MRSCN: A GNN-based Model for Mining Relationship Strength Changes Between Nodes in Dynamic Networks

Tianbao Wang[1,2], Yajun Yang[1,2(✉)], Hong Gao[3], and Qinghua Hu[1]

[1] College of Intelligence and Computing, Tianjin University, Tianjin, China
{tbwang,yjyang,huqinghua}@tju.edu.cn
[2] State Key Laboratory of Communication Content Cognition, Beijing, China
[3] College of Mathematics and Computer Science, Zhejiang Normal University, Jinhua, China
honggao@zjnu.edu.cn

Abstract. The relationship strength between individuals in the network is an essential task in network analysis. However, existing measures of relationship strength are mostly artificially predefined, which can only reflect the relationship strength from a single perspective. To compensate for this, we propose a novel GNN-based model for Mining Relationship Strength Changes between Nodes in dynamic networks, named MRSCN, which learns the reasonable relationship strength from networks. To verify the effectiveness of our measure on the relationship strength change, we further propose a novel ϵ-drastic group model. We develop two group mining algorithms. We conduct extensive experiments on real-life dynamic networks to evaluate our models. The results demonstrate the effectiveness of the proposed MRSCN model and the drastic group mining method.

Keywords: Relationship strength change · Drastic group · Dynamic networks

1 Introduction

In the real world, the relationships between various entities are constantly changing and can be described by dynamic networks, such as social networks. The graph sequence has been introduced to model the dynamic network. At each time point, the graph snapshot is taken to capture the status of the network. For example, in a mobile communication network, the calls between participants each day are modeled and captured as a graph snapshot.

Investigating relationship strength among individuals in the network is essential in network analysis. In social networks, the relationship strength indicates

X. Wang et al. (Eds.): DASFAA 2023, LNCS 13945, pp. 172–182, 2023.
https://doi.org/10.1007/978-3-031-30675-4_12

the possibility that two people become friends even though they may not know each other at the current time. However, the relationship strength also changes over time. Then it is crucial to study the relationship strength change for dynamic network analysis. Based on the relationship strength change, various drastically changing groups can be mined for human beings to track the significant or abnormal changes in networks. For example, in financial trade networks, gang frauds always occur much more frequently than individual frauds. Money laundering syndicates often evade regulation by making small but multiple transactions, resulting in frequent changes in relationship strength between individuals. Mining groups with relationship strengths changing frequently help regulators detect potential risks in time. Due to the importance of the relationship strength between entities, it is necessary to find a reasonable measure of relationship strength.

Several works [1,4] proposed various metrics to measure the relationship strength between nodes, such as connectivity and common neighbors. However, all these artificially predefined measurements can only reflect relationship strength from a single perspective. For the example in Fig. 1, relationship strengths between nodes of the same color change significantly when different metrics are used. It is necessary to design a reasonable relationship strength model to study the relationship strength change for dynamic networks.

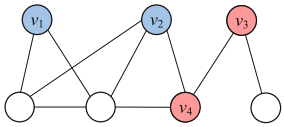


Fig. 1. An illustrative example of existing relationship strength measurements.

In this paper, we propose a novel GNN-based model for Mining Relationship Strength Changes between Nodes in dynamic networks, named MRSCN, which adaptively learns the relationship strength between nodes from networks. Then we further propose the concept of cumulative relational strength change. We also define a drastic group model and investigate the ϵ-drastic group mining problem to verify the usefulness of our measure on relationship strength change. Finally, we conduct extensive experiments to validate the effectiveness of our method.

2 Related Work

In this section, we review related studies on dynamic network analysis, graph embedding, and relationship strength.

As a hot area for researchers in recent years, researchers analyzed dynamic networks from different aspects. Yang et al. [18] developed an algorithm to capture frequently changing components in dynamic networks. Qin et al. [15] proposed a model to mine periodic cliques in dynamic networks. Jia et al. [10]

proposed a method to measure the community consistency. Li et al. [14] developed a new algorithm to find stable communities based on the density-based graph clustering framework. To the best of our knowledge, previous studies have paid little attention to the relationship strength change in dynamic networks.

