Triangle-oriented Community Detection considering Node Features and Network Topology

Guangliang Gao[a], Weichao Liang[b], Ming Yuan[a], Hanwei Qian[a], Qun Wang[a], Jie Cao[c,*]

[a]*Department of Computer Information and Cyber Security, Jiangsu Police Institute, Nanjing, China*
[b]*School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China*
[c]*Jiangsu Provincial Key Laboratory of E-Business, Nanjing University of Finance and Economics, Nanjing, China*

Abstract

The joint use of node features and network topology to detect communities is called community detection in attributed networks. Most of the existing work along this line has been carried out through objective function optimization and has proposed numerous approaches. However, they tend to focus only on lower-order details, i.e., capture node features and network topology from node and edge views, and purely seek a higher degree of optimization to guarantee the quality of the found communities, which exacerbates unbalanced communities and free-rider effect. To further clarify and reveal the intrinsic nature of networks, we conduct triangle-oriented community detection considering node features and network topology. Specifically, we first introduce a triangle-based quality metric to preserve higher-order details of node features and network topology, and then formulate so-called two-level constraints to encode lower-order details of node features and network topology. Finally, we develop a local search framework based on optimizing our objective function consisting of the proposed quality metric and two-level constraints to achieve both non-overlapping and overlapping community detection in attributed networks. Extensive experiments demonstrate the effectiveness and efficiency of our framework and its potential in alleviating unbalanced communities and free-rider effect.

Keywords: Community Detection, Graph Clustering, Community Structure, Attributed Network, Quality Metric, Optimization

1. Introduction

Networks provide a natural way to express the complex relationships in our daily life, such as scientific collaborations [1], friend interactions [2], information dissemination [3], and module associations [4]. Typically, the individuals and their relationships in real-world scenarios are represented as the nodes and edges in networks where each node is associated with one or more features characterizing the properties of the individual it corresponds. More formally, we call the networks following the above way of expression as attributed networks [5, 6]. The goal of community detection in attributed networks is to identify the communities hidden inside the networks by utilizing the node features and network topology comprehensively, which not only helps us get a deeper understanding of the network structure, but also gives the people new insights into a series of related issues [7–10].

Over the past several decades, objective function optimization has shown its significance in detecting communities in attributed networks, thus attracting a great deal of attention from numerous researchers [11–16]. Roughly speaking, existing studies in this field can be categorized into two families based on the strategies they employ in integrating node features and network topology. The former kind of approaches address the node features and network topology sequentially. For instance,

SA-Cluster [11], AGGMMR [12], and kNN-enhance [13] first embed the node features into the network topology adopting the way like adding (weighting) nodes or edges, and then deal with the objective optimization problem on the augmented network. In contrast, the latter one handles the node features and network topology simultaneously. For example, CESNA [14], CA-MAS [15], and MOEA-SA [16] first define the objective functions based on the node features and network topology respectively, and then find a balance of optimization through methods such as a trade-off parameter, multi-objective optimization, and non-negative matrix factorization.

Although the above methods have been proven successful in many cases, there remain some issues that deserves careful consideration when performing community detection in attributed networks. Firstly, most of the approaches focus on lower-order details only, i.e., capturing the node features and network topology from the node and edge views. Studies [17–19] have demonstrated that higher-order details, i.e., motifs, are conducive to uncovering the underlying mechanisms of networks. Though there are some attempts [20–22] which detect communities based on motifs, they are often not applicable to attributed networks. Secondly, it is a consensus [5, 6, 11–16] that we wish the nodes belonging to the same community are with dense edges and homogeneous features while the nodes falling into different communities are not. However, it is unwise to focus on nothing but the optimal value of the objective function to achieve the above goal, since it may lead the communities we find suffer from overload, unbalance [23] or free-rider

*Corresponding author
Email address: caojie690929@163.com (Jie Cao)

effect [24]. Thirdly, scalability, ignored by many researchers, is an important factor to consider. Optimization based on global models [14, 25, 26] is computational expensive, especially under the case with large-scale features and complex topology. Precisely for this reason, many of the existing approaches are impossible to be used for many tasks.

In this study, we still follow the principle of objective function optimization and try to improve the quality of the found communities through addressing the above-mentioned issues appropriately. Specifically, we first introduce a quality metric to preserve the higher-order details of the node features and network topology by making a trade-off between them in accordance with the information from the network. Then, we define two constraint items to factor into the lower-order details of the node features and network topology for the purpose of alleviating the unbalance and free-rider effect which may occur in the found communities. Finally, an optimization scheme is designed from a local perspective to make our method have high efficiency. The specific contributions of our work are listed as follows:

- We propose a parameter-free quality metric based on the concept of closed topology and feature triangles, which not only evaluates the quality of higher-order structures, but can also be treated as an optimization objective to achieve triangle-based community detection.

- We formulate the so-called two-level constraints from the node and edge perspectives to enhance the capability of the proposed metric as an optimization objective, which further improves the topological tightness and feature homogeneity of each community found.

- We develop a local search framework based on optimizing our objective function consisting of the proposed metric and two-level constraints, which effectively and efficiently reveals both non-overlapping and overlapping community structures in attributed networks.

The remainder of this study is organized as follows. We review the related work in Section 2, and Section 3 formulates the problem of community detection in attributed networks and illustrates the quality metric referenced in this study. A detailed description of our methodology is presented in Section 4, and Section 5 shows our experimental results. Finally, we conclude this study and give guidelines for our future work in Section 6.

2. Related Work

Since node features and network topology are two different kinds of information, community detection in attributed networks aims to make them complement each other to identify high quality communities. Existing approaches that follow objective optimization can be divided into three groups. In this section, we give a brief review of their recent advances and then discuss how our study differs from theirs.

The first group is node-oriented approaches [11, 27–30], which focus on formulating various distance or similarity functions to find communities through attributed network clustering. For instance, Zhou et al. [11] developed a clustering framework SA-Cluster, which uses a unified distance metric to measure both topological and feature similarity [31], and follows the clustering process of K-medoids. Xu et al. [27] transformed attributed network clustering into a standard probabilistic inference problem based on a defined Bayesian probabilistic model [32] and proposed a variational algorithm to solve it, which avoids the artificial design of distance functions. Bu et al. [28] formalized attributed network clustering as a dynamic cluster formation game, and found a balanced solution by designing the feasible action set, the utility function, and the self-learning strategy for each node. Xu et al. [29] first built an attributed network embedding framework and adopted a distributed algorithm to obtain the embedding vector of each node, and then automatically determined the number of found communities based on curvature and modularity. Finally, the community detection results are obtained by clustering the embeddings. Cao et al. [30] proposed an NMF-based model combining node features and network topology, which employs graph regularization to penalize the dissimilarity of nodes and introduces so-called K-near neighbor consistency to recover feature information.

The second group is edge-oriented approaches [12, 33–36]. Since community detection considering network topology has been widely studied [37, 38], an intuitive strategy is to enhance the network topology with node features, and then find communities based on the enhanced topology. For instance, Smith et al. [33] introduced node features into the description of information flow, and then fine-tuned the Infomap algorithm to find communities with large information flow among nodes and similar node features. Malhotra and Chug [34] designed four variants based on the label propagation algorithm that exploit node features and edge strength to improve the quality of found communities and overcome the random community allocation problem. Berahmand et al. [35] also considered the label propagation algorithm. First, a weighted network combining node features and network topology is generated. Then the influence of each node is calculated using Laplacian centrality, thus enhancing the update path of community labels. Zhe et al. [12] developed a three-stage framework AGGMMR consisting of augmented graph construction and weight initialization, weight learning with modularity maximization, and modularity refinement to find communities in attributed networks. Xie et al. [36] defined a scoring function to check the properties and influence of communities and developed two community search algorithms by maximizing it, and then designed a graph refining algorithm and pruning rules to ensure search efficiency.

