Triangle-oriented Community Detection Considering Node Features and Network Topology

GUANGLIANG GAO, Jiangsu Police Institute, China
WEICHAO LIANG, Southwest Jiaotong University, China
MING YUAN, HANWEI QIAN, and QUN WANG, Jiangsu Police Institute, China
JIE CAO, Hefei University of Technology, China

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.

CCS Concepts: • Information systems → Data mining; • Human-centered computing → Social network analysis; • Mathematics of computing → Non-parametric optimization;

Additional Key Words and Phrases: Community detection, graph clustering, community structure, attributed network, quality metric, optimization

ACM Reference format:
Guangliang Gao, Weichao Liang, Ming Yuan, Hanwei Qian, Qun Wang, and Jie Cao. 2023. Triangle-oriented Community Detection Considering Node Features and Network Topology. ACM Trans. Web 18, 1, Article 11 (November 2023), 21 pages. https://doi.org/10.1145/3626190

This work was supported in part by the Key Program of National Natural Science Foundation of China under Grant No. 92046026, and in part by the Natural Science Foundation of the Jiangsu Higher Education Institutions of China under Grant No. 23KJB520009, and in part by the Philosophy and Social Foundation of the Jiangsu Higher Education Institutions of China under Grant No. 2022SJB0466, and in part by the Key Discipline Construction Project of Cyberspace Security of the 14th Five-Year Plan of Jiangsu Province.

Authors’ addresses: G. Gao, M. Yuan, H. Qian, and Q. Wang, Department of Computer Information and Cyber Security, Jiangsu Police Institute, China; e-mails: guangliang.gao@njust.edu.cn, [ming.yuan, hanwei.qian, qun.wang]@jspi.edu.cn; W. Liang (Corresponding author), School of Computing and Artificial Intelligence, 999 Xi’an Road, Southwest Jiaotong University, Chengdu, 611756, China; e-mail: weicliang@foxmail.com; J. Cao, Research Institute of Big Knowledge, Hefei University of Technology, China; e-mail: cao_jie@hhu.edu.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, or 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.
1559-1131/2023/11-ART11 $15.00
https://doi.org/10.1145/3626190

ACM Transactions on the Web, Vol. 18, No. 1, Article 11. Publication date: November 2023.
1 INTRODUCTION

Networks provide a natural way to express the complex relationships in our daily life, such as friend interactions [33], information dissemination [8, 32], and biological modules collaborations [23, 38]. 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 [7, 12]. 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 [10, 15, 50].

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 [4, 26, 31, 57, 59, 60]. 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, kNN-enhance [26], AGGMMR [59], and SA-Cluster [60] 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 instance, CAMAS [4], MOEA-SA [31], and CESNA [57] 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 [18].

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. First, 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 [2, 41, 45] have demonstrated that higher-order details, i.e., motifs, are conducive to uncovering the underlying mechanisms of networks. Though there are some attempts [24, 52, 58] that detect communities based on motifs, they are often not applicable to attributed networks. Second, it is a consensus [4, 7, 12, 21, 26, 31, 57, 59, 60] 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 [16] or free-rider effect [51]. Third, scalability, ignored by many researchers, is an important factor to consider. Optimization based on global models [9, 17, 39, 57] 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 so-called two-level constraints from the perspective of nodes and edges 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 [5, 20, 54, 55, 60], which focus on formulating various distance or similarity functions to find communities through attributed network clustering. For instance, Zhou et al. [60] developed a clustering framework SA-Cluster, which uses a unified distance metric to measure both topological and feature similarity, and follows the clustering process of K-medoids. Xu et al. [55] transformed attributed network clustering into a standard probabilistic inference problem based on a defined Bayesian probabilistic model and proposed a variational algorithm to solve it, which avoids the artificial design of distance functions. Bu et al. [5] 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. [54] 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. Hu et al. [20] introduced a three-layer node-feature-value hierarchical structure and simulated the generative process of complex networks through a formulated Bayesian model, and then proposed a variational Bayesian algorithm called TARA to identify meaningful communities.