Graph embedding is used to map nodes to low-dimensional vectors based on network topology. Existing work focuses on preserving structural and attribute information in the embedding, such as DeepWalk [13], Node2Vec [8], DANE [7] and so on. Graph Neural Networks (GNNs) [16], which use a deep learning framework on graph data, have attracted lots of attention. GNNs have been widely used in graph-based problems, such as GCN [12], GAT [17] and GraphSage [9].

Existing research on relationship strength can be divided into two main categories: the first one is based on the network topology to calculate relationship strength, such as common neighbors [14] and connectivity [4]; the second one is a combination of user interactions and user profiles. Guo et al. [11] obtained the fused similarity matrix from different views of user interactions and user profiles. However, there are shortcomings such as artificially predefined or considering structural information from one aspect.

3 Preliminaries

We represent the dynamic network consisting of a sequence of undirected graphs as $\mathcal{G} = \left(G_1, G_2, \cdots, G_{\|\mathcal{G}\|}\right)$, where $G_t = (V_t, E_t)$ is a graph snapshot at time t with a set of vertices V_t and a set of edges E_t, and $\|\mathcal{G}\|$ is the number of G_t in $\mathcal{G}$. $N(u)$ denotes the neighbors of node u. Given nodes u and v, we denote $d(u, v)$ as the distance between u and v, namely the shortest number of hops from u to v. The i-hop neighbors of node u, denoted as $N_i(u)$, contain all the nodes whose distance to u are i, i.e., $N_i(u) = \{v \in V | d(u, v) = i\}$. The i-hop reachable neighbors of node u, denoted as $N_{\leq i}(u)$, contain all the nodes whose distance to u is no more than $i\,(i \geq 1)$. Clearly, $N_{\leq i}(u) = \cup_{j=1}^{i} N_j(u)$.

Relationship strength has no uniform definition and refers to the closeness between individuals. To address the problem of existing predefined models capturing a single relationship, we learn the relationship strength adaptively based on GNN. The relationship strength between u and v in G_t is denoted as $rs_t(u, v)$. Furthermore, the relationship strength change between u and v from G_t to G_{t+1} is defined as $\delta_t(u, v) = |rs_{t+1}(u, v) - rs_t(u, v)|$. Our aim is to learn the reasonable relationship strength measurement and propose the drastic group model to verify the effectiveness of the relationship strength measurement.

4 Mining Relationship Strength Changes Between Nodes

In this section, we introduce our novel MRSCN model in dynamic networks. The overview architecture of MRSCN is shown in Fig. 2. The key idea behind our model is to use the GNN model to mine the relationship strength by comprehensively considering global and local information.

4.1 Global Structure Information Capture

In this section, we use random walk and pointwise mutual information to encode the global structure information.

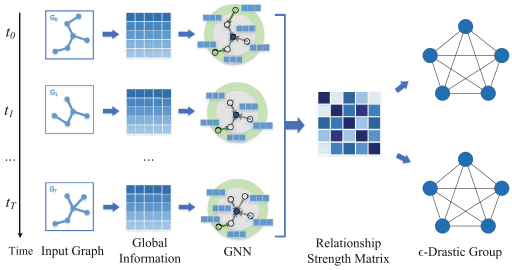


Fig. 2. The overview of the proposed MRSCN.

Calculating Node Co-occurrence Matrix C. We use matrix $\mathbf{C}$ to represent the co-occurrence frequencies between nodes. We use the random walk algorithm here. For each vertex v_i in the graph, we first select it as the starting vertex, which is also the current vertex. We define the current state as $s(t) = v_i$. Then we randomly select the next vertex v_j from the neighbors of v_i. The transition probability of jumping from the current node v_i to v_j is calculated as:

$$p(s(t+1) = v_j | s(t) = v_i) = A_{i,j} / \sum_j A_{i,j}, \tag{1}$$

where $\mathbf{A}$ is the adjacency matrix. Now, we mark this newly selected vertex v_j as the current vertex and repeat such a vertex sampling process. The algorithm terminates when the length of the vertex sequence reaches a pre-set number called walk length η. We repeat the above procedure γ times for each node and record the starting node v_i and node v_j in the sequence for each walk. For each pair (v_i, v_j), We add one to the values of $\mathbf{C}_{i,j}$ and $\mathbf{C}_{j,i}$ respectively, and finally obtain the node co-occurrence matrix $\mathbf{C}$.

