, v) \cup (u, v)\} & \text{otherwise}. \end{cases}$$

(2)

For nodes $u$, $v$, if node $v$ is not the $k$-hop neighbor of node $u$, then $CN_k(u, v)$ is a node triple $(u, v, w)$, and the node $w$ in the triple represents $w$ is the $k$-hop neighbor of these two nodes. While the node doubles $(u, v)$ refer to the situation where the nodes $u$ and $v$ are $k$-hop neighbors to each other. Therefore, $PF_T$ contains both node triples and node doubles. The triples are $CN_k(u, v)$, and the doubles refer to the situation where the nodes $u$ and $v$ are $k$-hop neighbors to each other. $PF_T$ measurement can be seen as an augment of $k$-line and $k$-triangle. By this measurement, we can not only combine $k$-line and $k$-triangle but also capture the potential friends in the obtained subgraph.

Let $L$ denote the set of node pairs that form $k$-lines, and $R$ denote the set of triples in subgraph $T$ that form $k$-triangles, respectively. Suppose $T_1$ represents the set of node pairs whose shortest distance are no more than $k$, and $T_2$ represents the set of rest node pairs in $T$. $L_T$ and $R_T$ are defined as follows:

$$L_T = \{(u, v) \mid (u, v) \text{ forms a } k - \text{line}\},$$

$$R_T = \{(u, v, w) \mid (u, v, w) \text{ forms a } k - \text{triangle}\}. \tag{3}$$

**Theorem 1.** $PF_T$ contains $L_T$, i.e., $L_T \subseteq PF_T$.

**Proof.** For convenience, $PF_T$ can be presented as $PF_T = PF_{T_1} \cup PF_{T_2} = \{CN_k(u, v) \cup (u, v) \mid u \in T_1, v \in T_1\} \cup \{CN_k(u, v) \mid u \in T_2, v \in T_2\}$. Due to $T_2$ represent the set of node pairs whose shortest distance are more than $k$ in $T$, thus $L_{T_2} = \emptyset$. $L_T$ can be presented as $L_T = L_{T_1} \cup L_{T_2} = L_{T_1} = \{(u, v) \mid u \in T_1, v \in T_1\}$. $PF_{T_1} \cap L_T = (PF_{T_1} \cap L_{T_1}) \cup (PF_{T_1} \cap L_{T_2}) = PF_{T_1} \cap L_{T_1}$.

For any node pair $(u, v) \in T_1$, we can find that $L_{T_1} = \{(u, v) \mid u \in T_1, v \in T_1\}$ from formula (3). Since $PF_{T_1} = \{CN_k(u, v) \cup (u, v) \mid u \in T_1, v \in T_1\}$. Thus $PF_{T_1}$ contains $L_{T_1}$, $PF_{T_1} \cap L_{T_1} = \{(u, v) \mid u \in T_1, v \in T_1\}$. In general, $PF_{T_1} \cap L_{T} = \{(u, v) \mid u \in T_1, v \in T_1\}$. Therefore, we can come to a conclusion that $PF_T$ contains $L_T$.

**Theorem 2.** $PF_T$ contains $R_T$, i.e., $R_T \subseteq PF_T$.

**Proof.** Since nodes in $T_2$ cannot form any $k$-triangles, thus $R_{T_2} = \emptyset$. $R_T$ can be represented as $R_T = R_{T_1} \cup R_{T_2} = R_{T_1} = \{(u, v, w) \mid u \in T_1, v \in T_1, w \in T_1\}$. $PF_T \cap R_T = (PF_{T_1} \cap R_{T_1}) \cup (PF_{T_1} \cap R_{T_2}) = PF_{T_1} \cap R_{T_1}$.

For any node pair $(u, v) \in T_1$, first, the shortest distance between $(u, v)$ are no more than $k$; second, $CN_k(u, v)$ are the set of nodes whose shortest distance between $u$ and $v$ are no more than $k$, that is they form a $k$-triangle. Thus $PF_{T_1} \cap R_{T_1} = \{CN_k(u, v) \mid u \in T_1, v \in T_1\}$. In general, $PF_{T_1} \cap R_T = \{CN_k(u, v) \mid u \in T_1, v \in T_1\}$. Therefore, we can come to a conclusion that $PF_T$ contains $R_T$.

Based on the above definitions, we give the formal definition of tenuous subgraph finding problem, which obtains a subgraph by minimizing the number of $|PF_T|$.

**Problem: Tenuous subgraph finding** Given an undirected, unweighted attribute graph $G = (V, E, X)$, size constraint $K$, returns a tenuous subset $T$, where $T$ satisfies the following conditions:

- $|PF_T|$ is minimized,
- $|T| \geq K$,

where $V = \{v_1, v_2, \ldots, v_n\}$ is the vertex set with $n$ nodes in total, $E$ is the edge set, $X = (x_1, x_2, \ldots, x_n)^D$ is the feature matrix, and $|T|$ is the number of nodes in $T$. 