The third group is motif-oriented approaches [20–22, 39–42]. Motifs lie between the microscopic proximity structure and mesoscopic community structure and help find communities that maintain building blocks in the network. Recent studies [20–22, 39, 40] usually only consider network topology and rarely involve node features. For instance, Huang et al. [21] studied the motif-based graph partitioning (MGP)

problem. First, a sampling-based MGP (SMGP) framework is designed, which employs an unbiased sampling mechanism to estimate edge weights. Furthermore, an adaptive sampling framework SMGP+ is proposed, which adaptively adjusts the sampling distribution and iteratively partitions the input graph based on up-to-date estimated edge weights. Sotiropoulos and Tsourakakis [39] demonstrated the power edge triangle counts for spectral sparsification and advanced the understanding of triangle-based graph partitioning by empirically analyzing two heuristics for community detection. Li et al. [40] proposed an edge enhancement method for motif-aware community detection in purely topological networks. For attributed networks, they [41] first formulated an AHMotif adjacency matrix to encode node features and network topology from a higher-order perspective, and then utilized proximity-based methods to find communities. Hu et al. [42] composed tensors to model higher-order patterns in terms of node features and network topology, and developed a novel algorithm to capture these patterns to find communities.

Although community detection in attributed networks through objective optimization has been extensively studied, our study is significantly different from most of the existing work in the following respects. First, we consider both node features and network topology based on observations of the nature of networks rather than pursuing an elaborate integration strategy. Second, we exploit node features and network topology from both higher- and lower-order perspectives, namely closed topology and feature triangles, and two-level constraints. Third, our proposed local search framework is suitable for different community detection tasks. To sum up, our study guarantees the robustness of the process of community detection and shows better effectiveness and efficiency.

3. Preliminaries

In this section, we first formulate the problem of community detection in attributed networks and give the notations used throughout this study. Then, we introduce the quality metric weighted community clustering (*WCC*).

3.1. Problem Formulation

Consider representing an attributed network as an undirected and unweighted graph $G = (V, E, F)$, where $V = \{v_1, v_2, ..., v_n\}$ is a set of n nodes, $E \subseteq V \times V$ is a set of m edges that connect two nodes of V, and $F = \{\mathbf{f}_1, \mathbf{f}_2, ..., \mathbf{f}_n\}$ is a set of feature vectors associated with the nodes in V. For any node $v_i \in V$, the neighborhood of v_i is the set $N(v_i) = \{v_j \in V \mid (v_i, v_j) \in E\}$, the degree of v_i is defined as $d_i = |N(v_i)|$, and $\mathbf{f}_i \in \mathbb{R}^{1 \times p}$ denotes the feature vector of v_i, where p is the dimension of the feature vector. We also use the adjacency matrix $\mathbf{A} \in \{0, 1\}_{n \times n}$ and the node feature matrix $\mathbf{B} = \{B_{ij}\} \in \mathbb{R}^{n \times p}$ to represent the topology and the attributes of graph G, respectively. Thus, an edge $(v_i, v_j) \in E$, $A_{ij} = A_{ji} = 1$; otherwise, $A_{ij} = A_{ji} = 0$. If the jth feature is presented in $\mathbf{f}_i$ of node v_i, then $B_{ij} \in (0, 1]$; otherwise, $B_{ij} = 0$. Note that our discussion is not limited to binary features, but also continuous-valued features.

Community detection in graph G aims to find a partition $C = \{C_1, C_2, ..., C_K\}$ of its nodes such that $V = \bigcup_{k=1}^{K} C_k$ and a certain balance between the following two objectives is achieved:

Tightness, i.e., a group of nodes have a high density of edges within them, and a lower density of edges between groups.

Homogeneity, i.e., a group of nodes have similar feature values within them, and may have diverse feature values between groups.

3.2. Quality Metric: WCC

In general, the probability of closed topological triangles among nodes in the same community is larger than the expected among nodes in different communities. *WCC* [43, 44] is inspired by this to measure the quality of a partition of nodes. Given the topology (V, E) of graph G, the degree of belonging of a node v_i to a community C_k, namely $WCC(v_i, C_k)$, is defined as follows:

$$
\begin{cases}
\dfrac{t(v_i, C_k)}{t(v_i, V)} \cdot \dfrac{vt(v_i, V)}{|C_k - \{v_i\}| + vt(v_i, V - C_k)}, & t(v_i, V) \neq 0; \\
0, & t(v_i, V) = 0;
\end{cases}
\tag{1}
$$

where $t(v_i, C_k)$ and $t(v_i, V)$ mean the number of closed topological triangles composed of node v_i and the nodes in C_k and V, respectively. $vt(v_i, V)$ and $vt(v_i, V - C_k)$ mean the number of nodes in V and $V - C_k$ that form at least one closed topological triangle with node v_i, respectively. $|C_k - \{v_i\}|$ means the size of C_k except node v_i.

Then, the *WCC* score of a community C_k is defined as follows:

$$
WCC(C_k) = \frac{1}{|C_k|} \sum_{\forall v_i \in C_k} WCC(v_i, C_k).
\tag{2}
$$

Finally, for a partition $C = \{C_1, C_2, ..., C_K\}$, the *WCC* score is defined as follows:

$$
WCC(C) = \frac{1}{|V|} \sum_{k=1}^{K} (|C_k| \cdot WCC(C_k)).
\tag{3}
$$

It is obvious that this metric is suitable for both non-overlapping and overlapping community detection, and a higher score means better community structure found.

4. Methodology

In this section, we first extend the *WCC* metric by introducing the concept of closed feature triangles. Then, we describe the proposed tightness and homogeneity constraints, which can improve the capability of the extended *WCC* as an optimization objective. Finally, we design a local search framework to achieve both non-overlapping and overlapping community detection by maximizing the objective function consisting of the extended *WCC* and the above constraints.

Table 1: Statistics of Edges in Closed Feature Triangles in *Facebook* and *Sinanet* Ground-truths

Network	Total number of triangles[1]	Number of triangles with different number of edges			
		no edge	one edge	two edges	three edges
Facebook	30, 738, 546	13, 748, 452	10, 110, 887	3, 803, 458	3, 075, 749
Sinanet	138, 407, 278	137, 037, 199	1, 307, 192	60, 282	2, 605

[1] Those closed feature triangles are formed by nodes belonging to the same community.

Table 2: Number of Closed Topology and Feature Triangles in *Facebook* and *Sinanet* Ground-truths

Network	Type of triangle	Number of triangles	
		ground-truths[1]	all communities[2]
Facebook	Topology	1, 209, 670	1, 125, 137
	Feature	1, 007, 762, 916	30, 738, 546
Sinanet	Topology	35, 882	5, 915
	Feature	208, 039, 606	138, 407, 278

[1] Those nodes that form the triangle are in the ground-truths.

[2] Those nodes that form the triangle belong to the same community in the ground-truths.

4.1. Extension of WCC

Triangles, as fundamental paths and motifs that recur in real-world networks, could be used to define and identify communities and more general classes of nodes. *WCC* only focuses on the properties of communities from the perspective of closed topological triangles. In fact, there is a consensus from extensive research [5, 6] that node feature information can be treated as a supplement to explain the formation mechanism of communities as it becomes available. Hence, we define the closed feature triangle based on the intuition of *WCC* as follows:

Definition 1. *A triangle is called the **closed feature triangle** if there exists at least one feature dimension such that for any three nodes $v_x, v_y, v_z \in V$, the following condition is satisfied:*

In term of binary features, we directly compare the feature values, i.e., $\exists l \in \{1, 2, ..., p\}$, $B_{xl} = B_{yl} = B_{zl} = 1$.