Calculating PPMI Matrix M. Pointwise mutual information [5] is often used to measure the correlation of variables. In this paper, we use it to measure the global relationship between nodes. Based on $\mathbf{C}$, we calculate the PPMI matrix $\mathbf{M} \in \mathbb{R}^{n \times n}$ as:

$$m_{i,j} = max\{\log(\frac{p_{i,j}}{p_{i,*}p_{*,j}}), 0\}. \tag{2}$$

Algorithm 1. GraphSage Embedding Generation Algorithm

Input: Graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$; input features $\{\mathbf{f}_v, \forall v \in \mathcal{V}\}$; depth K; weight matrices $\mathbf{W}^k$; aggregator functions AGGREGATE_k; neighborhood function $\mathcal{N} : v \rightarrow 2^{\mathcal{V}}$
Output: Vector representations $\mathbf{z}_v$ for all $v \in \mathcal{V}$
1: $\mathbf{h}_v^0 \leftarrow \mathbf{f}_v, \forall\ v \in \mathcal{V}$
2: **for** $k = 1, \ldots, K$ **do**
3: **for** $\forall v \in \mathcal{V}$ **do**
4: $\mathbf{h}_v^k \leftarrow \sigma(\mathbf{W}^k \cdot \text{MEAN}(\{\mathbf{h}_v^{k-1}\} \cup \{\mathbf{h}_u^{k-1}, \forall u \in \mathcal{N}(v)\}))$
5: **end for**
6: $\mathbf{h}_v^k \leftarrow \mathbf{h}_v^k / \|\mathbf{h}_v^k\|_2, \forall v \in \mathcal{V}$
7: **end for**
8: $\mathbf{z}_v \leftarrow \mathbf{h}_v^K, \forall v \in \mathcal{V}$

The global structure information is encoded by applying Eq. (2). $p_{i,j}$ is the estimated probability that nodes v_i and v_j occur during the random walk at the same time, i.e., $p_{i,j} = \frac{C_{i,j}}{\sum_{i,j} C_{i,j}}$. $p_{i,*}$ and $p_{*,j}$ are the estimated probability that nodes v_i and v_j occur during the walk respectively, i.e., $p_{i,*} = \frac{\sum_j C_{i,j}}{\sum_{i,j} C_{i,j}}$ and $p_{*,j} = \frac{\sum_i C_{i,j}}{\sum_{i,j} C_{i,j}}$. $m_{i,j}$ is the measure of the global correlation of nodes v_i and v_j. As we are focusing on the semantic relation, our method uses a nonnegative pmi. It is worth noting that matrix $\mathbf{C}$ and matrix $\mathbf{M}$ need to be recomputed for different graph snapshots.

4.2 Graph Neural Network Model

To jointly consider the global and local structure information, we integrate global information into the GNN model. GraphSage is used to generate embeddings by aggregating global features from the node's local neighborhood [9].

For each node, the algorithm iteratively aggregates global information from the node's neighbors. The process of aggregating information from neighbors is capturing local information. Algorithm 1 describes the embedding generation process. We use the row vector $\mathbf{M}_{v,:}$ in $\mathbf{M}$ as the node feature $\mathbf{f}_v$ and the inductive variant of the GCN approach as the aggregator function. The final representation of node v is expressed as $\mathbf{z}_v$, as shown in line 8 in Algorithm 1.

In order to learn representations in a fully unsupervised setting, we apply a graph-based loss function:

$$J_{\mathcal{G}}\left(\mathbf{z}_u\right) = -\log\left(\sigma\left(\mathbf{z}_u^{\top}\mathbf{z}_v\right)\right) - Q \cdot \mathbb{E}_{v_n \sim P_n(v)} \log\left(\sigma\left(-\mathbf{z}_u^{\top}\mathbf{z}_{v_n}\right)\right), \tag{3}$$

where v is a node that co-occurs near u on fixed-length random walk, σ is the sigmoid function, P_n is a negative sampling distribution, and Q defines the number of negative samples.