ACM Transactions on Knowledge Discovery from Data, Vol. 17, No. 8, Article 109. Publication date: May 2023.
4 PROPOSED MODEL

In this section, we introduce the framework of proposed model GNNM. As shown in Figure 2, there are three modules in our framework: motif-aware module, graph convolutional autoencoder module, and tenuous subset generation module. Specifically, the motif-aware module helps calculate the correlation between nodes based on network motifs and assigns weights to different adjacent nodes. The graph convolutional autoencoder module learns representations of nodes combining the high-order correlation obtained by the motif-aware layer. The tenuous subset generation part aims to get a node set that meets the conditions by leveraging the node representations.

4.1 Motif-aware Module

Network motifs are building blocks in complex networks and are essential to the understanding of higher-order structural information in networks. This module helps calculate the higher-order correlation between nodes based on three-node motifs and assign weights to different connected nodes.

Definition 3. Three-node motif $A$ three-node motif $M$ is defined on three nodes by a tuple $(A, S)$. $A$ is the adjacency matrix. $S = \{S_1, S_2, \ldots\}$, where $S_j$ is the set of three nodes that participates in a triangle. $S$ denotes a subset of $s$ nodes for defining the motif-based co-occurrence matrix. In other words, two connected nodes $u$ and $v$ will be regarded as co-occurring in a given motif only when they both belong to the same triangle.

We use the motif to capture the higher-order relationships between nodes in a graph when learning the network representations. Following the above definition of the three-node motif, when given a motif set $(A, S)$, we use the co-occurrence of two nodes to capture the corresponding...
higher-order relationships. Assuming that the co-occurrence matrix constructed based on the three-node motif is $M$, then the value corresponding to the matrix $M$ is the number of three-node motifs that nodes participates in. The formula is as follows:

$$M_{ij} = \sum_{k} (i, j) \in E \cap (i, j) \subseteq S_k.$$  \hfill (5)

As shown in Figure 3, there are four nodes in the network. The adjacency matrix is represented as $A$. $S$ is represented as $S_1 \in \{1, 2, 3\}$, $S_2 \in \{2, 3, 4\}$, because node sets $\{1, 2, 3\}$ and $\{2, 3, 4\}$ form a three-node motif, respectively. Nodes 2 and 3 appear in both $S_1$ and $S_2$ and participate together to form two three-node motifs, and there is an edge between them, thereby the value of $M_{23}$ is 3. And the number of three-node motifs formed by node 1 and node 2 is only one, and they are also connected, so the value of $M_{12}$ is 2.

Then, we normalize the weight coefficient as follows:

$$\alpha_{ij} = \sigma_j(m_{ij}) + I = \frac{\exp(m_{ij})}{\sum_{k \in N_i} \exp(m_{ik})} + I,$$  \hfill (6)

where $\sigma(\cdot)$ is the softmax function, $N_i$ are the neighbors of node $i$, $I$ is the identity matrix with the same size as $M$.

### 4.2 Graph Convolutional Autoencoder Module

The graph convolutional autoencoder aims to obtain a low-dimensional representation vector of nodes by fusing the adjacency matrix and feature matrix. It consists of two parts: encoder and decoder. We adopt GCN as the encoder. And the encoder can be represented as

$$Z^{(l+1)} = f \left( Z^{(l)}, A \mid W^{(l)} \right),$$  \hfill (7)

where $W^{(l)}$ is the weight matrix of the $l$th layer, which needs to be learned. $Z^{(l)}$ and $Z^{(l+1)}$ are the input and output of the $l$th layer, respectively.

The nonlinear function $f(\cdot)$ is defined as

$$f \left( Z^{(l)}, A \mid W^{(l)} \right) = \phi \left( D' \left( -\frac{1}{2} \right) A D' \left( -\frac{1}{2} \right) Z^{(l)} W^{(l)} \right),$$  \hfill (8)

where $A' = A + I$ (we assume diagonal elements set to 1, i.e., every node is connected to itself), $D'$ is the diagonal degree matrix, $D'_{ii} = \sum_j A_{ij}$, $\phi$ is the activation function. In this article, the encoder
is constructed with a two-layer GCN. Each layer is represented as follows:

\[
Z^{(1)} = f_{\text{relu}}(X, A|W^{(0)}),
\]
\[
Z^{(2)} = f_{\text{linear}}(Z^{(1)}, A|W^{(1)}),
\]

where \( W^{(0)} \) and \( W^{(1)} \) are the weight matrixes that need to be learned and they are trained using gradient descent. Here, for featureless, we simply drop the dependence on \( X \) and replace \( X \) with the identity matrix in the GCN.

In general, the encoder encodes both structure information and node features into the representation \( Z = \phi(Z \mid X, A) = Z^2 \).