The second group is edge-oriented approaches [3, 37, 48, 53, 59]. Since community detection considering network topology has been widely studied, 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. [48] 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 [37] 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. [3] 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. [59] 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. [53] 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 [22, 24, 29, 30, 49, 52, 58]. Motifs lie between the microscopic proximity structure and mesoscopic community structure and help find communities that maintain building blocks in the network. Recent studies [24, 30, 49, 52, 58] usually only consider network topology and rarely involve node features. For instance, Huang et al. [24] 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 [49] 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. [30] proposed an edge enhancement method for motif-aware community detection in purely topological networks. For attributed networks, they [29] 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. [22] 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 \( V = \{v_1, v_2, \ldots, v_n\} \) is a set of \( n \) nodes, \( E \subseteq V \times V \) is a set of \( m \) edges that connect two nodes of \( V \), and \( F = \{f_1, f_2, \ldots, f_n\} \) 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 \( f_i \in \mathbb{R}^{1 \times p} \) denotes the feature vector of \( v_i \), where \( p \) is the dimension of the feature vector. We also use the adjacency matrix \( A \in \{0, 1\}^{n \times n} \) and the node feature matrix \( B = \{B_{ij}\} \in \mathbb{R}^{n \times p} \) 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 \( j \)th feature is presented in \( f_i \) 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 $C = \{C_1, C_2, \ldots, C_K\}$ 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. $WCC$ \[35, 43\] 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:

$$
WCC(v_i, C_k) = \begin{cases}
\frac{t(v_i, C_k)}{t(v_i, V)} - \frac{vt(v_i, V)}{|C_k - \{v_i\}| + vt(v_i, V - C_k)}, & t(v_i, V) \neq 0, \\
0, & t(v_i, V) = 0,
\end{cases}
$$

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:

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

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

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

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 metric $WCC$ 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.

#### 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. Modularity is the most typical instance. Although it has been criticized for the “resolution limit” problem \[13\], its capabilities are still recognized due to the triangle counting embedded in the definition \[11\]. $WCC$ also focuses on the properties of communities from the perspective of closed topological triangles. In fact, there is a consensus from extensive research \[7, 12\] that node feature information can be treated as
Table 1. Statistics of Edges in Closed Feature Triangles in Facebook and Sinanet

| Network | Total number of triangles | 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 |

1Those 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

| Network | Type of triangle | Number of triangles |
|---------|-----------------|--------------------|
|         | ground-truths¹  | all communities²   |
| 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 |

¹Those nodes that form the triangle are in the ground-truths.
²Those nodes that form the triangle belong to the same community in the ground-truths.

A supplement to explain the formation mechanism of communities as it becomes available, but the contribution of node features and network topology to communities still needs to be further interpreted. How to integrate becomes the most critical issue to achieve community detection by considering both node features and network topology. Blind integration may provide biased results based on irrelevant ground-truths [42]. We try to balance the effects of these two profiles adaptively through the triangular structure. When the topological triangles in the community are more (or less) than the feature triangles, the tightness and homogeneity of the community mainly depend on the nodes that form the topological triangles (or feature triangles), and the nodes that form the feature triangles (or topological triangles) help to improve the properties of the community. Similarly, the number of topological triangles and feature triangles tends to be the same, and the community depends on both of them. Hence, we define the closed feature triangle based on the intuition of WCC as follows:

Definition 4.1. 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 terms of binary features, we directly compare the feature values, i.e., \(\exists l \in \{1, 2, \ldots, p\}, B_{xl} = B_{yl} = B_{zl} = 1\).

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

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 terms 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 far greater 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 structures [14, 27].

In terms of Sinanet, its feature type is continuous, and we count the number of closed feature triangles based on Definition 4.1. 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:

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

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.

**Discussion:** WCC*($v_i, C_k$) focuses on both the number of topological and feature triangles and the number of nodes forming those triangles to measure the quality of communities. In Equation (4), the first part $\frac{tf(v_i, C_k)}{tf(v_i, NC)}$ improves the quality of the community by triangle counting, the more triangles contained in the community, the higher the quality. However, only maximizing the first part will inhibit the growth of the community, because $tf(v_i, C_k)$ only considers the triangle counting among $v_i$ and any two nodes in $C_k$, while $tf(v_i, NC)$ further considers any two nodes in $N(v_i)$ and, respectively, in $N(v_i)$ and $C_k$. A smaller community is more likely to close the gap between $tf(v_i, C_k)$ and $tf(v_i, NC)$. To ensure that the communities found are meaningful, the second part $\frac{vtf(v_i, NC)}{|C_k - \{v_i\}| + vtf(v_i, N(v_i))}$ focuses on the number of nodes forming triangles. Once a network is given, $vtf(v_i, N(v_i))$ is constant. Obviously, maximizing the second part will promote the scale of the community to be larger, because the larger community will make the increase of $vtf(v_i, NC)$ significantly exceeds that of $|C_k - \{v_i\}|$. As a conclusion, simultaneously maximizing the first and second parts can guarantee the quality and scale of the communities found considering both node features and network topology. In addition, it is obvious that the minimum value of WCC*($v_i, C_k$) is 0. When community $C_k$ contains and only contains all neighbors of $v_i$ ($NC = N(v_i) = C_k$), WCC*($v_i, C_k$) = $1 \cdot \frac{vtf(v_i, N(v_i))}{d_i + vtf(v_i, N(v_i))}$, if all neighbors can form a triangle with $v_i$, then WCC*($v_i, C_k$) gets the maximum value 0.5 ($vtf(v_i, N(v_i)) = d_i$). Thus, the value range of WCC*($v_i, C_k$) is $[0, 0.5]$.