4.3 Relationship Strength Change Computation

After learning the embeddings of nodes in graphs, we aim to obtain the relationship strength. Here, we use the embedding vector obtained by the GNN model as the key feature of the relationship strength change computation.

We select *cosine* similarity to calculate the relationship strength because we are more interested in the directional similarity of different embeddings than in the absolute values. The relationship strength between u and v in G_t, denoted as $rs_t(u, v)$, can be presented as:

$$rs_t(u, v) = \frac{\mathbf{z}_u^{t\top} \cdot \mathbf{z}_v^t}{||\mathbf{z}_u^t|| \times ||\mathbf{z}_v^t||}, \tag{4}$$

The relationship strength change between u and v from G_t to G_{t+1}, denoted as $\delta_t(u, v)$, can be calculated as follows:

$$\delta_t(u, v) = |rs_{t+1}(u, v) - rs_t(u, v)|. \tag{5}$$

Thus, the **cumulative relationship strength change** between u and v in the dynamic network is given as follows:

$$\Delta(u, v) = \sum_{t=1}^{\|\mathcal{G}\|-1} \delta_t(u, v). \tag{6}$$

5 Drastic Group Mining

Based on the relationship strength change between nodes by the MRSCN model, we can further mine drastic groups. In this section, we introduce our *top-k* ϵ-drastic group mining method.

Definition 1. *ϵ-drastic group. A set of nodes $C \subset \mathcal{G}$ is called ϵ-drastic group if the following conditions holds:* (1) $\forall v_1, v_2 \in C$, *cumulative relationship strength change* $\Delta(v_1, v_2) \geq \epsilon$; (2) $|C|$ *is maximized;* (3) $\sum_{v_1, v_2 \in C} \Delta(v_1, v_2)$ *is maximized.*

ϵ-drastic group mining problem has two optimization objectives, namely large coverage and a large sum of relationship strength changes. In reality, they can't be satisfied simultaneously. We consider the problem in the following two cases: (1) priority to make the coverage as large as possible. (2) priority to make the sum of relationship strength changes as large as possible. Based on these two cases, the Coverage-First algorithm (CF) and Strength Change-First algorithm (SCF) are proposed.

First, we need to generate all drastic groups that satisfy $\Delta(v_1, v_2) \geq \epsilon$ for any two nodes v_1, v_2 in the drastic group, which consists of two steps. Firstly, we need to generate a graph containing all the nodes that satisfy $\Delta(v_1, v_2) \geq \epsilon$ for any two nodes v_1, v_2 in the graph. Secondly, we enumerate maximal cliques on the graph to return all drastic groups. Many widely used maximal clique enumeration algorithms can be adopted, such as BasicMCE [6].

Table 1. Datasets statistics.

Dataset	$n = \|V\|$	$m = \|E\|$	$\mathcal{T} = \|\mathcal{G}\|$	Avg. Degree
Chess	7301	65053	25	17.82
Lkml	30665	197356	24	12.87
Enron	66903	189353	14	5.66
P2P-Gnutella	23089	72131	9	6.25

Coverage-First Algorithm. First, all drastic groups are enumerated and sorted in non-increasing order of their coverages. Drastic groups with the same coverage are sorted in non-increasing order of their sums of cumulative relationship strength changes. Second, all sorted drastic groups are scanned sequentially based on the greedy strategy, i.e., the drastic group with the largest coverage is always given priority. Suppose that $\mathcal{K}$ is the current result set and C is the current scanned drastic group that is being decided whether to be added into $\mathcal{K}$ or not. If C overlaps too much with a drastic group C'in $\mathcal{K}$, C might be discarded. In order to evaluate the degree of overlap between C and C', we propose an indicator named the overlap ratio, denoted by $\frac{|C \bigcap C'|}{|C|}$. Given a threshold α, for C, if there exists a C' such that $\frac{|C \bigcap C'|}{|C|} > \alpha$, C will be discarded. Finally, *top-k* ϵ-drastic groups can be gained from the result set $\mathcal{K}$. The time complexity of CF is $O(nlogn)$, where n is the number of drastic groups.