In term of continuous-valued features, we intuitively consider the feature dimension with the largest value, i.e., $\exists l \in \{1, 2, ..., p\}$, $\max B_{xl} \neq 0 \land \max B_{yl} \neq 0 \land \max B_{zl} \neq 0$.

To better illustrate our definition of closed feature triangles, we select two real-world networks, *Facebook* and *Sinanet*, for empirical analysis. Their node feature types are binary and continuous-valued, respectively. More detailed descriptive information about them will be introduced in the experimental section.

As shown in Tables 1 and 2, in term of *Facebook*, the ratio of the number of closed topological triangles within all communities in ground-truths to the total number of closed topological triangles in ground-truths is as high as 93%, which means that considering closed topological triangles is helpful to improve the performance of community detection. As for closed feature triangles, the number is huge because the probability of formation is much larger than that of closed topological triangles. However, closed feature triangles are not all conducive to community detection. We further examine the number of edges in closed feature triangles within all communities. The results

show that most of closed feature triangles contain only zero or one edge, and the number of closed feature triangles containing two or three edges is roughly in the same order of magnitude as the number of closed topological triangles. This inspires us to construct closed feature triangles around adjacent nodes, which also seems to be more in line with the fact that communities are mesoscopic structure [45].

In term of *Sinanet*, its feature type is continuous, and we count the number of closed feature triangles based on Definition 1. It is not difficult to find from Tables 1 and 2 that the number of closed topological triangles within all communities is relatively small, while the number of closed feature triangles is still huge. This further reflects the significance of node features as a supplement. Furthermore, the conclusions presented on the number of edges within closed feature triangles are consistent with *Facebook*, which demonstrates that our definition is appropriate. To sum up, we believe that the influence of closed feature triangles formed by adjacent nodes is equal to that of closed topological triangles, and rewrite $WCC(v_i, C_k)$ to $WCC^*(v_i, C_k)$ as follows:

$$\begin{cases} \dfrac{tf(v_i, C_k)}{tf(v_i, NC)} \cdot \dfrac{vtf(v_i, NC)}{|C_k - \{v_i\}| + vtf(v_i, N(v_i))}, & tf(v_i, NC) \neq 0; \\ 0, & tf(v_i, NC) = 0; \end{cases}$$
(4)

where $NC = N(v_i) \cup C_k$ represents a set consisting of the neighbors of a given node and nodes within the candidate community. $tf(\cdot, \cdot)$ represents the number of closed topological and feature triangles, and $vtf(\cdot, \cdot)$ represents the number of nodes forming at least one closed topological or feature triangle. The rest have been explained before and will not be repeated here.

4.2. Two-Level Constraints

Intuitively, if only the network topology is considered, a good community should ensure that internal links are as dense as possible, or form more closed triangles. Extensive research [46, 47] proposes quality metrics such as modularity Q [48] and normalized cut [49] based on such a principle, and achieves community detection by maximizing or minimizing the value of the given quality metric. However, it is not true that the optimization of quality metrics always performs satisfactorily, e.g., the resolution limit of modularity [50], the unbalanced scale of communities [23], the free-rider effect [24]. Although the value of the corresponding quality metric is optimal, the natural community structure of a network is not clearly revealed.

In addition, in network representation, node features are the most important dimension besides the topology structure. Those approaches [5, 6] that utilize node features to detect communities usually formulate distance or similarity functions,

or consider feature embeddings to evaluate the differences between node features. As a result, nodes in the same community share more features. However, each node has multiple features, not all of which are helpful in determining the community a node belongs to. In some cases, node features even provide contradictory information with topological structure [6]. Moreover, existing evaluation methods are generally difficult to apply to different feature types, and too many features also affect the efficiency of evaluation.

Hence, to identify more natural communities from an attributed network, we cannot blindly pursue the best value for a given quality metric, but at the same time, we should prevent communities from being over-scaled and over-featured. We now describe the proposed two-level constraints from the perspective of tightness and homogeneity, which can alleviate these restrictions mentioned above when WCC^* is used as an optimization objective.

The tightness constraint emphasizes the topology of each community by focusing on the connections between nodes. In general, nodes with higher degrees can be regarded as leaders with greater influence, thus identifying their communities, and the belongings of the remaining nodes will be continuously determined [45]. However, once the leaders' communities are biased, the impact on the final outcome is severe. We intuitively consider the number of edges between nodes. It is worth noting that the more neighbor nodes of a node belong to the same community, the more likely the node itself belongs to that community. Therefore, we formulate the first-level constraint item as follows:

Definition 2. *The tightness constraint of a node v_i and a community C_k is defined as follows:*

$$T(v_i, C_k) = \frac{\sum_{v_j \in C_k} E^{in}_{v_i, v_j}}{d_i \cdot |C_k|}. \tag{5}$$

When there is an edge between node v_i and node $v_j \in C_k$, $E^{in}_{v_i, v_j} = 1$, otherwise 0. In other words, the more edges between node v_i and nodes in community C_k, the larger the value of $\sum_{v_j \in C_k} E^{in}_{v_i, v_j}$. Thus, by maximizing $T(v_i, C_k)$, the joined node will improve the topological properties of the community. Meanwhile, the degree d_i of node v_i is introduced for normalization, and the scale $|C_k|$ of community C_k is considered to prevent oversize, so that the detected communities are more compact.

The homogeneity constraint grasps the feature distribution of a community by highlighting the similarity between node features. Element-by-element matching [46] is the simplest method, such as jaccard and cosine similarity, the higher the matching degree, the higher the similarity. Regularization [51] is also a widely adopted method, such as L1 and L2 norm, which can mitigate the negative impacts of outliers and missing data. Since we consider both binary and continuous-valued features, the applicability of a given method is our primary concern. Therefore, we formulate the second-level constraint item as follows:

Definition 3. *The homogeneity constraint of a node v_i and a community C_k is defined as follows:*

$$H(v_i, C_k) = \frac{\sum_{v_j \in C_k} |\mathbf{f}_i - \mathbf{f}_j|}{p \cdot |C_k|}. \tag{6}$$

When the features of two nodes are highly consistent, the two nodes are strongly similar, and the corresponding value of $|\mathbf{f}_i - \mathbf{f}_j|$ will be small regardless of the feature type. Thus, by minimizing $H(v_i, C_k)$, finding nodes that agree on more features, the feature distribution of each community is more explicit. As for feature dimension p and community scale $|C_k|$, their effects are equivalent to d_i and $|C_k|$ in the tightness constraint, so the detected communities are more appropriate.

Discussion: In general, a well-separated community contains about 100 nodes, and meaningful larger communities can be generated by merging these relatively small communities [6, 45]. However, unrestricted merging will generate oversized communities, which will further lead to unbalanced communities and free-rider effect. Our quality metric considers higher-order details from the perspective of closed topological and feature triangles, which helps to extract tight core communities, but nodes and edges that are difficult to form closed triangles are excluded from the communities. Our two-level constraints consider lower-order details from the perspective of edge tightness and node feature homogeneity, which guide the growth of core communities as nodes and edges are selectively added. As a consequence, communities remain self-contained unless highly correlated. Therefore, the merging of communities will be regulated and the scale of communities will be widely distributed.

4.3. Local Search Framework

Combining the proposed quality metric with two-level constraints, denoted as U, the optimization objective function we maximize in this study is as follows:

$$U = \sum_{k=1}^{K} \sum_{\forall v_i \in C_k} [WCC^*(v_i, C_k) + T(v_i, C_k) - H(v_i, C_k)]. \tag{7}$$

When $WCC^*(v_i, C_k) \neq 0$, U is equivalent to the revision of $WCC^*(C)$, which enhances its applicability in different tasks. Otherwise, the optimization objective is reduced to two-level constraints, and the performance of community detection can still be used as a benchmark.