**4.2 Two-level Constraints**

Intuitively, extensive research [6, 44] achieves community detection by maximizing or minimizing the value of a given quality metric such as modularity, WCC, and normalized cut [47]. However, it is
not true that optimization of quality metrics always yields satisfactory performance for community detection, e.g., the resolution limit of modularity, the unbalanced scale of communities \[16\], the free-rider effect \[51\]. 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 \[7, 12\] 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 \[7\]. 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 \[7, 14\]. 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 4.2.** The tightness constraint of a node \(v_i\) and a community \(C_k\) is defined as follows:

\[
T(v_i, C_k) = \frac{\sum_{v_j \in C_k} E_{v_i,v_j}^{in}}{d_i \cdot |C_k|}. \tag{5}
\]

When there is an edge between node \(v_i\) and node \(v_j \in C_k\), \(E_{v_i,v_j}^{in} = 1\), otherwise 0. In other words, the minimum value of \(T(v_i, C_k)\) is 0, the more edges between node \(v_i\) and nodes in community \(C_k\), the larger the value of \(\sum_{v_j \in C_k} E_{v_i,v_j}^{in}\). If all neighbors of \(v_i\) are in community \(C_k\), then \(T(v_i, C_k) = \frac{1}{|C_k|}\). Thus, the value range of \(T(v_i, C_k)\) is \([0, 1]\). 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 \[44\] is the simplest method, such as jaccard and cosine similarity, the higher the matching degree, the higher the similarity. Regularization \[25\] 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 4.3.** The homogeneity constraint of a node \(v_i\) and a community \(C_k\) is defined as follows:

\[
H(v_i, C_k) = \frac{\sum_{v_j \in C_k} |f_i - f_j|}{p \cdot |C_k|}. \tag{6}
\]
When the features of two nodes are highly consistent, the two nodes are strongly similar, and the corresponding value of $|f_i - f_j|$ will be small regardless of the feature type. Thus, the minimum value of $H(v_i, C_k)$ is 0, if $v_i$ has binary features and is different from all nodes in community $C_k$, $H(v_i, C_k)$ gets the maximum value 1, i.e., the value range of $H(v_i, C_k)$ is $[0, 1]$. 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 [7, 14]. 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 objective function we maximize in this study is as follows:

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

(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 [59]. 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:

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

(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 us denote $u_{ik}^t$ 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 ($u_{ik'}^{t+1}$) is updated as follows:

$$\bar{u}_{ik'}^{t+1} = \alpha \cdot \Delta u_{ik \rightarrow ik'} + (1 - \alpha) \cdot u_{ik}^t. \tag{9}$$
ALGORITHM 1: Local Search Framework (LSF)

Input: An attributed network $G = (V, E, F)$;

Output: Non-overlapping (and Overlapping) partition $C$;

1: $t = 0$, initialize each node belongs to a community:
   $\forall v_i \in V$, $k \leftarrow i$, $CL_i^0 \leftarrow \{k\}$, $u_{ik}^0 \leftarrow 0$

2: $k' \leftarrow$ any community $C_{k'}$ where $v_i$’s neighbors belong

3: while $C$ changed in the previous round or $t \leq 20$ do

4:     for each node $v_i \in V$ do

5:         Cumulative Node Utility Updating

6:         Candidate Community Labels Filtering

7:         Appropriate Community Labels Assigning

8:     end for

9:     for each node $v_i \in V$ do

10:        $k \leftarrow$ community with the greatest cumulative utility

11:     end for

12:     $t \leftarrow t + 1$

13: end while

14: return $C$

• 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:

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

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 $\bar{u}_{ik'}^{t+1}$, and remove the last $1/|CL_i^{t+1}|$ [34] 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. 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 3â€“7, each node updates the cumulative utility, filters candidate community labels, and determines new communities based on different types of community detection; in lines 8â€“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 ($t > 20$), then the above procedures will terminate and output non-overlapping and overlapping partitions. Our framework only takes an attributed network $G = (V, E, F)$ as input. Let $\langle d \rangle = 2|E|/|V|$ denotes the average degree of the network. At initialization, we assign all nodes to different communities, and the cost is $O(|V|)$. Afterwards, the four steps are iterated continuously. First, we count the number of topological and feature triangles including each node by intersecting the neighbor node set and the neighbor node set of neighbor nodes, and then update the cumulative utility of each node. When the neighbor node set is sorted, the cost of topology- and feature-based intersection operations are $O(|d|)$ and $O(|d| \cdot |F|)$, respectively. Therefore, the cost of all nodes is $O(|V| \cdot \langle d \rangle + |V| \cdot \langle d \rangle \cdot |F|)$. Then, all nodes filter candidate community labels by comparing the cumulative utility of staying in the current community and joining the community containing neighbor nodes, and the cost is $O(|V| \cdot \langle d \rangle)$. Finally,
Table 3. Real-world Network Datasets Used in Experiments