Strength Change-First Algorithm. First, all drastic groups are enumerated and then sorted in non-increasing order of their sums of cumulative relationship strength changes, and drastic groups with the same cumulative relationship strength change are sorted in non-increasing order of their sizes, which indicate the coverages. Second, all sorted drastic groups are scanned sequentially based on the greedy strategy. As with CF, the overlap ratio is used to evaluate whether the drastic group is added to the result set $\mathcal{K}$. We can obtain *top-k* ϵ-drastic groups from the result set $\mathcal{K}$. The time complexity of SCF is the same as CF.

6 Experiments

6.1 Experimental Setup

We conduct experiments on four real-world datasets, including Chess, Lkml, Enron and P2P-Gnutella, whose statistics are summarized in Table 1.

In our experiments, we compare MRSCN with three different categories of methods, including DeepWalk [13], Node2Vec [8], GraRep [2] and DNGR [3].

Since most existing metrics are tailored for traditional graphs, we introduce four goodness metrics evaluating drastic groups for dynamic networks, which are motivated by *separability*, *density*, *common neighbors* and *clustering coefficient*. Let $\mathcal{C}$ be the mining group. The descriptions of evaluation metrics are as follows.

- *Variability of Separability* (VS) captures the intuition that the variability of the separation between groups and the rest of the network is drastic for the group with violent relationship strength changes between nodes: $\mathsf{VS} = \sum_{t=1}^{||\mathcal{G}||-1} |S(G_{t+1}) - S(G_t)|$, $S(G_t)$ is given by: $S(G_t) = \frac{|\{(u,v_1)\in E_t : u\in\mathcal{C}, v_1\in\mathcal{C}\}|}{|\{(u,v_2)\in E_t : u\in\mathcal{C}, v_2\notin\mathcal{C}\}|}$.
- *Variability of Density* (VD) catches the intuition that the group with violent relationship strength changes has drastic connection changes: $\mathsf{VD} = \sum_{t=1}^{||\mathcal{G}||-1} |DS(G_{t+1}) - DS(G_t)|$, $DS(G_t)$ is given by: $DS(G_t) = 2\frac{\sum_{v_j\in\mathcal{C}} d_{\mathcal{C}}^t(v_j)}{|\mathcal{C}|(|\mathcal{C}|-1)}$, where $d_{\mathcal{C}}^t(v_j)$ denotes the degree of v_j in the group $\mathcal{C}$ at t timestamp.
- *Variability of Common Neighbors* (VCN) builds on the intuition that the nodes in the group $\mathcal{C}$ with violent relationship strength changes have drastic changes in common neighbors: $\mathsf{VCN} = \sum_{t=1}^{||\mathcal{G}||-1} |CN(G_{t+1}) - CN(G_t)|$, $CN(G_t)$ is calculated as: $CN(G_t) = \frac{2}{|\mathcal{C}|(|\mathcal{C}|-1)} \sum_{v_i,v_j\in\mathcal{C}} \frac{|N_t^{\leq 2}(v_i,v_j)|}{\sqrt{|N_t^{\leq 2}(v_i)|\times|N_t^{\leq 2}(v_j)|}}$.
- *Variability of Clustering Coefficient* (VCC) is based on the premise that the connection in pair of nodes with common neighbors in $\mathcal{C}$ changes drastically: $\mathsf{VCC} = \sum_{t=1}^{||\mathcal{G}||-1} |CC(G_{t+1}) - CC(G_t)|$, $CC(G_t)$ is given by: $CC(G_t) = \frac{1}{|\mathcal{C}|}\sum_{v_j\in\mathcal{C}} |\frac{2\#edge(N_t(v_j,\mathcal{C}))}{d_{\mathcal{C}}^t(v_j)\cdot(d_{\mathcal{C}}^t(v_j)-1)}|$, where $\#edge(N_t(v_j,\mathcal{C}))$ is the number of edges in $\mathcal{C}$ whose two end nodes are v_j's neighbors in $\mathcal{C}$.