Greedy objective function maximization reduces the computational cost significantly, but the quality of the results highly depends on the order of processing [12]. To prevent this issue, we design a local search framework based on maximizing the utility function of each node extracted from the optimization objective U. Specifically, given a node v_i, its utility in the community C_k is defined as follows:

$$u_{ik} = WCC^*(v_i, C_k) + T(v_i, C_k) - H(v_i, C_k). \tag{8}$$

Based on Equation (8), the utility gain ($\Delta u_{ik \rightarrow ik'}$) of node v_i moving from the current C_k to a new community $C_{k'}$ can be

measured. Thus, our search process for both non-overlapping and overlapping communities is summarized as follows:

- **Step 1: (Cumulative Node Utility Updating).** Let's denote $\overline{u}_{ik}^{t}$ as the cumulative utility of node v_i in round t, which also reflects the degree to which node v_i belongs to community C_k at this time. Given any community $C_{k'}$ that includes at least one neighbor of v_i, then in round $t+1$, the cumulative utility of v_i joining the community $(\overline{u}_{ik'}^{t+1})$ is updated as follows:

$$\overline{u}_{ik'}^{t+1} = \alpha \cdot \Delta u_{ik \to ik'} + (1-\alpha) \cdot \overline{u}_{ik}^{t}. \qquad (9)$$

Note that the value of $\alpha \in [0,1]$ indicates a trade-off between historical utility and current utility gain. We empirically specify it as 0.2.

- **Step 2: (Candidate Community Labels Filtering).** Node v_i filters its candidate community labels set (CL_i^{t+1}) for round $t+1$ based on the cumulative utility collected from Step 1:

$$CL_i^{t+1} = \{k' \mid \overline{u}_{ik'}^{t+1} \geq \overline{u}_{ik}^{t}\}. \qquad (10)$$

Note that the community for v_i in round $t+1$ is either its current community C_k or the community $C_{k'}$ in CL_i^{t+1}.

- **Step 3: (Appropriate Community Labels Assigning).** 1) Sort the community labels in CL_i^{t+1} based on successive decreasing of $\overline{u}_{ik'}^{t+1}$, and remove the last $1/|CL_i^{t+1}|$ [52] community labels; 2) The remaining community labels in CL_i^{t+1} are regarded as overlapping communities to which v_i belongs in round $t+1$; 3) Among them, the community label with the greatest utility is regarded as the non-overlapping community to which v_i belongs in round $t+1$.

Algorithm 1 shows the detailed procedures of the above process step by step. Our framework only takes G as input. The initial configuration (line 1) is to assign each node a unique community label. After that, the program loops in four steps. Specifically, in lines 3–7, each node updates the cumulative utility, filters candidate community labels, and determines new communities based on different types of community detection; in lines 8–10, each node selects the community label with the greatest cumulative utility for the next round of search. Finally, if no nodes have changed community affiliation or satisfied the convergence criterion ($t > 20$), the above procedures will terminate and output non-overlapping and overlapping partitions.

5. Experiments

In this section, we evaluate the overall performance of our proposed framework, LSF, on seven real-world network datasets. All experiments were performed on a PC equipped with an Intel quad-core i7 processor (2.60 GHz) and 16GB memory. We conduct experiments as follows:

(1) By comparing the overall performance of LSF and twelve baseline approaches, we demonstrate the effectiveness of the proposed framework. (Section 5.2)

Algorithm 1 Local Search Framework (LSF)

Input: An attributed network $G = (V, E, F)$;
Output: Non-overlapping (overlapping) partition C;
1: $t = 0$, initialize each node belongs to a community:
$\forall v_i \in V, k \leftarrow i, CL_i^0 \leftarrow \{k\}, \overline{u}_{ik}^0 \leftarrow 0$
$k' \leftarrow$ any community $C_{k'}$ where v_i's neighbors belong
2: **while** C changed in the previous round or $t \leq 20$ **do**
3:    **for** each node $v_i \in V$ **do**
4:       Cumulative Node Utility Updating
5:       Candidate Community Labels Filtering
6:       Appropriate Community Labels Assigning
7:    **end for**
8:    **for** each node $v_i \in V$ **do**
9:       $k \leftarrow$ community with the greatest cumulative utility
10:    **end for**
11:    $t \leftarrow t+1$
12: **end while**
13: return C

---

(2) By investigating the running time of LSF and twelve baseline approaches, we verify the scalability and efficiency of the proposed framework. (Section 5.3)

(3) By revealing the convergence of LSF, the effect of two-level constraints, and the topological density and feature entropy of each found community, we make an in-depth analysis of the proposed framework. (Section 5.4)

5.1. Experimental setup

Experimental data. Seven real-world network datasets are used in our experiments, and their ground-truth communities are all known. *Facebook*, *Twitter*, and *Gplus* are friendship networks derived from online social networking sites, and the profile of each node is described by binary features. *Youtube* and *Livejournal* are friendship networks obtained from a video sharing site and a free blogging site, and nodes have no feature details. These five networks are available from the Stanford Network Analysis Platform (SNAP[1]). *Sinanet* is a microblog user relationship network extracted from the sina-microblog site. The topic distribution of each user in the forums generated by the LDA topic model is treated as a continuous-valued feature for each node. *Diabetes* is a citation network formed by citation relationships for scientific publications on diabetes from the PubMed database. The continuous-valued feature for each node corresponds to the TF/IDF weighted word representation of each publication. These two networks are provided by GitHub[2] and LINQS[3], respectively.

We provide some basic statistics for these datasets in Table 3, where $|V|$, $|E|$, $|F|$, $\langle d \rangle = 2|E|/|V|$, #*Comm.*, and *Overlaps* $= \sum_{k^*=1}^{K^*} |C_{k^*}^*|/|V|$ indicate the number of nodes, edges, and features, the average degree of the network, the number of ground-truth communities, and the overlap rate of the ground-truth par-

---

[1]http://snap.stanford.edu/data/index.html.
[2]https://github.com/smileyan448/Sinanet.
[3]https://linqs.soe.ucsc.edu/data.

Table 3: Network Datasets used in Experiments

Network	$|V|$	$|E|$	$|F|$	$\langle d \rangle$	#Comm.	Overlaps
Sinanet	3, 490	28, 657	10	16.42	10	1.00
Diabetes	19, 717	44, 338	500	4.50	3	1.00
Facebook	4, 039	88, 234	157	43.69	146	1.46
Twitter	81, 306	1, 342, 296	33, 208	33.02	3, 170	2.22
Gplus	102, 100	12, 113, 501	805	237.30	438	2.69
Youtube	1, 134, 890	2, 987, 624	-	5.27	8, 385	2.40
Livejournal	3, 997, 962	34, 681, 189	-	17.35	287, 512	5.88

tition, respectively, where $K^* = $ #Comm. and $C_{k^*}^*$ is the k^*th community in the ground-truth partition.

Baseline approaches. Twelve state-of-the-art community detection approaches are selected as baselines for performance comparison, and we divide them into three groups. The first group of baselines consider topology and binary features: CESNA [14] finds overlapping communities based on a probabilistic generative model combining network topology and node features. EDCAR [25] uses the greedy stochastic adaptive search principle to approximate the optimal clustering solution to detect overlapping clusters with high topology density and high feature similarity. CAMAS [15] is based on the established cluster-aware multi-agent system to achieve overlapping clustering in attributed networks.