The decoder mainly uses the latent representation vector generated by encoder to reconstruct the graph structure. The reconstructed graph structure can be used to predict whether there is an edge between two nodes. Specifically, let the reconstructed adjacency matrix be \( A' \), the decoder is defined as an inner product between the representation vectors of nodes:

\[
p(A'|Z) = \prod_{i=1}^{N} \prod_{j=1}^{N} p(A'_{ij} | \tilde{z}_i, \tilde{z}_j),
\]

with \( p(A'_{ij} = 1|\tilde{z}_i, \tilde{z}_j) = \sigma(\tilde{z}_i^T \tilde{z}_j) \),

where \( \sigma(\cdot) \) is the sigmoid function.

The loss of graph autoencoder is defined as the reconstruction error between \( A \) and \( A' \):

\[
L = \mathbb{E}_{\phi(z|X,A)} \left[ \log p(A'|Z) \right].
\]

Considering the weights coefficient \( a_{ij} \) calculated by motif-aware module, the final low-dimensional representation vector of nodes can be represented as follows:

\[
Z^{(l+1)} = \sum_{j \in N_i} a_{ij} Z^{(l)}.
\]

To reduce the required memory, we use the sparse representation of adjacency matrix \( A \) as the initial graph representation. So, the memory consumption is \( O(|E|) \).

### 4.3 Tenuous Subset Generation Module

This module leverages obtained nodes representation and design algorithms in vector space. To solve the problem of tenuous subgraph finding, we perform a heuristic algorithm based on local density expansion in low-dimensional vector space.

**Definition 4.** \( \epsilon \)-pair Given the representation vectors of two nodes, we call it a \( \epsilon \)-pair if their distance is not greater than \( \epsilon \). Here, we use Euclidean distance to measure the distance of two nodes in vector space:

\[
\text{dis} (x_i, x_j) = \sqrt{\sum_{t=1}^{h} (x_{it} - x_{jt})^2}.
\]

With the definition of \( \epsilon \)-pair, we can measure the tenuity of a given vector set. Assuming that the number of \( \epsilon \)-pairs in tenuous subset \( T \) is \( r \), the smaller \( r \), the more tenuous the set \( T \).

**Definition 5.** \( \epsilon \)-neighbor(\( N_r \)) Different from the definition of neighbors in the graph as the nodes with edges between them, the vector space has no explicit edge information. However, the distance between any nodes can be easily calculated. These distances reflect the relationship between nodes.
in the original graph. If the distance between two nodes is small, then we consider them as neighbors. Accordingly, for any node $u \in V$, given a distance parameter $\epsilon$, the $\epsilon$-neighbor of $u$ is defined as follows:

$$N_\epsilon(u) = \{v \in V \mid \text{dis}(u, v) \leq \epsilon\}. \quad (15)$$

It can be seen from the formula that for node $u$, those whose distance from node $u$ is less than or equal to $\epsilon$ belong to the $\epsilon$-neighbors of the node. Note that the neighbors do not include the node itself. It is easy to think that for node $u$, the smaller the number of its $\epsilon$-neighbor nodes, the fewer the number of surrounding nodes interacting with the node. This value measures the degree of connection between a node and all other nodes in the graph.

As shown in Figure 4, we draw circles with radius $\epsilon$ centered on node $u_1$ and $u_2$, respectively, and we use the big circle to denote the neighborhood boundary. Nodes in orange within the circle are the $\epsilon$-neighbors of $u_1$ and $u_2$, respectively.

**Definition 6.** density ($\rho(u)$) Intuitively, the number of neighbors represents the degree of interaction between the node and the other nodes. Therefore, the tenuity of a node can be simply expressed as the number of neighbors of it:

$$\rho(u) = \frac{|N_\epsilon(u)| \cap V}{|V|}. \quad (16)$$

Take Figure 4 as an example, we show how to calculate density $\rho$. We can see from Figure 4, node $u_1$ has four $\epsilon$-neighbors and there are 21 nodes in total, thus $\rho(u_1) = \frac{4}{21}$. While node $u_2$ has seven $\epsilon$-neighbors, thus $\rho(u_2) = \frac{7}{21}$.

Next, we propose a heuristic algorithm based on local density in low-dimensional vector space. Empirically, a greater distance constraint $\epsilon$ means a smaller subgraph. In other words, the size of the obtained tenuous subset is negatively related to the distance constraint. If the size of obtained subgraph does not meet the constraint, then we reduce the distance constraint $\epsilon$ to a smaller $\epsilon'$. Therefore, we can perform the binary search on the distance constraint for efficient calculation.

The algorithm first calculates the number of $\epsilon$-neighbor nodes and neighbors of all nodes; then selects an optimal node, adds it to the tenuous subset and deletes all nodes in its $\epsilon$-neighbor; finally, for the rest candidate nodes, iteratively selects the optimal node, which meets the conditions and adds it to the tenuous subset. The algorithm always selects the optimal node in current state when adding nodes to the tenuous subset.