| Network  | $|V|$  | $|E|$  | $|F|$  | $\langle d \rangle$ | #Comms | 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     |

we determine the community affiliation of all nodes, the cost is $O(|V|)$. In summary, the cost of each iteration is the sum of the above steps, which is $O(|V| \cdot \langle d \rangle + |V| \cdot |F| + |V| \cdot \langle d \rangle + |V|)$. Since $|E|$ is comparable to or even greater than $|V|$ in most real-world networks, the total cost of our framework is $O(t \cdot |E| \cdot |F|)$ in the worst case, where $t$ is the number of iterations.

5 EXPERIMENTS

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

1. By comparing the overall performance of LSF and twelve baseline approaches, we demonstrate the superiority and effectiveness of the proposed framework (Section 5.2).
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 effects of $WCC^*$ and 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. A series of synthetic network datasets and seven real-world network datasets are used in our experiments, and their ground-truth communities are all known. In terms of synthetic networks, we choose the ANC_Generator\(^1\) [28], which restricts each node to be associated with two numeric features and allows users to specify the number of nodes, edges, and communities via a set of parameters. Without loss of generality, we consider the number of nodes $|V| \in \{500, 1,000, 5,000, 10,000\}$ and generate 100 networks for each setting. In each network, the minimum number of edges $MTE = 10 \times |V|$, the number of ground-truth communities $K^* = 10$, and the threshold of community homogeneity $\theta$ is random in $[0, 1]$, the smaller the value, the more homogeneous.

In terms of real-world networks, 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)\(^2\). 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

\(^1\)http://perso.univ-st-etienne.fr/largeron/ANC_Generator
\(^2\)http://snap.stanford.edu/data/index.html
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 GitHub³ and LINQS,⁴ respectively. We provide some basic statistics for these datasets in Table 3, where |V|, |E|, |F|, ⟨d⟩ = 2|E|/|V|, #Comms, and Overlaps = ∑K∗=1 |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 partition, respectively, where K∗ = #Comms 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 [57] finds overlapping communities based on a probabilistic generative model combining network topology and node features. EDCAR [17] 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 [4] 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 [9] is implemented following the exploration principle of Louvain and optimizes the modularity and the proposed inertia-based modularity. NAGC [36] performs non-linear attributed network clustering via symmetric non-negative matrix factorization with positive unlabeled learning. kNN-enhance [26] 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 [46] 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 [19] 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 [1] explores locally well-connected overlapping communities by computing community connectedness and neighborhood connectedness scores for each node. BigClam [56] 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 [34] 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) [56], Modularity Q [40], Density [6], and Entropy [26]. Given two partitions C = {C1, . . . , CK} and C∗ = {C∗1, . . . , C∗K∗}, AvgF1 is defined as follows:

³https://github.com/smileyan448/Sinanet
⁴https://linqs.soe.ucsc.edu/data
Fig. 1. Effectiveness evaluation of LSF and baselines using node features.

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

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

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

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

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

$$\text{Density} \; (C_k) = \frac{2E_{\text{in}}^k}{|C_k| \cdot (|C_k| - 1)},$$

where $c_{ik}$ ($c_{jk}$) represents the degree of belonging of node $v_i$ ($v_j$) to the $k$th community, and $E_{\text{in}}^k$ represents the number of internal edges in the $k$th community.

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

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

where $P_{lk}$ is the fraction of nodes with $l$th feature in the $k$th community.

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

### 5.2 Effectiveness Evaluation

To better evaluate the effectiveness of our framework, we first compare the quality of non-overlapping communities found by LSF and six baselines using node features on synthetic datasets, and then comprehensively compare the quality of non-overlapping and overlapping communities found by LSF and twelve baselines on real-world datasets. Figure 1, Figure 2, and Table 4...
Table 4. Overall Performance 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|

Fig. 2. Overall performance evaluation of LSF and baselines using node features.

present the AvgF1 scores for all tested approaches. In Figure 2, since nodes in Sinanet and Diabetes have continuous-valued features, we compare the performance of the proposed framework (LSF) with three baselines considering continuous-valued features (I-Louvain, NAGC, and kNN-enhance). Correspondingly, nodes in Facebook, Twitter, and Gplus have binary features, and we compare the performance of LSF with three baselines considering binary features (CESNA, EDCAR, and CAMAS). The NaNs in Table 4 are due to some baseline tools running out of memory or too expensive to run as the scale of the network increases, and the best scenarios are in bold.