The second group of baselines consider topology and continuous-valued features: I-Louvain [26] is implemented following the exploration principle of Louvain and optimizes the modularity and the proposed inertia based modularity. NAGC [53] performs non-linear attributed network clustering via symmetric non-negative matrix factorization with positive unlabeled learning. kNN-enhance [13] adds the k nearest neighbor graph of node features to alleviate the sparsity and noise effects of the original network, thus strengthening the found community structure.

As for the remaining baselines, they only focus on topology: SCD [43] divides the network into non-overlapping groups by maximizing the weighted community clustering metric. InfoMap [54] reveals that non-overlapping communities aim to optimize a quality metric expressing the code length of an infinitely long random walk taking place on the network. SCoDA [55] randomly and uniformly picks an edge in the network that is more likely to connect two nodes in the same community than two nodes in different communities, and exploits this idea to build non-overlapping communities by local changes at each edge arrival. FOCS [56] explores locally well-connected overlapping communities by computing community connectedness and neighborhood connectedness scores for each node. BigClam [57] proposes a conceptual model of network community structure, cluster affiliation model, and then employs non-negative matrix factorization to find overlapping communities based on this. LPANNI [52] detects overlapping communities by adopting fixed label propagation sequence based on the ascending order of node importance and label update strategy based on neighbor node influence and historical label preferred strategy.

Parameter settings. All baseline packages are implemented in C++, Java, or Python and can be found on the websites provided by the papers. Since CESNA, NAGC, kNN-enhance, and BigClam require a user-given parameter as the number of communities they found, we set this parameter to #Comm. for comparison. All other parameters of the selected baselines use their default settings.

Evaluation measures. We do not make any special distinction between non-overlapping and overlapping community detection. Four evaluation measures are chosen to assess the quality of detected communities: Average F1 Score ($AvgF1$) [57], Modularity Q [48], Density [47], and Entropy [13]. Given two partitions $C = \{C_1, ..., C_K\}$ and $C^* = \{C_1^*, ..., C_{K^*}^*\}$, $AvgF1$ is defined as follows:

$$AvgF1 = \frac{1}{2K} \sum_{C_k \in C} \max_{C_{k^*}^* \in C^*} F1(C_k, C_{k^*}^*) + \frac{1}{2K^*} \sum_{C_{k^*}^* \in C^*} \max_{C_k \in C} F1(C_{k^*}^*, C_k),$$

$$F1(C_k, C_{k^*}^*) = 2 \cdot \frac{Precision(C_k, C_{k^*}^*) \cdot Recall(C_k, C_{k^*}^*)}{Precision(C_k, C_{k^*}^*) + Recall(C_k, C_{k^*}^*)},$$

$$Precision(C_k, C_{k^*}^*) = \frac{|C_k \cap C_{k^*}^*|}{|C_k|}, \quad Recall(C_k, C_{k^*}^*) = \frac{|C_k \cap C_{k^*}^*|}{|C_{k^*}^*|}.$$

Given a network with n nodes and m edges, and one partition $C = \{C_1, ..., C_K\}$, modularity and density are defined as follows:

$$Modularity\ Q = \frac{1}{2m} \sum_{i=1}^{n} \sum_{j=1, j \neq i}^{n} (A_{ij} - \frac{d_i d_j}{2m}) \sum_{k=1}^{K} c_{ik} c_{jk},$$

$$Density\ (C_k) = \frac{2E_k^{in}}{|C_k| \cdot (|C_k| - 1)},$$

where c_{ik} (c_{jk}) represents the degree of belonging of node v_i (v_j) to the kth community, and E_k^{in} represents the number of internal edges in the kth community.

Given a network with n nodes and p features per node, and one partition $C = \{C_1, ..., C_K\}$, entropy is defined as follows:

$$Entropy\ (C_k) = -\frac{|C_k|}{n} \sum_{l=1}^{p} P_{lk} \log(P_{lk}),$$

where P_{lk} is the fraction of nodes with lth feature in the kth community.

It is expected that better communities (high tightness and homogeneity) of a given network data will have larger values of $AvgF1$, modularity Q, density, and smaller value of entropy.

7

Table 4: Effectiveness Evaluation of LSF and Baselines ignoring Node Features ($AvgF1$)

Network	SCD	InfoMap	SCoDA	FOCS	BigClam	LPANNI	LSF
Sinanet	0.096	0.167	0.025	0.114	0.286	0.174	**0.308**
Diabetes	0.004	0.325	0.003	0.098	0.002	0.097	**0.397**
Facebook	0.197	0.405	0.223	0.328	0.422	0.368	**0.452**
Twitter	0.167	0.151	0.178	0.078	0.101	0.133	**0.299**
Gplus	0.039	0.106	0.049	0.162	0.093	NaN	**0.261**
Youtube	0.177	0.084	0.161	0.148	0.038	0.141	**0.194**
Livejournal	0.139	0.059	**0.169**	0.145	0.103	NaN	0.165

Table 5: Efficiency Evaluation of LSF and Baselines ignoring Node Features (in seconds)

Network	SCD	InfoMap	SCoDA	FOCS	BigClam	LPANNI	LSF
Sinanet	**0.18**	0.85	0.40	1.02	4.25	79.56	0.65
Diabetes	0.33	1.06	0.78	0.99	**0.26**	14.99	1.09
Facebook	0.53	1.17	0.96	1.21	19.50	510.40	1.16
Twitter	4.69	19.51	**1.44**	4.26	467.36	24, 930.29	50.24
Gplus	67.53	95.86	**12.35**	959.38	1, 804.60	NaN	494.67
Youtube	17.49	146.45	**5.57**	39.70	22, 680.47	80, 655.41	100.73
Livejournal	194.18	1, 553.29	**30.40**	232.65	37, 426.82	NaN	2, 057.95

Figure 1: Effectiveness Evaluation of LSF and Baselines using Node Features
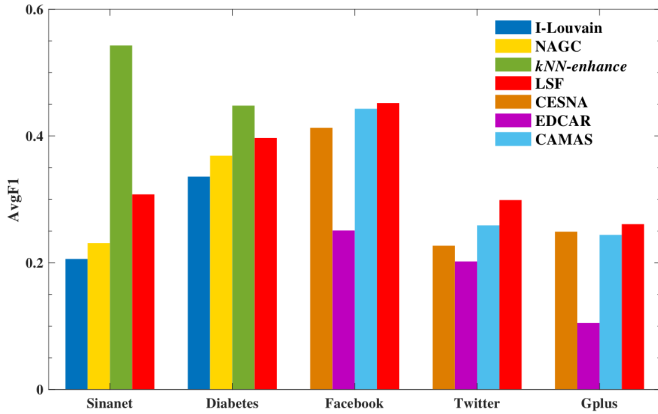

Figure 2: Efficiency Evaluation of LSF and Baselines using Node Features

5.2. Effectiveness Evaluation

To better evaluate the effectiveness of our framework, we comprehensively compare the quality of non-overlapping and overlapping communities found by LSF and twelve baselines on seven experimental datasets. Figure 1 and Table 4 present the $AvgF1$ scores for all tested approaches, with the best scenarios in bold. The NaNs in Table 4 are due to the fact that the node's feature type is not suitable for the baseline tool, or some baseline tools run out of memory or too expensive to run as the scale of the network increases.