There are two strategies to select optimal nodes, which can be seen as a trade-off between search efficiency and subgraph size.

The former prioritizes the selection of the nodes with largest $\rho$ to join the tenuous subset. Because in this way, the number of remaining optional nodes in the next iteration will be reduced, that is, the number of iterations will be reduced. Thus, this method has high search efficiency. The latter prioritizes the selection of the nodes with smallest $\rho$ to join the tenuous subset. After we
ALGORITHM 1: tenuous subset generation with lower bound on the size

Input: $V = \{v_1, v_2, \ldots, v_n\}$, node vector set $R^h$, size constraint $k$ distance parameter $d$.

Output: tenuous subset $T$.

unvisited set $U \leftarrow V$

tenuous subset $T \leftarrow \emptyset$

for $v_i \in V$ do

- find out the $\epsilon$ neighbor of node $v_i$: $N_\epsilon(v_i)$, and calculate the number of $\epsilon$-neighbors: $|N_\epsilon(v_i)|$.

end

while $|T| < k$ do

- while $U \neq \emptyset$ do

- $v_t \leftarrow$ the node with biggest density

- $T \leftarrow T \cup \{v_t\}$

- delete node $v_t$ and all its neighbors

end

- $d \leftarrow d'$

end

return $T$

select a node $u$ with smallest $\rho$, we need to delete nodes that are the neighbors of $u$. So, we will obtain a subgraph with more nodes that can easily meet the size constraint. Besides, the running time is slightly longer than that of former.

Here, we give the pseudo by prioritizing the selection of the nodes with largest density to join the tenuous subset in Table 1.

**Time Complexity**

When calculating neighbors, by adopting the kd-tree data index structure, the time complexity is $O(N \log N)$. When node is added to the tenuous subset, the linear search method is used to find the node with the smallest number of neighbors, and the time complexity is $O(N)$. The time complexity of deleting nodes in the neighbors of a given node is $O(p)$, where $p$ is the average number of neighbors of each node. In summary, the total time complexity of the tenuous subset discovering process is $O(l(N + p))$, where $l$ is the size of obtained tenuous subsets, usually $l \ll N$. It can be approximated that the time complexity of the tenuous subset generation is nearly linear. Note that the process of calculating neighbors can be done efficiently offline. Therefore, the time complexity of GNNM algorithm is approximately $O(l(N + p))$.

5 EXPERIMENTS AND RESULTS

In this section, we first describe the datasets and then evaluate the efficiency and effectiveness of GNNM with baselines. All algorithms are coded in Python, and all the experiments are implemented on a computer with a 3.20 GHz CPU and 64 GB memory.

5.1 Experimental Setup

5.1.1 Datasets. In the experimental part, we use several real-world datasets and synthetic datasets. The detailed information of these data sets are summarized in the following tables, in which $|V|$ and $|E|$ represent the number of nodes and edges, respectively.

Table 2 depicts four real-world datasets, including biological network, citation network and mail network. The details of these datasets are described as follows.

Hvr: The Hvr dataset is a gene network consisting of several networks of highly recombinant malaria parasite genes. It contains 304 nodes and 3,263 edges.

1https://dl.acm.org/doi/10.1145/3385415.
Table 2. Real-world Datasets

| Dataset | |V| |E| | Average Degree | Max Degree |
|---------|-----------------|-----------------|-----------------|-----------------|-----------------|
| Hvr     | 304             | 3,263           | 21.467          | 61              |
| Cora    | 2,708           | 5,429           | 4.009           | 169             |
| Citeseer| 3,264           | 4,532           | 2.777           | 99              |
| Enron   | 36,692          | 183,831         | 10.020          | 3044            |

Table 3. Synthetic Datasets for Efficiency Analysis

| Dataset      | |V| |E| | Average Degree | Max Degree |
|--------------|-----------------|-----------------|-----------------|-----------------|
| Synthetic_1000 | 1,000           | 1,373           | 2.746           | 5               |
| Synthetic_5000 | 5,000           | 6,875           | 2.750           | 5               |
| Synthetic_10000 | 10,000          | 16,757          | 3.351           | 5               |
| Synthetic_50000 | 50,000          | 69,104          | 2.764           | 5               |
| Synthetic_100000 | 100,000         | 168,026         | 3.360           | 5               |

Cora: The Cora dataset is a paper citation network that contains 2,708 nodes and 5,429 edges. The nodes represent the papers, and the edges represent the citation relationships between papers.

Citeseer: The Citeseer dataset is a citation network that contains 3,264 nodes and 4,532 edges. The nodes represent the publications, and the edges represent the links between publications.