We have the following observations and conclusions from these results. On synthetic network datasets, the performance of all approaches is stable with a small number of nodes. As the number of nodes increases, the performance of each approach degrades gradually and fluctuates more obviously, while our framework still retains its superiority and robustness. On real-world network datasets, in terms of approaches that consider continuous-valued features, kNN-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 framework performs significantly better than all of them, which also
Fig. 3. Efficiency evaluation of LSF and baselines using node features.

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 |

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 triangle-based 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 effectiveness 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 3 and Table 5. Like Figure 2 and Table 4, we first compare I-Louvain, NAGC, kNN-enhance, and LSF on Sinanet and Diabetes, and then compare CESNA, EDCAR, CAMAS, and LSF on Facebook, Twitter, and Gplus. Finally, NaNs also mean that the baselines are too expensive to run and the best scenarios are in bold.

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 network 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 effects of $W^{CC^*}$ and 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 4.

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 $W^{CC^*}$ and 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
Triangle-oriented Community Detection Considering Node Features and Network Topology

Fig. 5. Evaluation of the effects of $WCC^*$ and two-level constraints.

We can find: Whether the network is dense or sparse, extending $WCC$ to $WCC^*$ significantly improves overall performance. 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 to extend $WCC$ to $WCC^*$, and with two-level constraints is better than without. However, if only one constraint is considered, then 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 6.

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 community, 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 synthetic and 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 conduct a more in-depth critical review of community detection in attributed networks, especially the extensive work oriented by deep learning, such as graph neural network-based approaches and temporal graph learning.

REFERENCES

[1] Sanghamitra Bandyopadhyay, Garisha Chowdhary, and Debarka Sengupta. 2015. FOCS: Fast overlapped community search. *IEEE Trans. Knowl. Data Eng.* 27, 11 (2015), 2974–2985. DOI: https://doi.org/10.1109/TKDE.2015.2445775

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

[3] Kamal Berahmand, Sogol Haghani, Mehrdad Rostami, and Yuefeng Li. 2022. A new attributed graph clustering by using label propagation in complex networks. *J. King Saud Univ. Comput. Info. Sci.* 34, 5 (2022), 1869–1883. DOI: https://doi.org/10.1016/j.jksuci.2020.08.013

[4] Zhan Bu, Guangliang Gao, Huijia Li, and Jie Cao. 2017. CAMAS: A cluster-aware multiagent system for attributed graph clustering. *Info. Fusion* 37 (2017), 10–21. DOI: https://doi.org/10.1016/j.inffus.2017.01.002

[5] Zhan Bu, Huijia Li, Jie Cao, Zhen Wang, and Guangliang Gao. 2019. Dynamic cluster formation game for attributed graph clustering. *IEEE Trans. Cybernet.* 49, 1 (2019), 328–341. DOI: https://doi.org/10.1109/TCYB.2017.2772880

[6] Tanmoy Chakraborty, Ayushi Dalmia, Animesh Mukherjee, and Niloy Ganguly. 2017. Metrics for community analysis: A survey. *Comput. Surveys* 50, 4 (2017), 54:1–54:37. DOI: https://doi.org/10.1145/3091106

[7] Petr Chunaev. 2020. Community detection in node-attributed social networks: A survey. *Comput. Sci. Rev.* 37 (2020), 100286–100310. DOI: https://doi.org/10.1016/j.cosrev.2020.100286

[8] Matteo Cinelli, Gianmarco De Francisci Morales, Alessandro Galeazzi, Walter Quattrociocchi, and Michele Starnini. 2021. The echo chamber effect on social media. *Proc. Natl. Acad. Sci. U.S.A.* 118, 9 (2021), e2023301118:1–8. DOI: https://doi.org/10.1073/pnas.2023301118
Triangle-oriented Community Detection Considering Node Features and Network Topology

[9] David Combe, Christine Largeron, Mathias Géry, and Előd Egyed-Zsigmond. 2015. I-louvain: An attributed graph clustering method. In Proceedings of the 14th International Symposium on Advances in Intelligent Data Analysis (Lecture Notes in Computer Science), Vol. 9385. Springer, 181–192. DOI: https://doi.org/10.1007/978-3-319-24465-5_16

[10] Kaize Ding, Jundong Li, and Huan Liu. 2019. Interactive anomaly detection on attributed networks. In Proceedings of the 12th International Conference on Web Search and Data Mining. ACM, 357–365. DOI: https://doi.org/10.1145/3299600.3299646

[11] Ernesto Estrada. 2011. Combinatorial study of degree assortativity in networks. Phys. Rev. E 84, 4 (2011), 047101:1–4.