We have the following observations and conclusions from these results. In terms of approaches that consider continuous-valued features, *k*NN-enhance performs the best, which indicates that our processing of continuous-valued features still needs to be further improved. However, the performance of our framework is also acceptable as it outperforms I-Louvain and NAGC. In terms of approaches that consider binary features, although CESNA and CAMAS are competitive, our frame-
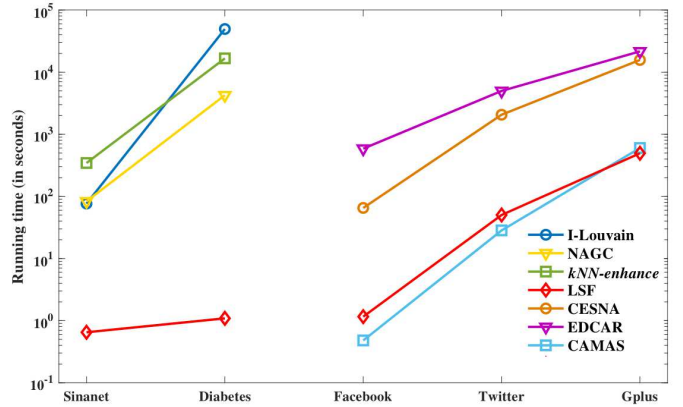
work performs significantly better than all of them, which also demonstrates that our processing of binary features is appropriate. In terms of approaches that only focus on topology, our framework maintains the best performance, and only SCoDA is slightly better than ours on *Livejournal*. This declares that the integration of network topology and node features is indeed helpful to improve the quality of the found communities.

It is not difficult to find that the superiority of our framework is obvious. Specifically, it can handle different types of node features and exhibits satisfactory overall performance in different community detection tasks.

5.3. Efficiency Evaluation

To better evaluate the efficiency and scalability of our framework, we present the running time of LSF and twelve baselines on seven experimental datasets in Figure 2 and Table 5, with the best scenarios also in bold. Like Table 4, the NaNs in Table 5 are also due to the fact that the node's feature type is not suitable
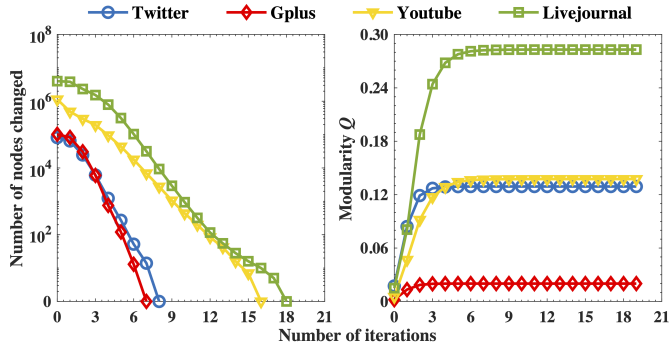
Figure 3: Convergence Verification of LSF based on the Number of Nodes Changed and Modularity Q

Figure 4: Evaluation of the Effects of Two-level Constraints

for the baseline tool, or some baseline tools run out of memory or too expensive to run as the scale of the network increases.

We have the following findings and determinations from these results. In terms of approaches that consider continuous-valued features, the superiority of our framework is obvious. Although the performance of kNN-enhance is better than ours, its time cost is too high to be applied to larger-scale networks. In terms of approaches that consider binary features, CAMAS is the most efficient, followed by our framework, and both consistently outperform CESNA and EDCAR. The efficiency difference between ours and CAMAS tends to weaken as the scale of the network increases. In terms of approaches that only focus on topology, SCoDA is the best, followed by SCD and FOCS, then InfoMap and ours, and finally BigClam and LPANNI. Since our framework considers both topology and node features, our efficiency is inferior to some baselines, but the overall performance is still acceptable.

It is not difficult to find that approaches that combine topology and node features are generally less efficient than approaches that only focus on topology. However, our framework maintains satisfactory efficiency and scalability, on the same order of magnitude as CAMAS and InfoMap. While still not as good as some baselines, we can identify higher quality communities, which is enough to make up the slight inferiority on efficiency.

5.4. Inside Analysis

To better interpret the good performance of our framework, we here take a further look inside the framework. Specifically, we analyze the convergence of LSF, the effect of the proposed constraints, and the quality of each community found.

Verification of convergence. We select four datasets *Twitter*, *Gplus*, *Youtube*, and *Livejournal* as representatives, and examine the number of nodes whose community labels change and the value of modularity Q in each round of our framework runs. The results are shown in Figure 3.

We can find: In terms of the number of nodes whose community labels change, *Twitter* and *Gplus* have similar trends, with the number of nodes decreasing by an order of magnitude in almost every round, while *Youtube* and *Livejournal* have similar trends, generally two or three rounds will also decrease by an
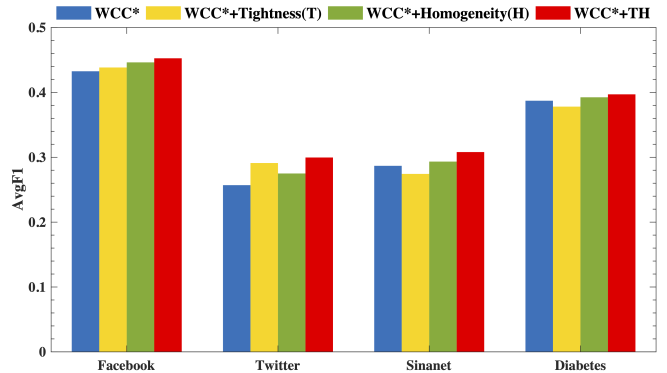
order of magnitude. In terms of the value of modularity Q, the trends are very similar regardless of the datasets. The Q value increases continuously over several rounds and then tends to be smooth. In summary, our framework is robust and converges fast, and exhibits great potential on large-scale networks. Considering the fact that *Twitter* and *Gplus* are dense, and *Youtube* and *Livejournal* are sparse, the results of the convergence verification reveal that our framework seems to be more suitable for dense networks.

Effects of two-level constraints. We select four datasets *Facebook*, *Twitter*, *Sinanet*, and *Diabetes* as representatives, and take an ablation study to investigate the effects of the proposed constraints. Specifically, we consider four cases: without the tightness and homogeneity constraints, with the tightness or homogeneity constraint, and with the tightness and homogeneity constraints. The results are shown in Figure 4.

We can find: When the network is dense (*Facebook* and *Twitter*), the effect of the tightness constraint is positive and the performance of our framework improves. The effect of the homogeneity constraint is unstable, and a few features further enhance the degree of improvement, but as the number of features increases, the effect of the homogeneity constraint decreases or even becomes negative. When the network is sparse (*Sinanet* and *Diabetes*), the effect of the tightness constraint is negative and the performance of our framework generally declines. The effect of the homogeneity constraint is positive, but also seems to be controlled as the number of features increases. In summary, it is indeed useful with two-level constraints than without. However, if only one constraint is considered, due to the heterogeneity of network topology and node features, the overall performance improvement depends on the trade-off between the density of the topology and the number of node features.

Analysis of found communities. We select two datasets *Twitter* and *Diabetes* as representatives, and further explore the topological density and feature entropy scores for each community found with and without two-level constraints. The results are shown in Figure 5.

We can find: In terms of tightness, larger communities are more likely to contain more edges and have a larger density score. *Twitter* is dense, and the tightness constraint makes it easier for closely connected nodes to be in the same commu-
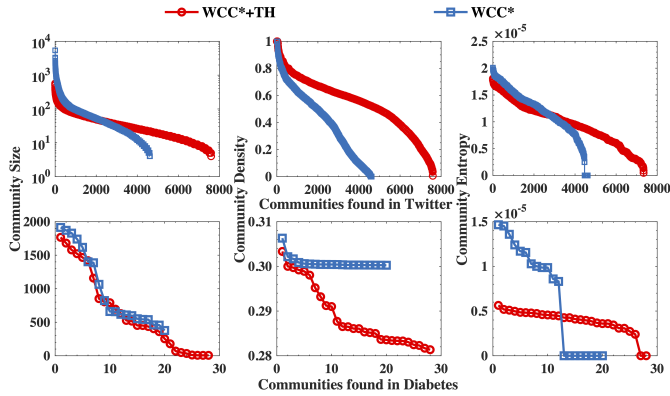
Figure 5: Evaluation of Topological Density and Feature Entropy for Each Community Found

nity, thus improving the quality of each community. In contrast, *Diabetes* is sparse, and the tightness constraint forces weakly connected nodes to the same community, and the overall performance is compromised. In terms of homogeneity, smaller communities tend to exhibit more uniform node features and have a smaller entropy score. *Twitter* has too many features, and the homogeneity constraint considers the differences in each dimension of node features, thus reducing the quality as the community scale becomes smaller. On the contrary, *Diabetes* has a small number of features and is the relatively smooth continuous-valued type. The homogeneity constraint guarantees the feature distribution of each community, and the overall performance is improved. In summary, by balancing the proportions of topology tightness and feature homogeneity, our searched communities are more compact and reasonable, resulting in better performance.