Enron: The Enron institutional mail Network is a commonly used public data set of e-mail, whose mail data is the correspondence of 150 senior executives of Enron in North America. The dataset contains 36,692 nodes and 183,831 edges. The nodes represent mail users, and the edges represent mail exchanges between users.

We also generate several synthetic datasets. The details of these datasets are listed in Tables 3 and 4. The datasets in Table 3 are used for efficiency analysis, while the datasets in Table 4 are used for effectiveness analysis.

5.1.2 Compared Algorithms and Evaluation Metrics. To evaluate the performance of our proposed algorithm GNNM, we use two existing baseline algorithms WK [38] and TERA [30], and the other two GNN methods as baselines in this section.

- TERA [30]: TERA is proposed for MKTG problem that aims to minimize the number of $k$-triangles.
- WK [38]: WK is an improvement of TERA that aims to minimize the number of $k$-lines.
- GAE(GAT) [35]: GAE (GAT) is a kind of variational graph autoencoder that replaces the linear layer with GAT.
- GAE(SGC) [39]: GAE (SGC) a kind of variational graph autoencoder that replaces the linear layer with SGC.

Considering using different kinds of metrics can make the evaluation of the effectiveness of GNNM algorithm proposed by us more convincing.

We use our provided novel metric $PF_1$ and $PF_2$ as well as the existing metrics 1-line and 1-triangle to evaluate the effectiveness of GNNM:

$$k\text{-line} = \sum_{i \in T} \sum_{j \in T} d_G(i, j) \leq k, \quad (17)$$

---

1. H. Sun et al.
2. http://linqs.cs.umd.edu.
3. http://citeseerx.ist.psu.edu.
4. https://www.cs.cmu.edu/~enron.
Table 4. Synthetic Datasets for Effectiveness Analysis

| Dataset    | |V| | |E| | Average Degree | Max Degree |
|------------|---|---|---|---|---|---|---|
| Synthetic_1000 | 1,000 | 1,373 | 2.746 | 5 |
| | 1,000 | 3,066 | 6.132 | 8 |
| | 1,000 | 4,401 | 8.802 | 10 |
| | 1,000 | 5,144 | 10.288 | 12 |
| | 1,000 | 6,873 | 13.746 | 16 |
| | 1,000 | 7,421 | 14.842 | 19 |
| | 1,000 | 8,082 | 16.164 | 19 |
| Synthetic_2000 | 2,000 | 2,341 | 2.341 | 3 |
| | 2,000 | 5,783 | 5.783 | 6 |
| | 2,000 | 8,856 | 8.856 | 10 |
| Synthetic_5000 | 5,000 | 6,875 | 2.750 | 5 |
| | 5,000 | 15,235 | 6.094 | 8 |
| | 5,000 | 22,020 | 8.808 | 10 |
| | 5,000 | 25,595 | 10.238 | 12 |
| | 5,000 | 30,695 | 12.278 | 14 |
| | 5,000 | 36,870 | 14.748 | 17 |
| | 5,000 | 41,840 | 16.736 | 19 |

\[
k\text{-triangle} = \sum_{\forall (u,v,w) \in T} d_G(u, v) \leq k \quad \text{and} \quad d_G(u, w) \leq k \quad \text{and} \quad d_G(v, w) \leq k,
\]

where \(d_G(u, v)\) is the shortest distance between node \(u\) and \(v\) in graph \(G\).

### 5.2 Overall Performance

Table 5 shows the results of the five algorithms on four real-world datasets. We listed the best results in bold and underlined the second best ones. From this table, we have the following observations.

First, the number of \(|PF|\) obtained by WK [38] algorithm is the smallest, because the metric \(|PF|\) is an augment of 1-line and 1-triangle, for the size of target tenuous subgraphs are small, the numbers of 1-line and 1-triangle obtained by WK are both zero. For this reason, we marked the results except this algorithm. But in another way, the result also provided that the metric \(|PF|\) we proposed has more sensitive measurement ability and suitable for the tenuous subgraphs with wider size scale.

Second, the number of \(|PF|\) obtained by GNNM algorithm on Cora and Enron datasets are smallest and on Hvr is second. For the tenuous subgraphs finding task, these three metrics are both with the principles that the values are negatively correlated with tenuity, that is, the values are more small, the performance is better. Therefore, the results show that, by GNNM algorithm, subgraphs with less interaction between nodes and weaker relationships can be obtained. Especially in the Cora and Enron datasets, the advantage of GNNM are more obvious. This is because that the node degree of these datasets are big, the graph structure is more complex, and there are more
Table 5. Results of Different Algorithms on Real-world Datasets