Intuitively, the group with drastic relationship strength changes between nodes should have high VS, VD, VCN and VCC values.

6.2 Experimental Results

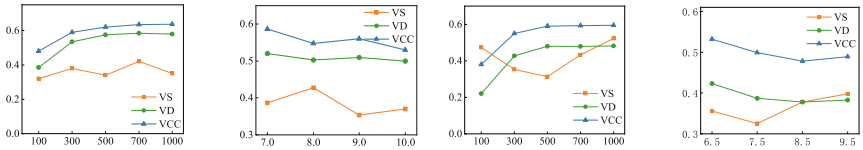
Exp 1. Effectiveness of the ϵ-Drastic Group Mining. In this experiment, we study the effectiveness of the ϵ-drastic group mining. We use the SCF algorithm to evaluate the effectiveness. Table 2 shows the performance of our method compared with other methods.

From the results, we can see that MRSCN performs better than other methods in general. For example, compared to the most powerful compared method DNGR, our MRSCN model reaches nearly 11.1%, 14.8% and 8.1% gain at VD, VCN and VCC, respectively. The experimental results demonstrate that MRSCN has a solid ability to mine relationship strength changes. This is due to the effectiveness of capturing both global structure information and local structure information. In terms of VS, our method has few obvious advantages.

Exp 2. Impact of Parameters. In this part, we analyze the impact of two key parameters in our method, i.e., the embedding dimension d and ϵ. Figure 3 describes the results of our method with varying parameters on VS, VD and VCC in Chess and Lkml, respectively. Similar results can also be observed in the other datasets. VCN follows the same trend as VD and VCC. We first illustrate the performance under various settings of embedding dimension while keeping other parameters fixed, as shown in Fig. 3(a) and Fig. 3(c). We can see that the performance of our method on VD and VCC improves as the embedding size increases and gradually becomes stable when the embedding size increases.

Table 2. Results of the ϵ-drastic group mining.

Methods	Chess				Lkml			
	VS	VD	VCN	VCC	VS	VD	VCN	VCC
DeepWalk	0.255	0.255	0.347	0.512	0.324	0.315	0.341	0.367
Node2Vec	0.281	0.405	0.424	0.508	0.310	0.356	0.385	0.387
GraRep	0.311	0.466	0.403	0.554	0.280	0.349	0.412	0.449
DNGR	0.325	0.468	0.429	0.543	0.314	0.377	0.403	0.436
MRSCN	**0.387**	**0.520**	**0.494**	**0.587**	**0.356**	**0.423**	**0.465**	**0.532**
Methods	Enron				P2P-Gnutella			
	VS	VD	VCN	VCC	VS	VD	VCN	VCC
DeepWalk	0.276	0.208	0.207	0.412	0.257	0.213	0.216	0.336
Node2Vec	0.355	0.261	0.231	0.398	0.290	0.209	0.230	0.342
GraRep	**0.376**	0.254	0.323	0.404	**0.315**	0.218	**0.243**	**0.393**
DNGR	0.314	0.266	0.284	0.436	0.287	0.225	0.207	0.341
MRSCN	0.358	**0.318**	**0.326**	**0.547**	0.266	**0.246**	0.228	0.354



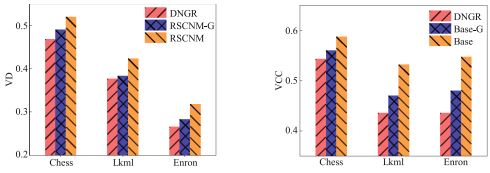
(a) vary d(Chess) (b) vary ϵ(Chess) (c) vary d(Lkml) (d) vary ϵ(Lkml)

Fig. 3. Effectiveness of our method with varying parameters on datasets.

However, we can see from Fig. 3(b) and Fig. 3(d) that both VD and VCC values in different settings of parameter ϵ are irregular, because it cannot guarantee that the total relationship strength change of the group increases with the increase of ϵ.