6. Conclusions and Future Work

In this study, we first introduce a novel quality metric based on the concepts of closed topology and feature triangles, which not only evaluates the quality of the found communities, but can also be treated as an optimization objective function. We then formulate two constraint items from the perspective of node features and network topology, which alleviate unbalanced communities and free-rider effect by regulating the topological tightness and feature homogeneity of each community found. Finally, combining the proposed metric and constraint items as the objective function, we develop a local search framework by optimizing it to achieve community detection in attributed networks. Experimental results on seven real-world networks show that the overall performance of our framework consistently outperforms all selected state-of-the-art approaches. In the future, we will focus on the dynamic evolution of the community structure and investigate how to design parallel schemes to make our framework more efficient.

References

[1] B. K. AlShebli, T. Rahwan, W. L. Woon, The preeminence of ethnic diversity in scientific collaboration, Nature Communications 9 (1) (2018) 1–10.

[2] Y. Liu, C. Liang, X. He, J. Peng, Z. Zheng, J. Tang, Modelling high-order social relations for item recommendation, IEEE Transactions on Knowledge and Data Engineering (2020) 1–12. doi:10.1109/TKDE.2020.3039463.

[3] F. Liu, Z. Li, B. Wang, J. Wu, J. Yang, J. Huang, Y. Zhang, W. Wang, S. Xue, S. Nepal, et al., eriskcom: An e-commerce risky community detection platform, Proceedings of the VLDB Endowment (2022) 1–17.

[4] X. Meng, W. Li, X. Peng, Y. Li, M. Li, Protein interaction networks: Centrality, modularity, dynamics, and applications, Frontiers of Computer Science 15 (6) (2021) 1–17.

[5] I. Falih, N. Grozavu, R. Kanawati, Y. Bennani, Community detection in attributed network, in: Companion Proceedings of the Web Conference, 2018, pp. 1299–1306.

[6] P. Chunaev, Community detection in node-attributed social networks: A survey, Computer Science Review 37 (2020) 100286–100310. doi:10.1016/j.cosrev.2020.100286.

[7] Y. Zhang, J. Wu, C. Zhou, Z. Cai, Instance cloned extreme learning machine, Pattern Recognition 68 (2017) 52–65. doi:10.1016/j.patcog.2017.02.036.

[8] K. Ding, J. Li, H. Liu, Interactive anomaly detection on attributed networks, in: Proceedings of the 12th International Conference on Web Search and Data Mining, ACM, 2019, pp. 357–365. doi:10.1145/3289600.3290964.

[9] F. Liu, S. Xue, J. Wu, C. Zhou, W. Hu, C. Paris, S. Nepal, J. Yang, P. S. Yu, Deep learning for community detection: Progress, challenges and opportunities, in: Proceedings of the 29th International Joint Conference on Artificial Intelligence, 2020, pp. 4981–4987. doi:10.24963/ijcai.2020/693.

[10] X. Su, S. Xue, F. Liu, J. Wu, J. Yang, C. Zhou, W. Hu, C. Paris, S. Nepal, D. Jin, et al., A comprehensive survey on community detection with deep learning, IEEE Transactions on Neural Networks and Learning Systems (2022) 1–21. doi:10.1109/TNNLS.2021.3137396.

[11] Y. Zhou, H. Cheng, J. X. Yu, Graph clustering based on structural/attribute similarities, Proceedings of the VLDB Endowment 2 (1) (2009) 718–729. doi:10.14778/1687627.1687709.

[12] C. Zhe, A. Sun, X. Xiao, Community detection on large complex attribute network, in: Proceedings of the 25th International Conference on Knowledge Discovery and Data Mining, ACM, 2019, pp. 2041–2049. doi:10.1145/3292500.3330721.

[13] C. Jia, Y. Li, M. B. Carson, X. Wang, J. Yu, Node attribute-enhanced community detection in complex networks, Scientific Reports 7 (1) (2017) 1–15.

[14] J. Yang, J. J. McAuley, J. Leskovec, Community detection in networks with node attributes, in: Proceedings of the 13th International Conference on Data Mining, IEEE Computer Society, 2013, pp. 1151–1156. doi:10.1109/ICDM.2013.167.

[15] Z. Bu, G. Gao, H. Li, J. Cao, CAMAS: A cluster-aware multiagent system for attributed graph clustering, Information Fusion 37 (2017) 10–21. doi:10.1016/j.inffus.2017.01.002.

[16] Z. Li, J. Liu, K. Wu, A multiobjective evolutionary algorithm based on structural and attribute similarities for community detection in attributed networks, IEEE Transactions on Cybernetics 48 (7) (2018) 1963–1976. doi:10.1109/TCYB.2017.2720180.

[17] A. R. Benson, D. F. Gleich, J. Leskovec, Higher-order organization of complex networks, Science 353 (6295) (2016) 163–166.

[18] A. Paranjape, A. R. Benson, J. Leskovec, Motifs in temporal networks, in: Proceedings of the 10th International Conference on Web Search and Data Mining, ACM, 2017, pp. 601–610. `doi:10.1145/3018661.3018731`.

[19] R. A. Rossi, N. K. Ahmed, E. Koh, Higher-order network representation learning, in: Companion of the Web Conference, ACM, 2018, pp. 3–4. `doi:10.1145/3184558.3186900`.

[20] H. Yin, A. R. Benson, J. Leskovec, D. F. Gleich, Local higher-order graph clustering, in: Proceedings of the 23rd International Conference on Knowledge Discovery and Data Mining, ACM, 2017, pp. 555–564. `doi:10.1145/3097983.3098069`.

[21] S. Huang, Y. Li, Z. Bao, Z. Li, Towards efficient motif-based graph partitioning: An adaptive sampling approach, in: Proceedings of the 37th International Conference on Data Engineering, IEEE, 2021, pp. 528–539. `doi:10.1109/ICDE51399.2021.00052`.

[22] F. Xia, S. Yu, C. Liu, J. Li, I. Lee, Chief: Clustering with higher-order motifs in big networks, IEEE Transactions on Network Science and Engineering (2021).

[23] G. Gao, Z. Wu, L. Zhang, J. Cao, X. Qi, Community detection via local learning based on generalized metric with neighboring regularization, IEEE Transactions on Systems Man Cybernetics-Systems 52 (1) (2022) 498–510. `doi:10.1109/TSMC.2020.3003019`.

[24] Y. Wu, R. Jin, J. Li, X. Zhang, Robust local community detection: On free rider effect and its elimination, Proceedings of the VLDB Endowment 8 (7) (2015) 798–809. `doi:10.14778/2752939.2752948`.

[25] S. Günnemann, B. Boden, I. Färber, T. Seidl, Efficient mining of combined subspace and subgraph clusters in graphs with feature vectors, in: Proceedings of the 17th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining, Vol. 7818 of Lecture Notes in Computer Science, Springer, 2013, pp. 261–275. `doi:10.1007/978-3-642-37453-1\_22`.