| Dataset | Algorithm | |PF| 1-line | 1-triangle |
|---|---|---|---|---|---|
| Hvr | WK | 12,707 | 332 | 2,377 |
|  | TERA | 2,035 | 66 | 40 |
|  | GAE(GAT) | 1,885 | 43 | 44 |
|  | GAE(SGC) | 2,389 | 58 | 49 |
| Cora | WK | 2,484 | 0 | 0 |
|  | TERA | 3,109 | 366 | 118 |
|  | GNNM | 2,002 | 210 | 43 |
|  | GAE(GAT) | 2,478 | 231 | 47 |
|  | GAE(SGC) | 2,783 | 238 | 50 |
| Citeseer | WK | 1,368 | 0 | 0 |
|  | TERA | 7,344 | 791 | 508 |
|  | GNNM | 5,562 | 502 | 171 |
|  | GAE(GAT) | 5,438 | 528 | 154 |
|  | GAE(SGC) | 5,685 | 594 | 206 |
| Enron | WK | 147,253 | 0 | 0 |
|  | TERA | 9,648,620 | 82,526 | 567,854 |
|  | GNNM | 6,171,096 | 50,390 | 327,414 |
|  | GAE(GAT) | 6,414,154 | 51,429 | 297,421 |
|  | GAE(SGC) | 6,774,526 | 54,856 | 395,452 |

Fig. 5. Running time of different algorithms on synthetic datasets.

potential friends of nodes. This is also the reason why baseline algorithms get subgraphs with larger |PF|.

At last, from the definitions of 1-line and 1-triangle, it can be found that compared with 1-triangle, 1-line is a linear metric with more sensitive measurement ability on big datasets and 1-triangle is more sensitive on small datasets. And |PF| is effective on datasets with different sizes.

5.3 Running Time

In this section, we mainly compare the time efficiency of the proposed algorithm GNNM with TERA [30] and WK [38].

Figure 5 reveals the running time of different algorithms on several synthetic datasets. From Figure 5, we can observe that compared with the other two algorithms, GNNM has the highest time efficiency. And GNNM shows clear advantages of efficiency on Synthetic_50000 and
Graph Neural Networks with Motif-aware for Tenuous Subgraph Finding

Fig. 6. Parameter sensitivity.

Synthetic_100000 when there are enough nodes. This is because WK [38] and TERA [30] methods need to calculate all paths between enumerated nodes to calculate the shortest distance between nodes, which has the problem of combinatorial explosion and low time complexity. As the network scale increases, the combinatorial explosion problem becomes more obvious. Thus, with the increase of network scale, the efficiency of the WK and TERA [30] algorithms becomes low.

We can also find that TERA [30] algorithm has the lowest time efficiency. Because TERA [30] needs to calculate the number of k-triangles that each node participates in, which is time-consuming. On the contrary, by mapping the original graph to the low-dimensional vector space, the time complexity of distance calculation in the vector space is greatly reduced compared with the shortest path calculation in the graph. Accordingly, our method presents superior performance.

5.4 Parameter Sensitivity

In this section, we discuss the effects of the parameter changes.

Figure 6(a) shows the relationship between $\epsilon$ and the number of $|PF|$. Figure 6(b) shows the relationship between $\epsilon$ and the obtained graph size. From Figure 6(a), we find that the number of $|PF|$ obtained by Hvr is the largest, because it is denser than others. As shown in Figure 6(b), when $\epsilon$ becomes large, small graphs are preferred, because larger graphs obtained by dense graphs tend to incur much more potential friends. Moreover, Citeseer can find a larger graph without generating any potential friends, because the average degree of Citeseer is small.

5.5 Ablation Study

In this section, we use three synthetic datasets to evaluate the efficiency and effectiveness of GNNM. We denote the variant of GNNM as Basic, which is not based on motif.

(1) Comparative analysis of GNNM and Basic

This section conducts experiments on relevant datasets to compare and analyze the performance of GNNM, WK [38], TERA [30], and Basic algorithms. WK and TERA are two existing baseline methods without the autoencoder structure. The experimental results are shown in the Table 6.

Table 6 shows the number of $|PF|$ of GNNM, WK [38], TERA [30] and Basic algorithms on synthetic datasets: Synthetic_1000, Synthetic_2000, and Synthetic_5000. These datasets are those with the corresponding maximum degree values of 5, 3, and 8 in Table 4, respectively.

It can be seen from the Table 6 that the GNNM algorithm is better than the Basic algorithm without the Motif module and much better than the WK [38] and TERA [30] algorithm, which are without the autoencoder structure. And the number of $|PF|$ obtained by Basic algorithm is higher than that of GNNM algorithm and the number of $|PF|$ obtained by WK and TERA algorithm are much higher than that of GNNM algorithm, respectively. The Motif module rather than the
autoencoder structure learns the high-order structural relationship between nodes, and integrating this high-order relationship into the node representation module, which can better learn the node vector and capture richer information in the network. The experimental results also demonstrate the effectiveness of using Motif to learn high-order structural information of nodes.