[12] Issam Falih, Nistor Grozavu, Rushed Kanawati, and Younés Bennani. 2018. Community detection in attributed network. In Proceedings of the Web Conference. 1299–1306.

[13] Santo Fortunato and Marc Barthelemy. 2007. Resolution limit in community detection. Proc. Natl. Acad. Sci. U.S.A. 104, 1 (2007), 36–41.

[14] Santo Fortunato and Darko Hric. 2016. Community detection in networks: A user guide. Phys. Rep. 659 (2016), 1–44.

[15] Santo Fortunato and Mark E. J. Newman. 2022. 20 years of network community detection. Nature Phys. 18, 8 (2022), 848–850.

[16] Guangliang Gao, Zhihai Wu, Lu Zhang, Jie Cao, and Xinzhe Qi. 2022. Community detection via local learning based on generalized metric with neighboring regularization. IEEE Trans. Syst. Man Cybernet.-Syst. 52, 1 (2022), 498–510. DOI: https://doi.org/10.1109/TSMC.2020.3003019

[17] Stephan Günnemann, Brigitte Boden, Ines Färber, and Thomas Seidl. 2013. 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 (Lecture Notes in Computer Science), Vol. 7818. Springer, 261–275. DOI: https://doi.org/10.1007/978-3-642-37453-1_22

[18] Chaobo He, Xiang Fei, Qweil Cheng, Hanchoo Li, Zeng Hu, and Yong Tang. 2022. A survey of community detection in complex networks using nonnegative matrix factorization. IEEE Trans. Comput. Soc. Syst. 9, 2 (2022), 440–457. DOI: https://doi.org/10.1109/TCCS.2021.3114419

[19] Alexandre Hollocou, Julien Maudet, Thomas Bonald, and Marc Lelarge. 2017. A linear streaming algorithm for community detection in very large networks. Retrieved from https://arxiv.org/abs/1703.02955.

[20] Lun Hu, Keith CC Chan, Xiaohui Yuan, and Shengwu Xiong. 2019. A variational Bayesian framework for cluster analysis in a complex network. IEEE Trans. Knowl. Data Eng. 32, 11 (2019), 2115–2128.

[21] Lun Hu, Xiangyu Pan, Zehai Tang, and Xin Luo. 2021. A fast fuzzy clustering algorithm for complex networks via a generalized momentum method. IEEE Trans. Fuzzy Syst. 30, 9 (2021), 3473–3485.

[22] Lun Hu, Xiangyu Pan, Hong Yan, Pengwei Hu, and Tiantian He. 2021. Exploiting higher-order patterns for community detection in attributed graphs. Integr. Comput.-Aided Eng. 28, 2 (2021), 207–218. DOI: https://doi.org/10.1023/JICA-200645

[23] Lun Hu, Jun Zhang, Xianguo Pan, Hong Yan, and Zhu-Hong You. 2021. HiSCF: Leveraging higher-order structures for clustering analysis in biological networks. Bioinformatics 37, 4 (2021), 542–550.

[24] Shixun Huang, Yuchen Li, Zhifeng Bao, and Zhao Li. 2021. Towards efficient motif-based graph partitioning: An adaptive sampling approach. In Proceedings of the 37th International Conference on Data Engineering. IEEE, 528–539. DOI: https://doi.org/10.1109/ICDE51399.2021.00052

[25] Muhammad Aqib Javed, Muhammad Shahzad Younis, Siddique Latif, Junaid Qadir, and Adeel Baig. 2018. Community detection in networks: A multidisciplinary review. J. Netw. Comput. Appl. 108 (2018), 87–111. DOI: https://doi.org/10.1016/j.jnca.2018.02.011

[26] Caiyan Jia, Yafang Li, Matthew B. Carson, Xiaoyang Wang, and Jian Yu. 2017. Node attribute-enhanced community detection in complex networks. Sci. Rep. 7, 1 (2017), 1–15.

[27] Di Jin, Zhizhu Ye, Pengfei Jiao, Shuirui Pan, Dongxiao He, Jia Wu, Philip S. Yu, and Weixiong Zhang. 2023. A survey of community detection approaches: From statistical modeling to deep learning. IEEE Trans. Knowl. Data Eng. 35, 2 (2023), 1149–1170. DOI: https://doi.org/10.1109/TKDE.2021.3104155

[28] Christine Largeron, Pierre-Nicolas Mougel, Reihanreh Rabbany, and Osmar R. Zaïane. 2015. Generating attributed networks with communities. PloS One 10, 4 (2015), e0122777:1–13.