[26] D. Combe, C. Largeron, M. Géry, E. Egyed-Zsigmond, I-louvain: An attributed graph clustering method, in: Proceedings of the 14th International Symposium on Advances in Intelligent Data Analysis, Vol. 9385 of Lecture Notes in Computer Science, Springer, 2015, pp. 181–192. `doi:10.1007/978-3-319-24465-5\_16`.

[27] Z. Xu, Y. Ke, Y. Wang, H. Cheng, J. Cheng, A model-based approach to attributed graph clustering, in: Proceedings of the 31st International Conference on Management of Data, ACM, 2012, pp. 505–516. `doi:10.1145/2213836.2213894`.

[28] Z. Bu, H. Li, J. Cao, Z. Wang, G. Gao, Dynamic cluster formation game for attributed graph clustering, IEEE Transactions on Cybernetics 49 (1) (2019) 328–341. `doi:10.1109/TCYB.2017.2772880`.

[29] X.-L. Xu, Y.-Y. Xiao, X.-H. Yang, L. Wang, Y.-B. Zhou, Attributed network community detection based on network embedding and parameter-free clustering, Applied Intelligence 52 (7) (2022) 8073–8086.

[30] J. Cao, H. Wang, D. Jin, J. Dang, Combination of links and node contents for community discovery using a graph regularization approach, Future Generation Computer Systems 91 (2019) 361–370. `doi:10.1016/j.future.2018.08.009`.

[31] J. Wu, Z. Cai, S. Ao, Hybrid dynamic K-nearest-neighbour and distance and attribute weighted method for classification, International Journal of Computer Applications in Technology 43 (4) (2012) 378–384. `doi:10.1504/IJCAT.2012.047164`.

[32] J. Wu, S. Pan, X. Zhu, Z. Cai, P. Zhang, C. Zhang, Self-adaptive attribute weighting for naive bayes classification, Expert Systems with Applications 42 (3) (2015) 1487–1502. `doi:10.1016/j.eswa.2014.09.019`.

[33] L. M. Smith, L. Zhu, K. Lerman, A. G. Percus, Partitioning networks with node attributes by compressing information flow, ACM Transactions on Knowledge Discovery from Data 11 (2) (2016) 15:1–15:26. `doi:10.1145/2968451`.

[34] D. Malhotra, A. Chug, A modified label propagation algorithm for community detection in attributed networks, International Journal of Information Management Data Insights 1 (2) (2021) 100030–100041.

[35] K. Berahmand, S. Haghani, M. Rostami, Y. Li, A new attributed graph clustering by using label propagation in complex networks, Journal of King Saud University - Computer and Information Sciences 34 (5) (2022) 1869–1883. `doi:10.1016/j.jksuci.2020.08.013`.

[36] X. Xie, M. Song, C. Liu, J. Zhang, J. Li, Effective influential community search on attributed graph, Neurocomputing 444 (2021) 111–125.

[37] F. Liu, J. Wu, C. Zhou, J. Yang, Evolutionary community detection in dynamic social networks, in: Proceedings of the 29th International Joint Conference on Neural Networks, IEEE, 2019, pp. 1–7. `doi:10.1109/IJCNN.2019.8852006`.

[38] F. Liu, J. Wu, S. Xue, C. Zhou, J. Yang, Q. Sheng, Detecting the evolving community structure in dynamic social networks, World Wide Web 23 (2) (2020) 715–733. `doi:10.1007/s11280-019-00710-z`.

[39] K. Sotiropoulos, C. E. Tsourakakis, Triangle-aware spectral sparsifiers and community detection, in: Proceedings of the 27th International Conference on Knowledge Discovery and Data Mining, ACM, 2021, pp. 1501–1509. `doi:10.1145/3447548.3467260`.

[40] P. Li, L. Huang, C. Wang, J. Lai, Edmot: An edge enhancement approach for motif-aware community detection, in: Proceedings of the 25th International Conference on Knowledge Discovery and Data Mining, ACM, 2019, pp. 479–487. `doi:10.1145/3292500.3330882`.

[41] P. Li, L. Huang, C. Wang, D. Huang, J. Lai, Community detection using attribute homogenous motif, IEEE Access 6 (2018) 47707–47716. `doi:10.1109/ACCESS.2018.2867549`.

[42] L. Hu, X. Pan, H. Yan, P. Hu, T. He, Exploiting higher-order patterns for community detection in attributed graphs, Integrated Computer-Aided Engineering 28 (2) (2021) 207–218. `doi:10.3233/ICA-200645`.

[43] A. Prat-Perez, D. Dominguez-Sal, J.-L. Larriba-Pey, High quality, scalable and parallel community detection for large real graphs, in: Proceedings of the 23rd International Conference on World Wide Web, ACM, 2014, pp. 225–236. `doi:10.1145/2566486.2568010`.

[44] T. Lyu, L. Bing, Z. Zhang, Y. Zhang, Efficient and scalable detection of overlapping communities in big networks, in: Proceedings of the 16th International Conference on Data Mining, IEEE Computer Society, 2016, pp. 1071–1076. `doi:10.1109/ICDM.2016.0138`.

[45] S. Fortunato, D. Hric, Community detection in networks: A user guide, Physics Reports 659 (2016) 1–44.

[46] M. Rezaei, P. Fränti, Set matching measures for external cluster validity, IEEE Transactions on Knowledge and Data Engineering 28 (8) (2016) 2173–2186. `doi:10.1109/TKDE.2016.2551240`.

[47] T. Chakraborty, A. Dalmia, A. Mukherjee, N. Ganguly, Metrics for community analysis: A survey, ACM Computing Surveys 50 (4) (2017) 54:1–54:37. `doi:10.1145/3091106`.

[48] M. E. Newman, Modularity and community structure in networks, Proceedings of the National Academy of Sciences 103 (23) (2006) 8577–8582.

[49] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (8) (2000) 888–905. `doi:10.1109/34.868688`.

[50] S. Fortunato, M. Barthelemy, Resolution limit in community detection, Proceedings of the National Academy of Sciences 104 (1) (2007) 36–41.

[51] M. A. Javed, M. S. Younis, S. Latif, J. Qadir, A. Baig, Community detection in networks: A multidisciplinary review, Journal of Network and Computer Applications 108 (2018) 87–111. `doi:10.1016/j.jnca.2018.02.011`.

[52] M. Lu, Z. Zhang, Z. Qu, Y. Kang, LPANNI: overlapping community detection using label propagation in large-scale complex networks, IEEE Transactions on Knowledge and Data Engineering 31 (9) (2019) 1736–1749. `doi:10.1109/TKDE.2018.2866424`.

[53] S. Maekawa, K. Takeuchi, M. Onizuka, Non-linear attributed graph clustering by symmetric NMF with PU learning, CoRR abs/1810.00946 (2018).

[54] M. Rosvall, C. T. Bergstrom, Maps of information flow reveal community structure in complex networks, CoRR physics.soc-ph/0707.0609 (2007).

[55] A. Hollocou, J. Maudet, T. Bonald, M. Lelarge, A linear streaming algorithm for community detection in very large networks, CoRR abs/1703.02955 (2017).

[56] S. Bandyopadhyay, G. Chowdhary, D. Sengupta, FOCS: fast overlapped community search, IEEE Transactions on Knowledge and Data Engineering 27 (11) (2015) 2974–2985. `doi:10.1109/TKDE.2015.2445775`.

[57] J. Yang, J. Leskovec, Overlapping community detection at scale: A non-negative matrix factorization approach, in: Proceedings of the 6th International Conference on Web Search and Data Mining, ACM, 2013, pp. 587–596. `doi:10.1145/2433396.2433471`.