(2) The number of $|PF|$ under different graph degrees

Figure 7 shows the number of $|PF|$ in the obtained tenuous subgraph by Basic and GNNM on the three synthetic datasets, respectively. As shown in the figure, GNNM outperforms Basic. The number of $|PF|$ obtained by GNNM is less than that of Basic. Especially, with the increase of degree, the difference between the two algorithms becomes obvious. This is because the average degree of this network is relatively large, and there are more three-node motifs in the network. GNNM can capture this higher-order structure by network representation based on the three-node motif. Thereby, it can reduce the number of $|PF|$ in the obtained subgraph.

(3) The number of $|PF|$ under subgraphs with different sizes

Figure 8 shows the relationship between the obtained subgraph size and $|PF|$. As shown in the figure, when graph size increases, the number of $|PF|$ also increases. Moreover, the number of $|PF|$ obtained by Basic is larger than that of GNNM. And the difference becomes bigger and bigger with the increase of graph size. The reason is that GNNM uses the motif-aware module to pay more attention to node pairs that participate in more three-node motifs. Such node pairs are more similar with each other and should be avoided to be added to the final tenuous subgraph. Thus, GNNM can reduce the number of the latent triangles in the obtained tenuous subgraph.
5.6 Case Study

To show the performance of GNNM, we conduct a case study on the Aminer dataset, which was extracted and mined from academic social networks. We use this dataset to do reviewer selection and visualize the result. We select one of the subnet from Aminer and show the result in Figure 9. Larger nodes are the ones obtained by GNNM, which form a tenuous subgraph whose names are Daniel C. McFarlane, Robert Gray, Seyed Amir Iranmanesh, Haifeng Chen, Nick Feamster, Isao Nagasawa, Ben Y. Zhao, and Fabian E. Bustamante. Their research fields contain network and system measurement and management. They work in different universities and have less intersections. We can notice from the figure that these nodes are evenly distributed. And we can also see that there are no edges in the tenuous subgraph and the relationship between nodes is very weak. Since we set size constraint $K$ to 8, we only obtain a subgraph with eight nodes. If we increase the value of $K$, then we can find a larger tenuous subgraph.

6 CONCLUSION

In this article, we proposed a novel method for tenuous subgraph finding based on motif-based network representation. Considering the higher-order relationship between nodes based on network three-node motif, we developed a framework that consists of three modules: motif-aware module, graph autoencoder with motif-aware module, and tenuous subset generation module. Extensive experiments are designed to demonstrate the efficiency and effectiveness of the motif-based network representation. The experimental results demonstrate the superiority of our model. Since we just simply use the node features information without guaranteeing that the features of nodes in the tenuous subgraph should be closely related, we will study it in the future.
REFERENCES

[1] Austin R. Benson, David F. Gleich, and Jure Leskovec. 2016. Higher-order organization of complex networks. Science 353, 6295 (2016), 163–166.

[2] Cecile Bothorel, Juan David Cruz, Matteo Magnani, and Barbora Micenková. 2015. Clustering attributed graphs: models, measures and methods. arXiv:1501.01676 [cs.SI]

[3] Marin Bougeret, Nicolas Bousquet, Rodolphe Giroudeau, and Rémi Watrigant. 2014. Parameterized complexity of the sparsest k-subgraph problem in chordal graphs. In Proceedings of the 40th International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM’14), Viliam Geffert, Bart Preneel, Branislav Rovan, Július Štuller, and A. Min Tjoa (Eds.). Springer International Publishing, Cham, 150–161.

[4] Giorgos Bouritsas, Fabrizio Frasca, Stefanos Zaferiou, and Michael M. Bronstein. 2020. Improving graph neural network expressivity via subgraph isomorphism counting. Retrieved from https://arXiv:2006.09252.

[5] HongYun Cai, Vincent W. Zheng, and Kevin Chen-Chuan Chang. 2018. A comprehensive survey of graph embedding: Problems, techniques, and applications. IEEE Trans. Knowl. Data Eng. 30, 9 (2018), 1616–1637.

[6] Shaosheng Cao, Wei Lu, and Qiongkai Xu. 2015. Grarep: Learning graph representations with global structural information. In Proceedings of the 24th ACM International on Conference on Information and Knowledge Management. ACM, 891–900.

[7] Shaosheng Cao, Wei Lu, and Qiongkai Xu. 2016. Deep neural networks for learning graph representations. In Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAA’16). AAAI Press, 1145–1152.

[8] Alessio Conte, Tiziano De Matteis, Daniele De Sensi, Roberto Grossi, Andrea Marino, and Luca Versari. 2018. D2K: Scalable community detection in massive networks via small-diameter k-plexes. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 1272–1281.

[9] U. Feige, D. Peleg, and G. Kortsarz. 2001. The dense k-subgraph problem. Algorithmica 29, 3 (Mar. 2001), 410–421.

[10] Hongyang Gao, Zhengyang Wang, and Shuiwang Ji. 2018. Large-scale learnable graph convolutional networks. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 1416–1424.