Exp 3. Ablation Study. To get a better understanding of how different components affect the performance of MRSCN, we conduct ablation tests on three datasets with one variant: MRSCN-G, which removes global features when generating the node embedding. In addition, we choose the best-performing DNGR model as a comparison. The results w.r.t. VD and VCC are shown in Fig. 4.

We can find that the method performs better by combining global features, which shows that capturing global structure information is able to mine relationship strength changes more effectively. Moreover, we find that the basic GNN model works better than DNGR. The experiment demonstrates the effectiveness of mining the relationship strength changes between nodes by jointly considering global and local structure information.



(a) VD (b) VCC

Fig. 4. Performance comparison with variants of our method on three datasets.

7 Conclusion

In this paper, we propose a novel GNN-based model for Mining Relationship Strength Changes between Nodes in dynamic networks, named MRSCN. We use random walk and pointwise mutual information to capture the global structure information. After that, we learn the reasonable relationship strength change by GNN. Based on MRSCN, we propose the ϵ-drastic group model and develop mining algorithms. We conduct experiments on real-world datasets. The results demonstrate the effectiveness of MRSCN and the drastic group mining method.

Acknowledgements. This work is supported by the National Key Research and Development Program of China No. 2019YFB2101903, the State Key Laboratory of Communication Content Cognition Funded Project No. A32003.

References

1. Adamic, L.A., Adar, E.: Friends and neighbors on the web. Soc. Networks **25**(3), 211–230 (2003)
2. Cao, S., Lu, W., Xu, Q.: Grarep: learning graph representations with global structural information. In: CIKM, pp. 891–900. ACM (2015)
3. Cao, S., Lu, W., Xu, Q.: Deep neural networks for learning graph representations. In: AAAI, pp. 1145–1152. AAAI Press (2016)
4. Cho, J.J., Chen, Y., Ding, Y.: On the (co)girth of a connected matroid. Discret. Appl. Math. **155**(18), 2456–2470 (2007)
5. Church, K.W., Hanks, P.: Word association norms, mutual information and lexicography. In: ACL, pp. 76–83. ACL (1989)
6. Eppstein, D., Löffler, M., Strash, D.: Listing all maximal cliques in sparse graphs in near-optimal time. In: Cheong, O., Chwa, K.-Y., Park, K. (eds.) ISAAC 2010. LNCS, vol. 6506, pp. 403–414. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-17517-6_36
7. Gao, H., Huang, H.: Deep attributed network embedding. In: IJCAI, pp. 3364–3370. ijcai.org (2018)
8. Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks. In: KDD, pp. 855–864. ACM (2016)
9. Hamilton, W.L., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. In: NIPS, pp. 1024–1034 (2017)

10. Jia, X., et al.: Tracking community consistency in dynamic networks: an influence-based approach. IEEE Trans. Knowl. Data Eng. **33**(2), 782–795 (2021)
11. Ju, C., Tao, W.: Relationship strength estimation based on wechat friends circle. Neurocomputing **253**, 15–23 (2017)
12. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. CoRR abs/1609.02907 (2016)
13. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: online learning of social representations. In: KDD, pp. 701–710. ACM (2014)
14. Qin, H., Li, R., Wang, G., Huang, X., Yuan, Y., Yu, J.X.: Mining stable communities in temporal networks by density-based clustering. IEEE Trans. Big Data **8**(3), 671–684 (2022)
15. Qin, H., Li, R., Wang, G., Qin, L., Cheng, Y., Yuan, Y.: Mining periodic cliques in temporal networks. In: ICDE, pp. 1130–1141. IEEE (2019)
16. Scarselli, F., Gori, M., Tsoi, A.C., Hagenbuchner, M., Monfardini, G.: The graph neural network model. IEEE Trans. Neural Networks **20**(1), 61–80 (2009)
17. Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks. CoRR abs/1710.10903 (2017)
18. Yang, Y., Yu, J.X., Gao, H., Pei, J., Li, J.: Mining most frequently changing component in evolving graphs. World Wide Web **17**(3), 351–376 (2014)