[29] Pei-Zhen Li, Ling Huang, Chang-Dong Wang, and Jian-Huang Lai. 2018. Community detection using attribute homogenous motif. IEEE Access 6 (2018), 47707–47716. DOI: https://doi.org/10.1109/ACCESS.2018.2867549

[30] Pei-Zhen Li, Ling Huang, Chang-Dong Wang, and Jian-Huang Lai. 2019. EdMot: An edge enhancement approach for motif-aware community detection. In Proceedings of the 25th International Conference on Knowledge Discovery and Data Mining. ACM, 479–487. DOI: https://doi.org/10.1109/KDD.2019.00087

[31] Zhangtao Li, Jing Liu, and Kai Wu. 2018. A multijobjective evolutionary algorithm based on structural and attribute similarities for community detection in attributed networks. IEEE Trans. Cybernet. 48, 7 (2018), 1963–1976. DOI: https://doi.org/10.1109/TCYB.2017.2720180

ACM Transactions on the Web, Vol. 18, No. 1, Article 11. Publication date: November 2023.
[32] Fanzhen Liu, Zhao Li, Baokun Wang, Jia Wu, Jian Yang, Jiaming Huang, Yiqing Zhang, Weiqiang Wang, Shan Xue, Surya Nepal, and others. 2022. eRiskCom: An e-commerce risky community detection platform. Proc. VLDB Endow. 31, 5 (2022), 1085–1101. DOI: https://doi.org/10.14778/3307778-0010-007723-z

[33] Yang Liu, Chen Liang, Xiangnan He, Jiaying Peng, Zhibin Zheng, and Jie Tang. 2022. Modelling high-order social relations for item recommendation. IEEE Trans. Knowl. Data Eng. 34, 9 (2022), 4385–4397. DOI: https://doi.org/10.1109/TKDE.2020.3094463

[34] Meilian Lu, Zhenglin Zhang, Zhihe Qu, and Yu Kang. 2019. LPANNI: Overlapping community detection using label propagation in large-scale complex networks. IEEE Trans. Knowl. Data Eng. 31, 9 (2019), 1736–1749. DOI: https://doi.org/10.1109/TKDE.2018.2866424

[35] Tianshu Lyu, Lidong Bing, Zhao Zhang, and Yan Zhang. 2016. Efficient and scalable detection of overlapping communities in big networks. In Proceedings of the 16th International Conference on Data Mining. IEEE Computer Society, 1071–1076. DOI: https://doi.org/10.1109/ICDM.2016.0138

[36] Seiji Maekawa, Koh Takeuchi, and Makoto Onizuka. 2018. Non-linear attributed graph clustering by symmetric NMF with PU learning. Retrieved from https://arxiv.org/abs/1810.09946.

[37] Deepanshu Malhotra and Anuradha Chug. 2021. A modified label propagation algorithm for community detection in attributed networks. Int. J. Info. Manage. Data Insights 1, 2 (2021), 100030–100041.

[38] Xiangmao Meng, Wenkai Li, Xiaoqing Peng, Yaohang Li, and Min Li. 2021. Protein interaction networks: Centrality, modularity, dynamics, and applications. Front. Comput. Sci. 15, 6 (2021), 1–17.

[39] Vincenzo Moscato, Antonio Picariello, and Giancarlo Sperli. 2019. Community detection based on game theory. Eng. Appl. Artific. Intell. 85 (2019), 773–782. DOI: https://doi.org/10.1016/j.engappai.2019.08.003

[40] Mark E. J. Newman. 2006. Modularity and community structure in networks. Proc. Natl. Acad. Sci. U.S.A. 103, 20 (2006), 8577–8582.

[41] Ashwin Paranjape, Austin R. Benson, and Jure Leskovec. 2017. Motifs in temporal networks. In Proceedings of the 10th International Conference on Web Search and Data Mining. ACM, 601–610. DOI: https://doi.org/10.1145/3018661.3018731

[42] Leto Peel, Daniel B. Larremore, and Aaron Clauset. 2017. The ground truth about metadata and community detection in networks. Sci. Adv. 3, 5 (2017), e1602548:1–8.