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

[12] Sogol Haghani and Mohammad Reza Keyvanzour. 2019. A systemic analysis of link prediction in social network. Artif. Intell. Rev. 52, 3 (2019), 1961–1995.

[13] Guyue Han and Harish Sethu. 2016. Waddling random walk: Fast and accurate sampling of motif statistics in large graphs. Retrieved from https://arXiv:1605.09776.

[14] Bay-Yuan Hsu, Yi-Feng Lan, and Chih-Ya Shen. 2018. On automatic formation of effective therapy groups in social networks. IEEE Trans. Comput. Soc. Syst. 5, 3 (2018), 713–726.

[15] Xie Jierui, Stephen Kelley, and Boleslaw K. Szymanski. 2011. Overlapping community detection in networks: The state of the art and comparative study. Retrieved from https://arXiv:1110.5813.

[16] Thomas N. Kipf and Max Welling. 2016. Variational graph auto-encoders. Retrieved from https://arXiv:1611.07308.

[17] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. Retrieved from https://arXiv:1609.02907.

[18] John Boaz Lee, Ryan A. Rossi, Xiangnan Kong, Eunyee Koh, and Anup Rao. 2019. Graph convolutional networks with motif-based attention. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM’19). ACM, New York, NY, 499–508.

[19] Yang Li, Heli Sun, Liang He, Jianbin Huang, Jiyan Chen, Hui He, and Xiaolin Jia. 2020. Querying tenuous group in attributed networks. Comput. J. 65 (2020), 858–873.

[20] Fanzhen Liu, Shan Xue, Jia Wu, and Chuan Zhou. 2020. Deep learning for community detection: Progress, challenges, and opportunities. Retrieved from https://arXiv:2005.08225.

[21] Fanzhen Liu, Shan Xue, Jia Wu, Chuan Zhou, Wenbin Hu, Cécile Paris, Surya Nepal, Jian Yang, and Philip S. Yu. 2020. Deep learning for community detection: Progress, challenges and opportunities. Retrieved from https://arXiv:2005.08225.

[22] Qingqing Long, Yilun Jin, Guoqie Song, Yi Li, and Wei Lin. 2020. Graph structural-topic neural network. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD’20). ACM, New York, NY, 1065–1073.

[23] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. 2002. Network motifs: Simple building blocks of complex networks. Science 298, 5594 (2002), 824–827.

[24] Ryan A. Rossi Nesreen K. Ahmed, Jennifer Neville, and Nick Duffield. 2015. Efficient graphlet counting for large networks. In Proceedings of the IEEE International Conference on Data Mining, 1–10.

[25] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 701–710.
[26] Ali Pinar, C. Seshadhri, and Vaidyanathan Vishal. 2017. ESCAPE: Efficiently counting all 5-vertex subgraphs. In Proceedings of the 26th International Conference on World Wide Web (WWW’17). International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 1431–1440.

[27] Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. 2019. Dropedge: Towards deep graph convolutional networks on node classification. Retrieved from https://arXiv:1907.10903.

[28] Rahmin Rotabi, Krishna Kamath, Jon M. Kleinberg, and Aneesh Sharma. 2017. Detecting strong ties using network motifs. Retrieved from https://arXiv:1702.07390.

[29] Seyed-Vahid Sanei-Mehri, Apurba Das, and Srikantha Tirthapura. 2018. Enumerating top-k quasi-cliques. In Proceedings of the IEEE International Conference on Big Data (Big Data’18). IEEE, 1107–1112.

[30] Chih-Ya Shen, Hong-Han Shuai, De-Nian Yang, Guang-Siang Lee, Liang-Hao Huang, Wang-Chien Lee, and Ming-Syan Chen. 2022. On extracting socially tenuous groups for online social networks with k-Triangles. IEEE Transactions on Knowledge and Data Engineering 34, 7 (2022), 3431–3446. https://doi.org/10.1109/TKDE.2020.302591

[31] Chih-Ya Shen, Hong-Han Shuai, De-Nian Yang, Yi-Feng Lan, Wang-Chien Lee, Philip S. Yu, and Ming-Syan Chen. 2015. Forming online support groups for internet and behavior related addictions. In Proceedings of the 24th ACM International on Conference on Information and Knowledge Management (CIKM’15). ACM, New York, NY, 163–172.

[32] Heli Sun, Fang He, Jianbin Huang, Yizhou Sun, and Xiaolin Jia. 2020. Network embedding for community detection in attributed networks. ACM Trans. Knowl. Discov. Data 14, 3 (2020), 1–25.

[33] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. 2021. A comprehensive survey on graph neural networks. IEEE Trans. Neural Netw. Learn. Syst. 32, 1 (2021), 4–24.

Received 4 November 2021; revised 11 November 2022; accepted 20 March 2023.