[43] Arnau Prat-Perez, David Dominguez-Sal, and Josep-Lluís Larriba-Pey. 2014. High quality, scalable and parallel community detection for large real graphs. In Proceedings of the 23rd International Conference on World Wide Web. ACM, 225–236. DOI: https://doi.org/10.1145/2566486.2568010

[44] Mohammad Rezaei and Pasi Fränti. 2016. Set matching measures for external cluster validity. IEEE Trans. Knowl. Data Eng. 28, 8 (2016), 2173–2186. DOI: https://doi.org/10.1109/TKDE.2016.2551240

[45] Ryan A. Rossi, Nesreen K. Ahmed, and Eunyee Koh. 2018. Higher-order network representation learning. In Proceedings of the Web Conference. ACM, 3–4. DOI: https://doi.org/10.1145/3184558.3186900

[46] Martin Rosvall and Carl T. Bergstrom. 2007. Maps of information flow reveal community structure in complex networks. Retrieved from https://arxiv.org/abs/0707.0609.

[47] Jianbo Shi and Jitendra Malik. 2000. Normalized cuts and image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 22, 8 (2000), 888–905. DOI: https://doi.org/10.1109/34.868688

[48] Laura M. Smith, Linhong Zhu, Kristina Lerman, and Allon G. Percus. 2016. Partitioning networks with node attributes with PU learning. Retrieved from https://arxiv.org/abs/1810.00946.

[49] Konstantinos Sotiropoulos and Charalampos E. Tsourakakis. 2021. Triangle-aware spectral sparsifiers and community detection. In Proceedings of the 27th International Conference on Knowledge Discovery and Data Mining. ACM, 1501–1509. DOI: https://doi.org/10.1145/3447548.3467260

[50] Xing Su, Shan Xue, Fanzhen Liu, Jia Wu, Jian Yang, Chuan Zhou, Wenbin Hu, Cecile Paris, Surya Nepal, Di Jin, and others. 2022. A comprehensive survey on community detection with deep learning. IEEE Trans. Neural Netw. Learn. Syst. (2022), 1–21. DOI: https://doi.org/10.1109/TNNLS.2021.3137396

[51] Yubao Wu, Ruoming Jin, Jing Li, and Xiang Zhang. 2015. Robust local community detection: On free rider effect and its elimination. Proc. VLDB Endow. 8, 7 (2015), 798–809. DOI: https://doi.org/10.14778/2752939.2752948

[52] Feng Xia, Shuo Yu, Chengfei Liu, Jianxin Li, and Ivan Lee. 2022. Chief: Clustering with higher-order motifs in big networks. IEEE Trans. Netw. Sci. Eng. 9, 3 (2022), 990–1005. DOI: https://doi.org/10.1109/TSNE.2021.3108974

[53] Xiaoqin Xie, Mingjie Song, Chiming Liu, Jiaming Zhang, and Jiahui Li. 2021. Effective influential community search based on attributed graph. Neurocomputing 444 (2021), 111–125. DOI: https://doi.org/10.1016/j.neucom.2020.08.088

[54] Xin-Li Xu, Yun-Yue Xiao, Xu-Hua Yang, Lei Wang, and Yan-Bo Zhou. 2022. Overlapping network community detection based on network embedding and parameter-free clustering. Appl. Intell. 52, 7 (2022), 8073–8086.

[55] Zhiqiang Xu, Yiping Ke, Yi Wang, Hong Cheng, and James Cheng. 2012. A model-based approach to attributed graph clustering. In Proceedings of the 31st International Conference on Management of Data. ACM, 505–516. DOI: https://doi.org/10.1145/2215836.2213894
[56] Jaewon Yang and Jure Leskovec. 2013. Overlapping community detection at scale: A nonnegative matrix factorization approach. In Proceedings of the 6th International Conference on Web Search and Data Mining. ACM, 587–596. DOI: https://doi.org/10.1145/2433396.2433471

[57] Jaewon Yang, Julian J. McAuley, and Jure Leskovec. 2013. Community detection in networks with node attributes. In Proceedings of the 13th International Conference on Data Mining. IEEE Computer Society, 1151–1156. DOI: https://doi.org/10.1109/ICDM.2013.167

[58] Hao Yin, Austin R. Benson, Jure Leskovec, and David F. Gleich. 2017. Local higher-order graph clustering. In Proceedings of the 23rd International Conference on Knowledge Discovery and Data Mining. ACM, 555–564. DOI: https://doi.org/10.1145/3097983.3098069

[59] Chen Zhe, Aixin Sun, and Xiaokui Xiao. 2019. Community detection on large complex attribute network. In Proceedings of the 25th International Conference on Knowledge Discovery and Data Mining. ACM, 2041–2049. DOI: https://doi.org/10.1145/3292500.3330721

[60] Yang Zhou, Hong Cheng, and Jeffrey Xu Yu. 2009. Graph clustering based on structural/attribute similarities. Proc. VLDB Endow. 2, 1 (2009), 718–729. DOI: https://doi.org/10.14778/1687627.1687709

Received 9 November 2022; revised 21 August 2023; accepted 12 September 2023